Advancing Eye-Gaze Writing through the Integration of Computer Vision and Predictive Text

***Walid Abdallah Shobaki, Dr. Mariofanna Milanova**

Engineering, University of Arkansas at Little Rock (UALR)

***Corresponding Author**

DOI: https://doi.org/10.51244/IJRSI.2025.12040069

Received: 02 April 2025; Accepted: 03 April 2025; Published: 10 May 2025

ABSTRACT

This study presents a comparative evaluation of two eye-gaze-based text prediction models—YOLOv8 and Haar—both with and without integrating a text suggestion mechanism. The analysis focuses on key performance indicators, including precision, recall, F1-score, and inference time, to assess model accuracy and system efficiency. In the absence of text suggestion, the Haar model demonstrated superior performance, achieving precision, recall, and F1-scores of 0.85, 0.83, and 0.85, respectively, along with an inference time of 52 ms. Conversely, YOLOv8 yielded slightly lower precision (0.88), recall (0.80), and F1-score (0.83), with a higher inference time of ms. However, the integration of text suggestion revealed the advantages of YOLOv8 in terms of scalability and computational stability. Its compact architecture allowed it to accommodate the added complexity without compromising system performance, whereas the larger Haar model experienced frequent crashes. These results indicate that while Haar offers higher accuracy in isolation, YOLOv8 is better suited for real-time applications when text suggestion is incorporated due to its efficiency and robustness. The findings emphasize the critical role of model size and inference speed in embedded systems, particularly in practical domains such as healthcare and assistive technologies. Future research will aim to enhance the YOLOv8 model by leveraging the latest developments in the YOLO framework, integrating online learning for dynamic adaptability and exploring advanced text recommendation techniques to further improve user experience and system performance.

Keywords: Eye-Gaze, YOLO, Haar, MPIIgaze, Text Suggestion, computer vision.

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. Journal Not Specified 2025, 1, 0. https://doi.org/

INTRODUCTION

Eye-gaze writing represents a promising modality for human-computer interaction, particularly for individuals with severe motor impairments. By enabling users to generate text using only their eye movements, this technology offers an alternative communication channel that can significantly enhance autonomy and quality of life. However, despite their potential, current eye-gaze writing systems encounter several limitations, including slow input speed, diminished accuracy in suboptimal conditions, and increased user fatigue. Performance is often compromised by external factors such as low-light environments, involuntary head movements, and frequent blinking, all of which hinder the precision of eye-tracking and cursor control. Moreover, prolonged use of such systems can lead to eye strain, further degrading performance over time. These challenges underscore the necessity of developing more robust, efficient, and user-friendly solutions that enhance both the Version April 10, 2025 submitted to Journal Not Specified usability and reliability of eye-gaze-based communication systems.

Eye-gaze writing holds significant potential as an alternative communication modality for individuals with

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue IV April 2025

limited mobility. Although current systems offer valuable assistance, they face several inherent challenges, such as slow input speed, reduced accuracy under various environmental conditions, and increased user fatigue. Factors like low-light settings, excessive head movement, and blinking can severely impact eye-tracking performance, hindering users' ability to control the cursor or select items with precision. Additionally, prolonged usage of these systems can lead to eye strain, further compromising speed and accuracy. These obstacles highlight the critical need for advancements that can improve both the usability and efficiency of eye-gaze-based writing systems. Building on our previous work [], which involved developing a comprehensive dataset for eye-gaze recognition, this study aims to address these limitations by integrating a text suggestion system that predicts user intentions based on gaze patterns. This enhancement not only increases typing speed but also reduces the cognitive and physical strain that users experience, making eye-gaze writing more effective and accessible for real-time applications.

Building upon the foundation of our previous work, the present study seeks to improve the performance of eye-gaze-based writing systems through the integration of a text suggestion mechanism. Traditional eye-gaze systems require users to fixate on individual letters to construct words, a process that is inherently slower and more labor-intensive compared to conventional typing methods. To address this limitation, we introduce a predictive text suggestion component that anticipates the user's intended words based on their gaze patterns. By reducing the number of required fixations, this approach significantly accelerates text input while simultaneously reducing cognitive load and physical strain. The incorporation of text suggestion not only enhances system efficiency but also provides a more intuitive and user-friendly communication experience, especially for individuals with limited mobility.

To evaluate the effectiveness of the proposed text suggestion system, we conducted a series of user tests comparing typing speed and accuracy between the standard eye-gaze input and our enhanced system. By analyzing performance across various conditions, we demonstrate that the integration of predictive text significantly enhances the usability and practicality of eye-gaze writing technology.

- Enhancing Eye-Gaze Writing Speed with Text Suggestions: To bring eye-gaze typing speeds closer to those of traditional keyboard users, we incorporate predictive text suggestions into the eye-gaze writing system. This enhancement substantially increases writing efficiency by reducing the time required to form words and sentences while also alleviating user fatigue.

- Optimizing Eye-Gaze Detection Models: We assess several object detection and image segmentation models to improve the accuracy of eye-gaze recognition. Our results indicate that the Haar classifier achieves the highest accuracy (0.85), while YOLOv8 offers a balanced solution, providing a respectable accuracy of 0. alongside faster processing and a compact model size (6. KB), making it particularly suitable for real-time applications.

- Developing a Comprehensive Eye-Gaze Dataset: To overcome the limitations of existing eye-tracking datasets, we compile a new dataset by merging multiple publicly available datasets. This enhanced dataset improves model training by incorporating a wide range of conditions, such as eye fatigue, irritation, blinking, and low-light scenarios, which are crucial for real-world performance.

- Real-World Validation Through User Testing: We perform extensive user testing to evaluate the performance of our models under real-world conditions. The tests involve users wearing glasses and individuals with various eye conditions, ensuring that the system remains robust, accessible, and effective in diverse practical settings.

RELATED WORK

In the early 20th century, researchers such as J. W. Baird and L. T. Clark introduced the concept of utilizing eye movements as an alternative input method to manual interaction. Significant progress in eye-tracking technology emerged by the mid-20th century, laying the foundation for practical eye-gaze writing systems. However, early implementations faced considerable challenges, including large, cumbersome hardware and limited accuracy.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue IV April 2025

With advancements in modern technology, these limitations have been largely addressed, leading to the development of more precise, reliable, and cost-effective eye-gaze tracking systems.

Overview of Traditional Eye-Gaze Tracking Methods

Pupil Center Corneal Reflection Technique (PCCR)

The Pupil Center Corneal Reflection (PCCR) technique operates similarly to a small flashlight directing a beam of light into the eye, with the reflected light being used to accurately determine the user's gaze position. While this method is cost-effective, it presents two significant limitations. First, it requires frequent calibration for each user [2]. Second, the system exhibits a low tolerance for head movements, necessitating that users remain uncomfortably still during operation. These challenges have hindered the widespread adoption of PCCR-based systems. As a result, numerous studies have been conducted to enhance this technique and address its limitations [3].

Appearance-Based Methods

Appearance-based methods estimate gaze direction by analyzing the overall eye region in images or video frames. These techniques track the gaze by examining features such as pupil size and iris patterns. Research [4] has shown the potential of this approach for Eye-Gaze Writing (EGW), but several challenges persist. Variations in eye color and

shape among individuals can affect accuracy, while external factors like lighting changes, sunglasses, or makeup may further hinder performance. To enhance the robustness of appearance-based methods, researchers developed the MPIIGaze dataset [5]. which includes images capturing a wide range of gaze directions and head poses in real-world environments. Furthermore, studies [6] have investigated the use of convolutional neural networks (CNNs), such as AlexNet and VGG, for feature extraction in appearance-based gaze tracking.

Saccadic and Fixational Analysis

Saccadic and fixational analysis monitors eye movements, including rapid flicks (saccades) and brief pauses (fixations), to determine the user's gaze position [7]. By analyzing these movement patterns, the system can infer and select the intended letter. However, a key limitation of this approach is the natural tendency of the eyes to twitch or move involuntarily, which may lead to misinterpretations and cause the system to register unintended selections. Ongoing research aims to refine these methods to better distinguish between deliberate gazes at a target letter and involuntary eye movements, thereby enhancing the accuracy and reliability of this technique.

Model-Based Methods

Model-based methods estimate gaze direction by utilizing geometric and mathematical models of the eye. These approaches rely on key eye landmarks, such as the pupil center, corneal reflection (PCCR), and overall eye geometry, to determine gaze position. One of the primary advantages of model-based methods is their interpretability, as well as their ability to function effectively with fewer training samples. However, these methods are highly sensitive to external factors, including lighting variations, head movements, and individual differences in eye structure. Studies [] have demonstrated the effectiveness of model-based methods for Eye-Gaze Writing (EGW), although they also emphasize the computational demands necessary to achieve smooth real-time performance [8].

Haar Cascade Classifier

The Haar Cascade Classifier, as shown in Figure 1, was introduced by Viola and Jones in 2001 [10,11]. It leverages Haar-like features—patterns of black and white rectangular regions—to analyze contrast differences within an image [12]. In eye-gaze applications, Haar-based detection identifies eye regions by scanning facial features and recognizing patterns, such as the contrast between the dark iris and the lighter sclera.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue IV April 2025

Figure 1. Haar features used for Viola Jones's face detection method

Suggestion Techniques for Enhancing Eye-Gaze Writing

Building on our previous research [1], which identified YOLO and Haar as the most effective models for eye-gaze tracking, we aim to further enhance their performance by incorporating advanced text suggestion techniques. These algorithms are crucial for improving the speed and accuracy of gaze-based writing by predicting the next word or character, thereby reducing user effort and minimizing fatigue. Various text suggestion methods have been developed in natural language processing (NLP), each offering different levels of accuracy, processing speed, dataset requirements, and model size.

N-Gram Models

N-Gram models [13] predict the next word by analyzing the probability of word sequences within a given dataset. These models offer moderate accuracy, which depends on factors such as dataset size and domain specificity. N-Gram models are computationally efficient due to their reliance on simple probability calculations, making them suitable for real-time applications. Typically trained on large text corpora such as Wikipedia or Common Crawl, they maintain a relatively small model size. However, a key limitation is their inability to capture long-range language dependencies, resulting in lower prediction accuracy compared to deep learning-based methods.

Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM)

RNNs and LSTMs [14] enhance text prediction by capturing sequential dependencies in text, allowing for more context-aware suggestions. Compared to N-Gram models, they offer higher accuracy, especially for longer phrases. However, their computational complexity results in moderate processing speeds, requiring more resources than probabilistic models. These models are typically trained on large-scale NLP datasets, such as OpenSubtitles, BooksCorpus, or domain-specific texts. Their model size ranges from medium to large, depending on the architecture and dataset size. While they improve prediction accuracy, their higher computational demands may impact real-time performance in eye-gaze writing systems.

Transformer-Based Models (BERT, GPT, T5)

Transformer-based models, such as BERT, GPT, and T5 [15], represent the latest advancements in NLP, offering state-of-the-art text prediction accuracy by learning deep contextual relationships in text. These models are pre-trained on extensive datasets, including Common Crawl, Wikipedia, and WebText, which enable them to generate highly accurate suggestions. However, their computational complexity results in slower processing speeds compared to traditional approaches. Techniques such as model distillation and pruning can improve efficiency, but their large size and resource-intensive nature present challenges for real-time deployment in eye-gaze writing systems. Despite their superior performance, lightweight alternatives may be more suitable for real-time applications.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue IV April 2025

Statistical Language Models with Word Completion

Statistical language models [16], when combined with word completion techniques, enhance text prediction by suggesting full words based on partially entered input. These models leverage frequency-based predictions or hybrid approaches that integrate deep learning with statistical methods. Their efficiency arises from the use of lookup tables and probability calculations, which make them faster than deep learning-based models. The accuracy of these models depends on the quality and size of the training dataset, with domain-specific corpora improving precision. Due to their relatively small to medium model size, statistical language models offer a balanced trade-off between speed and accuracy, making them suitable for real-time applications, such as eye-gaze writing. Table 1 provides a comparison of various text suggestion algorithms, highlighting their key features, strengths, and limitations.

Table 1. Comparison of Text Suggestion Algorithms.

Algorithm	**Accuracy**	**Speed**	**Training Dataset**	**Model Size**
N-Gram RNN/LSTM	Moderate High	Fast Moderate	Wikipedia, Common Crawl OpenSubtitles, BooksCorpus	Small Medium
Transformer (BERT, GPT)	Very High	Slow (unless optimized)	Common Crawl, WebText	Large
Statistical Word Completion	Moderate to High	Fast	Domain-Specific Text	Small to Medium
FastAutocomplete	Moderate to High	Very Fast	Customizable	Small

METHODOLOGY

Datasets, and Data Preprocessing.

We build upon our previous work by employing a combination of datasets and preprocessing techniques. Initially, we utilized the MPIIGAZE dataset [7] [17], which comprises 213, images collected under diverse conditions. However, after conducting several trials with multiple datasets, we created a new, more comprehensive dataset by combining several existing datasets, each contributing distinct advantages.

The combined dataset includes:

- The Eyes Dataset [18], which contains images classified into two categories: "Eyes OPEN" and "Eyes CLOSE."
- The Dataset-Pupil [19], consisting of images captured under purple lighting conditions, with and without glasses.
- The Pupil Detection Computer Vision Project [20], which provides 5, images taken in various lighting conditions.
- The Pupils Computer Vision Project [27], comprising 1, images featuring diverse lighting, eye colors, and makeup, making it a valuable resource for pupil detection. This newly created dataset aims to enhance the robustness of our model by incorporating a wide range of eye conditions and environmental factors, thereby improving the generalizability and accuracy of the eye-gaze detection system.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue IV April 2025

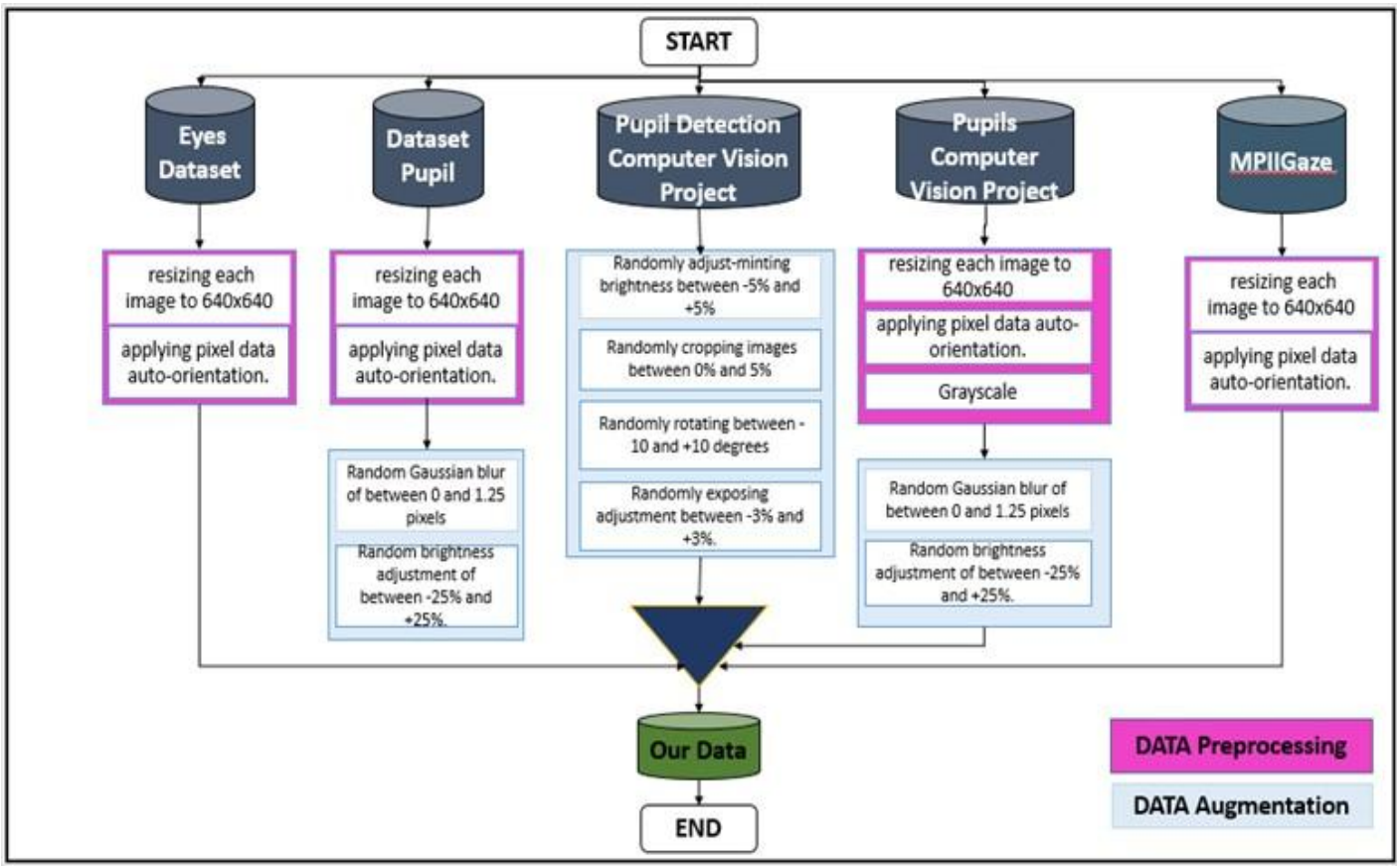


Figure 2. Combination of multiple datasets to create the final dataset used in this study.

Figure 2 illustrates the data preprocessing and augmentation pipeline used to merge multiple eye-gaze datasets. The images were resized to a resolution of 640x640 pixels, with automatic orientation adjustment applied. Augmentation techniques were customized for each dataset to improve model performance and mitigate potential biases. The resulting combined dataset, referred to as Our Data, comprises 221, images collected under diverse conditions, including varying lighting, makeup, and glasses. This integration, coupled with tailored augmentation strategies, was designed to enhance the model's generalization capability and minimize the risk of overfitting.

Preprocessing Steps

- Resize images to 640x640 pixels.
- Apply auto-orientation for pixel alignment.
- Implement dataset-specific augmentation techniques.
- Combine augmented data into a single dataset (Our Data).
- Ensure diverse conditions, including lighting, makeup, and glasses.

Table 2 summarizes the datasets used in this study, detailing the key characteristics and conditions of each dataset.

Table 2. Summary of Datasets Used.

Aspect	**No. Of Images**	**Datasets Properties**
MPIIGaze	213,659	Diverse conditions with varying appearance and illumination.
Eyes Dataset	857	Eyes images: Open and Close

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue IV April 2025

Dataset-Pupil	308	Eyes images with and without glasses in purple lighting
Pupil Detection Computer Vision Project	5193	Eye images in diverse lighting conditions.
Pupils Computer Vision Project	1344	Eyes images in gray, with and without makeup

Proposed Gaze Prediction Models

Building upon our previous study [1], and as summarized in Table 3, the YOLOv8 model and Haar Cascade Classifier were selected for their strong performance in eye-gaze tracking. The selection of these models was based on their performance metrics, including precision, recall, and F1 score, as detailed in Table 3. A comprehensive discussion of these evaluation metrics—such as precision, recall, F1 score, similarity, and inference time—along with their relevance in this study, will be provided in the Evaluation Metrics section.

Table 3. Comparison of The proposed models. [1]

Model	**Precision**	**Recall**	**F1 Score**	**Inference Time (s)**	**Model Size (KB)**
Haar	0.85	0.83	0.85	45	97.358
YOLOv8	0.88	0.80	0.83	52	6.083
DeepLabv3	0.72	0.50	0.82	55	238.906
U-Net	0.50	0.42	0.84	48	121.234
Faster R-CNN	0.15	0.05	0.82	41	534.013
SSD	0.40	0.38	0.49	51	87.071
SimCLR	0.05	0.02	0.77	43	45.327

YOLOv8 was selected for its real-time object detection capabilities, providing an optimal balance between speed and accuracy, making it highly suitable for real-time applications. Haar was chosen for its precision in detecting the eye region, especially in cases where the iris is less distinguishable.

Alternative architectures, such as EfficientDet and DETR, were not considered due to computational constraints and their incompatibility with the current experimental setup.

YOLO: You Only Look Once.

YOLOv8, introduced by Ultralytics on January 10, 2023, represents a significant advancement in detection accuracy and speed compared to its predecessors. The model is organized into three primary components, as shown in Figure 3: the backbone, neck, and head [21]. The backbone utilizes CSPDarknet53 [22], a robust feature extraction network that improves the model's ability to process complex visual data. The neck incorporates the PAN-FPN system [23], which enhances feature fusion, thereby improving overall performance. The head of YOLOv8 adopts a decoupled structure with separate processes for object classification and localization, allowing the model to optimize each task independently. Furthermore, the Task-Aligned Assigner [24] is integrated into the training process, dynamically assigning samples and improving both detection accuracy and model robustness.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue IV April 2025

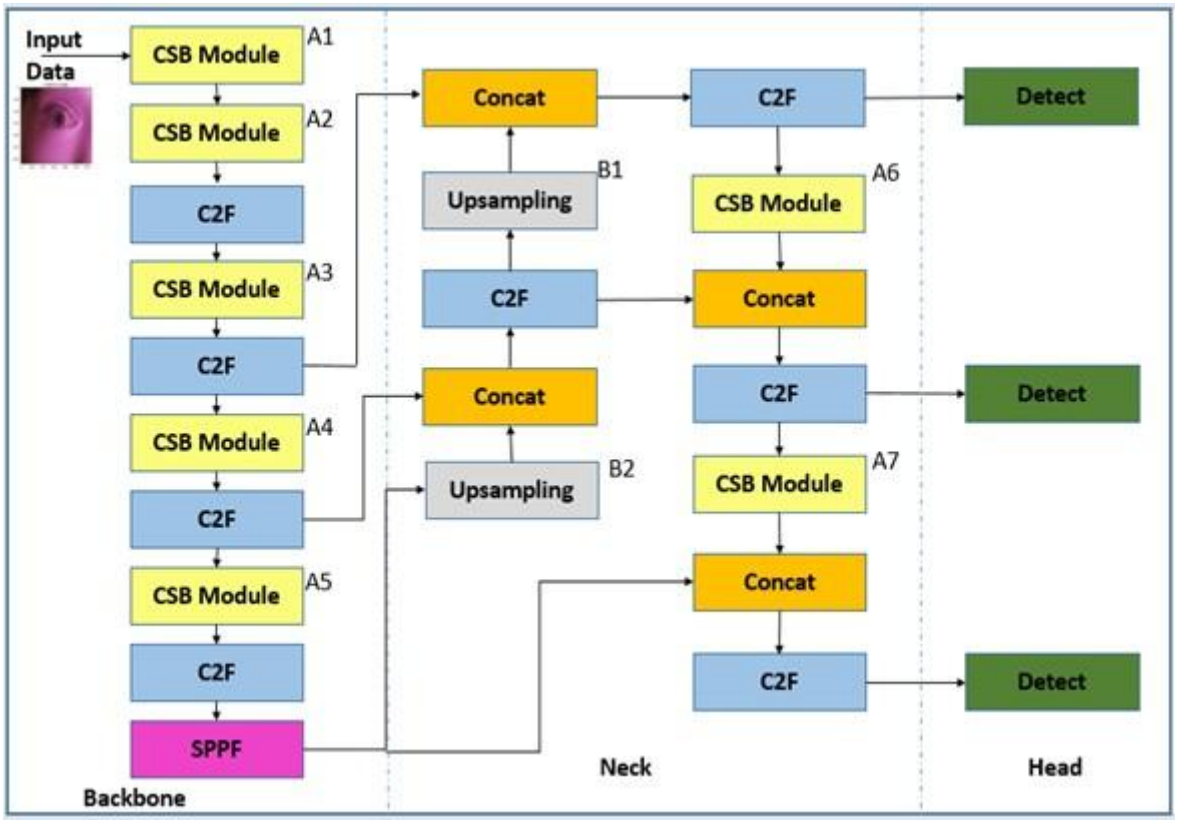


Figure 3. The network structure of YOLOv8.

Proposed Text Suggestion Models

We integrate FastAutocomplete, a text suggestion model, to enhance the speed and accuracy of the eye-gaze writing system. FastAutocomplete [25] predicts the next word or character based on partial user input, thereby reducing the number of required gaze selections and improving the overall user experience.

FastAutocomplete

FastAutocomplete is a highly efficient and lightweight text suggestion algorithm designed specifically for real-time applications, such as eye-gaze writing systems. Its primary strength lies in delivering rapid predictions with minimal computational overhead, making it particularly well-suited for environments that demand low-latency responses.

- Accuracy: The accuracy of FastAutocomplete generally ranges from moderate to high, depending on the quality and breadth of the training data. Since the model is based on pre-established patterns and datasets, its performance is strongly influenced by the relevance and diversity of the training data used.

- Speed: A key feature of FastAutocomplete is its exceptional speed, driven by the use of precomputed index structures and trie-based data organization. These mechanisms enable fast retrieval of word suggestions based on partial input, significantly reducing processing time. This makes FastAutocomplete an ideal solution for real-time applications where minimizing latency is crucial.

- Dataset: The dataset used to train FastAutocomplete can be customized for specific domains, improving its ability to provide relevant word suggestions. For instance, incorporating domain-specific corpora such as medical or legal texts can enhance prediction accuracy in those fields. This flexibility makes FastAutocomplete a versatile tool, adaptable to a wide range of applications.

- Model Size: The algorithm is known for its small model size, making it ideal for deployment in real-time systems with limited computational resources. This compactness ensures fast loading and inference times, crucial for smooth and uninterrupted user interaction.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue IV April 2025

Integrating FastAutocomplete with eye-gaze tracking models like YOLO and Haar enhances both the speed and accuracy of gaze-based writing systems. By combining FastAutocomplete's low-latency predictions with robust object detection models, we create an efficient, real-time assistive system that significantly improves the user experience. This integration aims to deliver a more responsive and intuitive writing system, specifically tailored to the needs of individuals who rely on eye-gaze for communication.

Evaluation Metrics

Several evaluation metrics were used to assess the models' performance throughout the training and user testing. Precision (P) is the ratio of true positive predictions to the total number of positive predictions made by the model. A higher precision value signifies that a larger proportion of the predicted positive cases are accurate.

$$P = TP/(TP + FP) \quad (1)$$

Where: TP represents True Positives, FP denotes False Positives, and FN refers to False

Negatives.

Recall (R) is the ratio of correctly identified positive observations to the total number of actual positive instances in the dataset. A higher recall value signifies improved performance in detecting the positive class.

$$R = TP/(TP + FN) \quad (2)$$

F1 Score is the harmonic mean of Precision and Recall, offering a unified metric that balances both measures. It is particularly valuable in situations where a balance between Precision and Recall is required, especially when it is crucial to minimize both false positives and false negatives.

$$F1 = 2 \times [(P \times R)/(P + R)] \quad (3)$$

Similarity (S) is a measure of how closely the word written by the user aligns with the intended or actual word. It quantifies the degree of correspondence between the predicted and true text, typically evaluated using different distance or similarity metrics.

$$S = 1 - [D(W1, W2)/\max(|W1|, |W2|)] \quad (4)$$

Where D(W1, W2) is the Levenshtein distance, and |W| is the length of the word.

The total inference time (T) refers to the time taken for the user to input a single word using one of the proposed models. This metric is crucial for evaluating the model's performance, especially in real-time applications where fast response times are essential.

$$Ttotal = \sum Ti^{N}_{i=1} \quad (5)$$

Where: Ttotal = Total inference time, Ti = Inference time for each input, N = Number of inputs.

The mean Average Precision (mAP) is a metric used to evaluate the performance of object detection models by calculating the average precision over multiple classes and varying threshold values. It is an important measure of the model's capability to detect objects accurately at different confidence levels. The mAP is computed as follows:

$$mAP = (1/\sum_{i=1} N^{i})AP \quad (6)$$

Where: N is the number of classes, and APi represents the Average Precision for class i.

The term Confidence refers to the model's level of certainty that a predicted bounding box contains an object of interest. It represents the probability that the detected object belongs to a specific class, acting as a measure of

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue IV April 2025

the prediction's reliability. A higher confidence score suggests a stronger likelihood that the object has been accurately classified.

Confidence = P(Object) × P(Class | Object) (7)

Where: P(Object): The probability that an object exists within the predicted bounding box, P(Class | Object): The probability that the object belongs to a specific class, given that an object exists in the bounding box.

The Mean, also known as the arithmetic average, is a statistical measure that represents the central tendency of a dataset. It is calculated by summing all the values in the dataset and dividing the total by the number of observations. The mean provides insight into the overall distribution of data.

$$\text{Mean}(\mu) = \sum X/N \quad (8)$$

Where N represents the number of users, and $\sum X$ denotes the summation of the precision values for each model across all participants.

The Standard Deviation is defined as the square root of the variance, which is the average of the squared differences from the mean. In summation form, the standard deviation for a population is given by:

$$SD = \sum(X - \mu)/N \quad (9)$$

Experimental setup

We conducted our experiments using Python (version 3.9.20) as the primary programming language. To implement and optimize the various algorithms, we utilized several key libraries, including:

- TensorFlow (version 2.10.0)
- Keras (version 2.10.0)
- PyTorch (version 2.0.0+cu117)
- torchvision (version 0.15.0+cu117)
- Ultralytics (version 8.2.38)

These libraries were integral to the development, training, and optimization of the models, enabling efficient handling of both eye-gaze tracking and text suggestion tasks.

For the experiments, we utilized the GPU and camera from our laptop. The specifications of the laptop are outlined in Table 4 below.

Table 4. Our Hardware Characteristics.

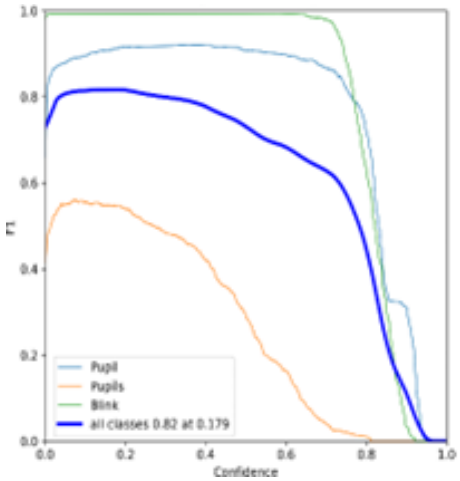
Hardware Component	**Specification**
Storage Capacity	GB
Processor Model	Intel Core i7
Graphics Card Model	NVIDIA GeForce RTX 3050 GDDR6 Graphics
Camera	
Type	HD RGB camera

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue IV April 2025

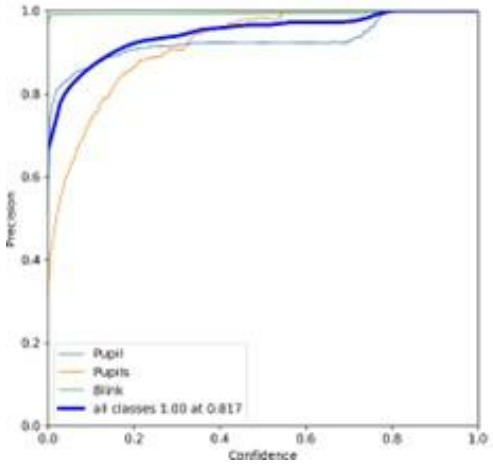
RESULTS AND DISCUSSION

YOLOv8 Training Results

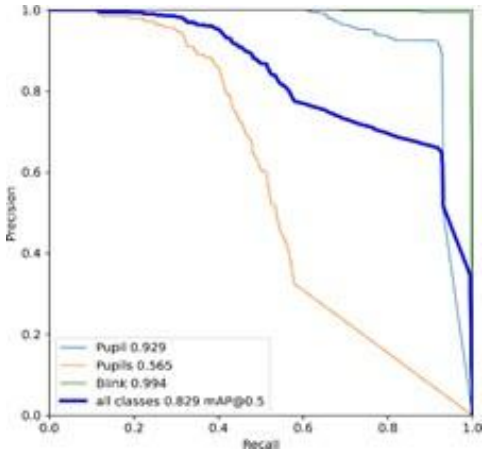
To input images into YOLOv8, all images were resized to a resolution of 640x360 pixels. During the training phase, several performance curves were generated to evaluate the model's effectiveness, as shown in Figure 4.



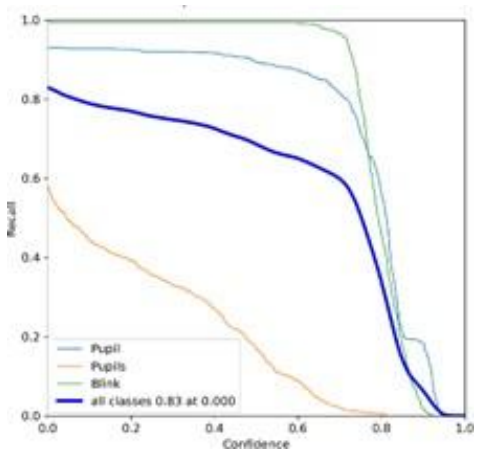
(a) F1-Confidence Curve.



(b) Precision-Confidence Curve.



(c) Precision-Recall Curve.



(d) Recall-Confidence Curve.

Figure 4. YOLOv8 Performance Curves. (a) F1-Confidence Curve, (b) Precision-Confidence Curve, (c) Precision-Recall Curve, (d) Recall-Confidence Curve.

The F1−Confidence Curve shown in Figure 4 assesses the trade−off between precision and recall, with all classes achieving an F1 score of 0. at a confidence threshold of 0.179. Higher curves represent improved model performance. The Precision−Confidence Curve in Figure measures the accuracy of object detection, with all classes reaching a precision of 1. at a confidence of 0.817. As confidence increases, precision improves. The Precision-Recall Curve in Figure reflects the balance between precision and recall, with the mean Average Precision (mAP) for all classes calculated at 0. at a recall rate of 0.5. Finally, the Recall−Confidence Curve in Figure Figure shows the detection of true positive objects, with all classes achieving a recall of 0. at a confidence of 0.080. A higher recall at lower confidence indicates that the model can detect more objects even when less certain.

FastAutocomplete Training

For text suggestion, we curated a dataset composed of the 3, most frequently used English words. The primary objective was to develop a lightweight system, as our approach integrates computer vision with text processing,

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue IV April 2025

constrained by the available laboratory resources. The ultimate aim is to create a compact, efficient application that can be easily installed and used on smartphones.

Moreover, by selecting the most common words, we ensure broad coverage of typical daily conversations. Future improvements could include the incorporation of online learning capabilities, enabling the system to expand its vocabulary based on user input. Alternatively, the system could be personalized for individual users by adapting the text dataset to reflect specific interests. For instance, users with an interest in astronomy could benefit from a system that prioritizes vocabulary related to that field, with similar customizations available for other specialized domains.

User Testing and Evaluation

We assessed the real-time performance of our model to evaluate its speed, ability to manage environmental challenges such as low lighting, user fatigue, and overall user-friendliness. Testing involved a diverse group of participants with varying eye shapes and sizes. To maintain consistent testing conditions across all users and minimize the influence of individual differences, a set of calibration procedures was implemented.

- Initial Setup Calibration: Before each trial, the system was calibrated to record the user's eye-gaze direction at four distinct points: upper left, upper right, lower left, andlower right. These recorded coordinates were subsequently used in the calculations todetermine the user's point of gaze.

- Control Light: Participants were instructed to conduct the test under controlled lighting conditions to minimize the impact of environmental factors on the accuracyof the eye-gaze tracking system.

- Position Standardization: Participants were instructed to maintain a consistent, perpendicular gaze toward the camera throughout the testing process to ensure uniformalignment and reduce variations in eye-gaze tracking accuracy.

To ensure fairness in our evaluation, we selected a mix of individual words and sentences for the test. The experiment included five words: WORK, TIME, GOOD, INFORMATION, and DIFFERENT, along with both short and long sentences. The short sentence "I need help" was chosen to represent common, everyday communication. The longer sentence, "Technology makes life easier for many people," was selected to assess the system's performance with more complex text inputs.

The participants followed a standardized testing procedure, where they were instructed to input words and sentences in a predefined sequence. The goal was for each user to complete all entries within a single session. To evaluate performance, the actual text entered by the user was compared with the intended text. Various evaluation metrics, including precision, recall, F1 score, similarity, and total inference time, were then calculated to assess the system's effectiveness.

Each participant underwent testing under two conditions: once using YOLOv8 and Haar without text suggestion and again with the text suggestion feature enabled. This setup allowed for a comparative analysis, specifically focusing on the impact of text suggestion on both speed and accuracy across different input lengths. The testing process was automated within the system, with the results recorded in an Excel file to facilitate performance comparison across all participants.

Test User Analysis

During user testing, we recruited participants with varying laptop specifications to evaluate our system. In our previous study, YOLOv8 demonstrated lower performance than Haar; however, it had the smallest model weight among the tested models. The lightweight nature of YOLOv8 significantly contributed to its performance, allowing it to achieve near-perfect speed and accuracy. In contrast, Haar exhibited instability, making it impossible to obtain reliable readings, and our system frequently crashed.

We observed that the integration of Haar with real-time systems introduced several performance challenges. First, the classifier processes every incoming frame from the webcam without delay, leading to an excessive

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue IV April 2025

computational load. To address this, we introduced a delay mechanism that processes one of every five frames. However, this modification did not fully resolve the system instability. Second, the execution loop of the Haar model overlaps with that of the text suggestion module. Since the suggestion component relies on eye-gaze data, separating the two processes is not feasible without disrupting the flow of predictions. Third, we noted significant differences in performance across hardware environments. In GPU-equipped setups, the system was able to handle a few words before crashing. Conversely, in CPU-only environments, the system failed to start altogether.

We believe that the Haar-based approach may perform more reliably on high-performance computational platforms capable of handling multi-model processing. Nonetheless, our core objective is to maintain system efficiency while using lightweight models. This strategy minimizes resource consumption and supports broader deployment across various assistive technologies, including smartphone-based floating keyboards, wheelchair control systems, and hospital-grade medical applications.

Table 5 presents the average readings obtained from participants using YOLOv8. The results for Haar are not included, as the majority of participants were unable to complete the experiment due to frequent system crashes. Even in cases where Haar functioned initially, it often failed during the experiment, rendering it unreliable. Consequently, no meaningful results could be derived from the Haar model.

Table 5. YOLOv8 Performance Metrics for Text Suggestion System.

Actual Word	**Written Word**	**Precision**	**Recall**	**F1**	**Similarity**	**Total Inference Time(SEC)**
WORK	work	1	1	1	1	37.072403
TIME	time	1	1	1	1	30.113294
GOOD	good	1	1	1	1	38.950360
INFORMATION	information	1	1	1	1	36.394551
DIFFERENT	different	1	1	1	1	35.662451
I need help	I NEED HELP	1	1	1	1	50.433213
Technology make life easy for	Technology make life easy for	1	1	1	1	
many people	many people					106.5298455

Similarity Score Analysis: The similarity measure, also equal to across all cases, confirms that the predicted words or phrases were perfectly aligned with the actual words.

This implies that YOLOv8's output matched the ground truth exactly, without any discrepancies, which is a key indicator of the algorithm's effectiveness in text recognition tasks.

Precision, Recall, and F1 Score: The YOLOv8 algorithm consistently achieved perfect scores for precision, recall, and F1 score across all the test cases (all values equal to 1). This indicates that the model was able to correctly identify each word or phrase without any false positives or false negatives. Precision measures the proportion of true positive predictions relative to the total number of positive predictions made by the model, while recall reflects the model's ability to correctly identify all relevant instances. An F1 score of 1, which is the harmonic mean of precision and recall, suggests a balanced performance between these two metrics, emphasizing the model's reliability and accuracy in this task.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue IV April 2025

Total Inference Time: The total inference time varied considerably, which reflects the complexity of the input data. The shortest inference time, 30. seconds, was recorded for the word "TIME," while the longest, 106. seconds, was associated with the phrase "Technology makes life easy for many people." This increase in processing time for longer or more complex phrases indicates that the YOLOv8 model's performance is sensitive to input size, which is typical for models that require additional computation for larger or more detailed inputs. Despite this, YOLOv8 performed efficiently in terms of inference time, especially given its capability to process both computer vision and text inputs. The model maintained high accuracy across all samples, making it a strong candidate for applications that involve both visual and textual data.

CONCLUSIONS

The results demonstrate a clear performance disparity between the YOLO and Haar models, particularly when text suggestion is utilized. Without text suggestion, the Haar model outperforms YOLO, achieving a precision of 0.85, recall of 0.83, and F1 score of 0.85, with an inference time of 45 ms and an accuracy of 97.358. In contrast, YOLOv8 achieved a precision of 0.88, recall of 0.80, and F1 score of 0.83, with an inference time of ms and a lower overall accuracy of 6.083. The Haar model, with its high accuracy and reasonable speed, proved to be the best choice for the task when text suggestion was not employed. However, when the system was integrated with text suggestion, the situation changed. Despite the initial advantage of the Haar model without text suggestion, its larger model size caused issues, leading to system instability and crashes during processing.

On the other hand, the compact size of YOLOv8 allowed it to excel in this scenario, as it could handle the additional load from text suggestion without compromising system performance. This demonstrates that while Haar may offer better results without text suggestion, the smaller size of YOLO makes it a more practical choice when text suggestion is included, as it ensures both efficiency and stability in real-time applications. Thus, the integration of text suggestion highlighted the limitations of the Haar model in terms of scalability, while YOLO's compact model size made it more suitable for a fully embedded system capable of handling both high accuracy and the added complexity of text suggestion without crashing.

The deployment of this eye-gaze tracking system in assistive technologies could significantly enhance the independence and quality of life of individuals with disabilities. By allowing hands-free interaction with wheelchairs, mobile devices, and smart home systems, the system could provide greater autonomy, convenience, and accessibility in daily tasks. Continued development and adaptation to real-world challenges will be essential to maximize the impact of the system on assistive technology applications.

Deployment in Wheelchairs: The user could navigate the wheelchair using only their eye movements. By incorporating eye-gaze tracking technology into the wheelchair control system, the user could direct the wheelchair movement more intuitively. For example, focusing on a particular direction could steer the wheelchair, while blinking or other eye gestures could be used to initiate specific actions, such as stopping or opening doors. In addition, this integration could include environmental control systems that enable the user to operate lights, televisions, or other appliances by simply looking at them.

Deployment in Mobile Devices: Users can scroll through websites, answer calls, or select options on the screen by simply looking at icons or buttons, offering a seamless and efficient alternative to touch-based interaction. Furthermore, eye-gaze tracking could facilitate more advanced interactions, such as controlling virtual keyboards, enabling users to type by looking at the appropriate letters or words, and this is our goal.

Expansion to Smart Home Devices: The system could also be extended to smart home technology, where users can control various devices in their living environment. Smart thermostats, lights, security cameras, and even entertainment systems.

Abbreviations

Eye-Gaze Writing (EGW).	Eye Gaze Writing.
YOLO	You Only Look Once.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue IV April 2025

DNN	Deep Neural Network.
CNN	Convolutional Neural Network.
PCCR	Pupil Center Corneal Reflection.
RNNs	Recurrent Neural Networks.
LSTM	Long Short-Term Memory.
BERT	Bidirectional Encoder Representations from Transformers.
GPT	Generative Pretrained Transformer.
T5	Text-to-Text Transfer Transformer.

REFERENCES

1. W. Shobaki. A Comparative Study of YOLO, SSD, Faster R-CNN, and More for Optimized Eye-Gaze Writing. Unpublished manuscript, 2025.
2. Zhang, C.; Chi, J.N.; Zhang, Z.H.; Wang, Z.L. A novel eye gaze tracking technique based on pupil center cornea reflection technique. Jisuanji Xuebao(Chinese Journal of Computers) 2010, 33, 1272–1285.
3. Zhu, Z.; Ji, Q. Eye gaze tracking under natural head movements. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, 2005, Vol. 1, pp. 918–923.
4. Roth, P.M.; Winter, M. Survey of appearance-based methods for object recognition. Inst. for computer graphics and vision, Graz University of Technology, Austria, technical report ICGTR0108 (ICG-TR-01/08) 2008.
5. Fischer, T.; Chang, H.J.; Demiris, Y. Rt-gene: Real-time eye gaze estimation in natural environments. In Proceedings of the Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 334–352.
6. Wang, Y.; Shen, T.; Yuan, G.; Bian, J.; Fu, X. Appearance-based gaze estimation using deep features and random forest regression. Knowledge-Based Systems 2016, 110, 293–301.
7. Salvucci, D.D.; Goldberg, J.H. Identifying fixations and saccades in eye-tracking protocols. In Proceedings of the Proceedings of the 2000 symposium on Eye tracking research & applications, 2000, pp. 71–78.
8. Hansen, D.W.; Ji, Q. In the eye of the beholder: A survey of models for eyes and gaze. IEEE transactions on pattern analysis and machine intelligence 2009, 32, 478–500.
9. Zhang, X.; Sugano, Y.; Bulling, A. Evaluation of appearance-based methods and implications for gaze-based applications. In Proceedings of the Proceedings of the 2019 CHI conference on human factors in computing systems, 2019, pp. 1–13.
10. Whitehill, J.; Omlin, C.W. Haar features for FACS AU recognition. In Proceedings of the 7th international conference on automatic face and gesture recognition (FGR06). IEEE, 2006, pp. 5–pp.
11. Li, Y.; Xu, X.; Mu, N.; Chen, L. Eye-gaze tracking system by haar cascade classifier. In Proceedings of the 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA). IEEE, 2016, pp. 564–567.
12. Ngo, H.T.; Rakvic, R.N.; Broussard, R.P.; Ives, R.W. An FPGA-based design of a modular approach for integral images in a real-time face detection system. In Proceedings of the Mobile Multimedia/Image Processing, Security, and Applications 2009. SPIE, 2009, Vol. 7351, pp. 83–92.
13. Yazdani, A.; Safdari, R.; Golkar, A.; Niakan Kalhori, S.R. Words Prediction Based on N-Gram Model for Free-Text Entry in Electronic Health Records. Health Information Science and Systems, 2019, 7, 1–7.
14. Sherstinsky, A. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. Physica D: Nonlinear Phenomena, 2020, 404, 132306.
15. Zaki, M. Z. Revolutionising Translation Technology: A Comparative Study of Variant Transformer

Models–BERT, GPT and T5. Computer Science and Engineering–An International Journal, 2024, 14(3), 15-27.
16. Owens, M., O'Boyle, P., McMahon, J., Ming, J., & Smith, F. J. A comparison of human and statistical language model performance using missing-word tests. Language and Speech, 1997, 40(4), 377-389.
17. Bulling, A. DaRUS Eye-Gaze Writing Models. Available online: https://darus.uni-stuttgart.de/dataset.xhtml?persistentId=doi:10.18419/darus-3230&version=1.0 accessed on March 2025).
18. RoboFlow. Eyes Dataset. Available online: https://universe.roboflow.com/rethinkai2/eyes-mv4fm/dataset/1 (accessed on 1 March 2025).
19. RoboFlow. Dataset Pupil. Available online: https://universe.roboflow.com/politeknik-negeri-padang-yeiym/dataset-pupil (accessed on 1 March 2025).
20. RoboFlow. Pupil Detection Dataset. Available online: https://universe.roboflow.com/rocket-vhngd/pupil-detection-fqerx (accessed on 1 March 2025).
21. Wang, G.; Chen, Y.; An, P.; Hong, H.; Hu, J.; Huang, T. UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios. Sensors 2023, 23, 7190.
22. Redmon, J. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 2018.
23. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8759–8768.
24. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. Tood: Task-aligned one-stage object detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE Computer Society, 2021, pp. 3490–3499.
25. Bast, H., & Weber, I. Type less, find more: fast autocompletion search with a succinct index. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 364-371, August 2006.
26. Demir, A.; Yilmaz, F.; Kose, O. Early detection of skin cancer using deep learning architectures: resnet-and inception-v3. In *Proceedings of the 2019 Medical Technologies Congress (TIPTEKNO)*, IEEE, 2019, pp. 1–4.
27. RoboFlow. Pupil Detection Dataset. Available online: https://universe.roboflow.com/rocket-vhngd/pupil-detection-fqerx (accessed on March 2025).