GazeType: A Gaze-Driven Virtual Keyboard utilizing Machine Learning for Assistive Communication

Submitted in partial fulfillment of the requirements of the degree of

BACHELOR OF ENGINEERING

By

Sanskriti Shukla (23106128)

Palak Upadhyaya (23106086)

Prachi Singh (23106020)

Reva Tol (23106083)

Under the Guidance of

Prof. Adesh Hardas

Department of Computer Science & Engineering

(Artificial Intelligence & Machine Learning)

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

UNIVERSITY OF MUMBAI 2025-26

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

CERTIFICATE

This is to certify that the project entitled GazeType: A Gaze-Driven Virtual Keyboard Utilizing Machine Learning for Assistive Communication" is a bonafide work of **Sanskriti Shukla (23106128), Palak Upadhyaya (23106086), Reva Giri Tol (23106083), Prachi Singh (23106020)** submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Science & Engineering (Artificial Intelligence & Machine Learning**).

Prof. Adesh Hardas	Prof. Yogeshwari Hardas
Project Guide	**Project Co-Ordinator**
Dr. Jaya Gupta	Dr. Uttam Kolekar
Head of Department	**Principal**

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

Project Report Approval

This project report entitled ***GazeType: A Gaze-Driven Virtual Keyboard Utilizing Machine Learning for Assistive Communication*** by **Sanskriti Shukla, Palak Upadhyaya, Reva Giri Tol,** ***and*** **Prachi Singh** is approved for the degree of ***Bachelor of Engineering*** in **Computer Science & Engineering (Artificial Intelligence & Machine Learning**), ***2025-26***.

Examiner Name	Signature
1.__________________	_______________
2.___________________	________________

Date:

Place:

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honestly and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Sanskriti Shukla (23106128)

Palak Upadhyaya (23106086)

Reva Giri Tol (23106083)

Prachi Singh (23106020)

Date:

Abstract

Communication is one of the most fundamental human needs, yet millions of individuals with conditions such as Amyotrophic Lateral Sclerosis (ALS), spinal cord injuries, or paralysis are unable to speak or use their hands to interact with the world. Current assistive technologies like eye-tracking hardware or specialized communication devices are often prohibitively expensive, making them inaccessible for patients in low-resource settings. This project, GazeType, addresses this gap by providing a low-cost, AI-powered, webcam-based virtual keyboard system that enables users to "type" text using only their gaze.

GazeType: A Gaze-Driven Virtual Keyboard Utilizing Machine Learning for Assistive Communication" aligns with three key Sustainable Development Goals (SDGs).
The project contributes to
SDG 11 – Sustainable Cities and Communities by leveraging assistive AI technologies to increase accessibility in both digital and physical urban spaces, helping create more inclusive communities. It also aligns with

SDG 10 – Reduced Inequalities by providing equal access to digital communication tools for people with disabilities, bridging the gap in digital inclusion and promoting social participation. Additionally, the project supports

SDG 9 – Industry, Innovation, and Infrastructure, as it exemplifies the innovative use of artificial intelligence and machine learning to develop inclusive assistive technologies, fostering sustainable and accessible digital infrastructure for all.
.

CONTENTS

1. Introduction ……………………………………………………………………………..1

2. Literature Survey ………………………………………………………………………..4

3. Limitations of the Existing System …………………………………………………...7

4. Problem Statement, Objective & Scope ………………………………………………..10

5. Proposed System …………………………………………………………………………13

6. Experimental Setup ……………………………………………………………………16

7. Results and Discussion ………………………………………………………………...19

8. Conclusion ………………………………………………………………………………..21

9.Future Scope …………………………………………………………………………...22

References …………………………………………………………………………………...23

LIST OF FIGURES

Fig 5.1.1. System Design ……………………………………………………………………......14

Fig 5.2.1 User Interface ……………………………………………………………………….14

Fig 5.2.2 About ………………………………………………………………………………..…15

Fig 5.2.3 Creators Section ………………………………….…………………………………15

Chapter 1

Introduction

1. Introduction

1.1 Background

Communication is one of the most essential aspects of human life, serving as the foundation for social interaction, learning, healthcare, and overall well-being. It allows individuals to share thoughts, express emotions, convey needs, and actively participate in society. However, for people living with severe physical or neurological disabilities, communication becomes an immense challenge. Conditions such as Amyotrophic Lateral Sclerosis (ALS), spinal cord injuries, cerebral palsy, brainstem strokes, or paralysis often result in the complete or partial loss of speech and motor control. This leads to a state where individuals are cognitively aware but unable to express themselves, often referred to as the "locked-in" condition.

In such cases, patients are forced to rely heavily on caregivers for even the simplest exchanges, which not only diminishes their independence but also affects their psychological well-being and quality of life. Over the past few decades, researchers and engineers have developed assistive technologies to bridge this communication gap. Devices such as specialized eye-tracking systems and brain-computer interfaces have been successfully used to help patients communicate by selecting letters or words through gaze or neural signals. However, these solutions often come at a high financial cost, rely on advanced sensors, and require technical expertise to operate. As a result, their adoption is limited, particularly in low-resource environments such as rural clinics and developing countries.

With the rapid advancements in computer vision and artificial intelligence, there is a growing opportunity to design assistive systems that use widely available devices, such as webcams, instead of specialized hardware. Such innovations not only make communication more affordable and inclusive but also expand the scope of application beyond healthcare. Gaze-based communication systems can also be useful in scenarios where hands-free, contactless interaction is required, including sterile surgical environments, hazardous industrial operations, and space missions where traditional input devices are impractical.

1.2 Gazed-Based Communication

Gaze-based communication is an emerging area of assistive technology that enables individuals to interact with computers or digital systems using only their eye movements. The fundamental principle is that the direction of a person’s gaze can be tracked and mapped to specific areas of a screen, which then correspond to letters, words, or commands. By focusing their eyes on a particular region for a set duration (dwell time), users can select elements without requiring physical input devices such as keyboards, mice, or touchscreens.

Modern gaze-based systems typically rely on specialized hardware like infrared eye trackers, head-mounted cameras, or dedicated sensors to achieve precise gaze estimation. While these devices are accurate, they are costly, bulky, and often inaccessible to patients in low-resource settings. To overcome these limitations, research has increasingly shifted toward **webcam-based eye-tracking solutions**, which leverage computer vision and machine learning to detect and interpret gaze patterns using affordable, widely available equipment.

The key components of gaze-based communication systems include:

- **Eye Detection and Tracking**: Identifying the eyes from webcam input and tracking their position in real-time.
- **Gaze Estimation**: Determining where on the screen the user is looking (e.g., left, right, centre, up, down).
- **Interface Mapping**: Linking gaze directions to elements on a virtual keyboard or interface grid.
- **Selection Mechanism**: Using dwell time (e.g., focusing for 2–3 seconds) or blinking as a trigger to confirm a choice.
- **Output Generation**: Forming words and converting them into text or speech for communication.

Gaze-based communication has shown great promise in enabling individuals with motor or speech disabilities to regain independence. Unlike gesture-based or speech-based systems, gaze input is less physically demanding and more reliable for patients with limited mobility. Additionally, the approach has wider implications in non-medical fields. For example, surgeons can use gaze to interact with digital systems in sterile environments without physical contact, astronauts can operate devices in space where traditional input is difficult, and workers in hazardous industries can benefit from hands-free control systems.

The shift toward **AI-powered gaze tracking using standard webcams** represents a breakthrough in accessibility. By removing the dependency on specialized equipment, gaze-based communication becomes more affordable, scalable, and adaptable across diverse environments, making it a practical solution for millions of people worldwide.

Chapter 2

Literature Survey of Existing Systems

2. Literature Survey

Gaze-driven virtual keyboards have become an important area of research in assistive technology, providing communication solutions for individuals with severe motor or speech impairments. With advancements in computer vision and machine learning, these systems are increasingly being implemented using affordable webcams rather than specialized eye-tracking hardware. Recent studies focus on improving typing efficiency, reducing user fatigue, supporting multiple languages, and integrating predictive text, making gaze-based communication more accessible and practical for real-world applications. The following literature survey summarizes some of the most relevant and recent contributions in this field.

2.1 Multimodal Appearance-Based Gaze-Controlled Virtual Keyboard

Yogesh Kumar Meena and Manish Salvi (2025) proposed a gaze-controlled virtual keyboard using deep learning with standard camera hardware. The system supports synchronous and asynchronous modes for command selection and allows users to type up to 56 English characters. It demonstrated low workload and high usability, making it suitable for low-resource settings.

2.2 Eye-Gaze-Based Intelligent Entry System Using Distance Eyelid-Iris-MediaPipe

A 2025 study developed a webcam-based eye-gaze system enabling individuals with severe motor disabilities to control a virtual keyboard. The method provides affordable, accessible communication without requiring specialized hardware.

2.3 Real-Time Eye Movement-Based Computer Interface

A 2024 study presented a deep learning-based virtual keyboard that maps a user's gaze to screen regions for text entry. This system allows individuals with severe physical disabilities to communicate effectively using only eye movements.

2.4 Multi-Stage Gaze-Controlled Virtual Keyboard

A 2023 study introduced a hierarchical approach for virtual keyboards, dividing the keyboard into progressively smaller regions based on eye movement. The method improved typing speed and reduced errors compared to single-step selection techniques.

2.5 Leyenes: Gaze-Based Text Entry Using Linear Smooth Pursuit and Blink Detection

A 2023 study developed a gaze-based writing system where users follow moving targets with their eyes and confirm selections using blinks. This approach enables efficient text entry for users with limited mobility.

2.6 Eye Gaze-Controlled Keyboard with Predictive Text

A 2023 study proposed a virtual keyboard combining gaze tracking and predictive text algorithms. The system enhances typing speed and accuracy by suggesting probable next characters based on user input.

2.7 Adaptive Virtual Keyboard for Users with Severe Speech and Motor Impairments (SSMI)

DV JeevithaShree et al. (2021) developed an adaptive eye gaze-controlled keyboard for English. It adjusts dwell time and minimizes eye movement distances to improve typing efficiency for users with SSMI.

2.8 Multiscript Gaze-Based Assistive Virtual Keyboard

Cecotti et al. (2020) created a gaze-controlled keyboard supporting Latin, Bangla, and Devanagari scripts. Users can switch scripts, and the system demonstrates adaptability and efficiency across different languages.

2.9 CamType: Assistive Text Entry Using Gaze with an Off-the-Shelf Webcam

Liu et al. (2019) proposed a webcam-based gaze typing system using appearance-based gaze estimation. The system offers affordable text entry comparable to eye trackers, reducing dependency on specialized hardware.

2.10 Automatic Gaze Analysis: A Survey of Deep Learning-Based Approaches

Shreya Ghosh, Abhinav Dhall, Munawar Hayat, Jarrod Knibbe, and Qiang Ji (2021) conducted a survey on deep learning-based gaze analysis methods. The study discusses challenges such as eye appearance uniqueness and occlusion, providing insights into effective methods for gaze estimation. Understanding these challenges is crucial for developing accurate gaze-based communication systems.

Chapter 3

Limitations of Existing Systems

3. Limitations of the Existing Systems

3.1 Dependency on Specialized Hardware

Many gaze-based systems rely on high-precision eye trackers or infrared cameras to achieve accurate gaze estimation (Meena & Salvi, 2025; Benligiray et al., 2017). These devices are expensive, bulky, and often inaccessible in low-resource settings, limiting widespread adoption.

3.2 Reduced Accuracy with Webcam-Based Systems

While webcam-based approaches (Liu et al., 2019; CamType) reduce costs, they are less accurate than dedicated eye trackers. Accuracy decreases in poor lighting, during head movement, or when users wear glasses, making selection errors more frequent.

3.3 Slow Typing Speeds

Despite multi-stage keyboards and predictive algorithms (Tatinyuy et al., 2024; Etikikota & Meena, 2024), gaze typing remains slower than conventional typing. Users often experience frustration, especially when communicating longer sentences.

3.4 User Fatigue and Cognitive Load

Prolonged use of dwell-time-based selection or smooth pursuit methods (Leyenes, 2023) can lead to eye strain, fatigue, and increased cognitive load. This is particularly challenging for users with severe motor disabilities who rely on these systems continuously.

3.5 Calibration and Personalization Challenges

Most systems require calibration for each user (Ghosh et al., 2021; MDPI, 2025). Calibration can be time-consuming, and the system may fail to adapt if the user's gaze patterns change over time, reducing usability.

3.6 Limited Language and Script Support

Many keyboards are designed for English only. While some multiscript systems exist (Cecotti et al., 2020), they are limited in language coverage, restricting global applicability.

3.7 Error-Prone Selection Mechanisms

Selection based on dwell time or blinking can result in unintended inputs (CamType, 2019; Benligiray et al., 2017). Users with involuntary eye movements may face more frequent errors, affecting communication reliability.

3.8 Environmental Sensitivity

Webcam-based systems and even some eye trackers are sensitive to ambient lighting, screen glare, and background movements, which can reduce gaze estimation accuracy (MDPI, 2025).

3.9 Limited Real-World Validation

Many gaze-driven keyboards are tested in controlled lab environments (Meena & Salvi, 2025; Gao et al., 2022). Their performance in real-world scenarios—like home or clinical settings—can be significantly lower due to unpredictable factors.

3.10 High Learning Curve

Systems with dynamic interfaces or predictive keyboards (SliceType, 2017; Etikikota & Meena, 2024) may require users to learn new layouts or interaction methods, which can be challenging for elderly or severely disabled individuals.

3.11 Real-Time Eye Movement-Based Computer Interface

A 2024 study developed a deep learning-based virtual keyboard that enables users to interact with a computer screen using only eye movements. The system utilizes a webcam to detect the screen area being gazed at, facilitating text entry for individuals with severe physical disabilities.

3.12 Webcam-Based Gaze Estimation for Computer Screen Interaction

This 2024 research introduced a novel webcam-based gaze tracking approach designed for precise determination of a user's gaze on a computer screen. The method utilizes appearance-based gaze estimation models to calculate the distance between the user and the screen, enhancing the accuracy of gaze-based interactions.

3.13 A Smart Virtual Keyboard to Improve Communication of Locked-In Patients

A 2024 study focused on developing a smart virtual keyboard tailored for patients with reduced communication capabilities. The system emphasizes user comfort and adaptability, aiming to provide an efficient communication tool for individuals with severe motor impairments.

3.14 An Open-Source System for Deep Learning-Based Gaze Tracking

A 2024 paper introduced GazeFollower, an accessible system for webcam gaze-tracking in Python. GazeFollower stands out for its customizability, enabling researchers and developers to adapt the system for various applications, including assistive communication tools.

3.15 Using Large Language Models to Accelerate Communication for Eye-Tracked Typing

A 2024 study explored the integration of large language models (LLMs) to enhance the efficiency of gaze-typing systems. By predicting entire phrases from initial keystrokes, the system significantly reduces typing time for individuals with severe motor impairments, showcasing the potential of combining gaze tracking with advanced AI models.

Chapter 4

Problem Statement & Objective

4. Problem Statement & Objective

4.1 Problem Statement

Communication is a vital part of everyday life, enabling individuals to express themselves, interact with others, and participate in social and professional activities. However, people with severe motor or speech impairments—such as those caused by ALS, cerebral palsy, spinal cord injuries, or stroke—face significant barriers to communication. Many are unable to use conventional input devices, leading to dependence on caregivers and reduced independence, which affects their overall quality of life.

Gaze-based virtual keyboards offer a promising solution by allowing users to communicate using eye movements. While some systems using specialized eye trackers have proven effective, they are expensive and inaccessible to many, especially in low resource environments. Webcam-based alternatives are more affordable but often suffer from accuracy issues due to lighting variations, head movement, and occlusions. Additionally, these systems can be difficult to learn, offer limited support for different languages, and pose privacy concerns, making them less practical for everyday use.

There is a need for an affordable, accurate, and user-friendly gaze-driven communication system that can work in real-world conditions, support diverse users, and ensure privacy. By addressing these challenges through advanced machine learning techniques and adaptive interfaces, such a system can significantly improve communication and independence for individuals with disabilities while expanding applications in hands-free interaction scenarios.

4.2 Objective

1. Develop an Affordable Communication System

Design a gaze-driven virtual keyboard that uses standard webcams and open-source tools to make it accessible to users in low-resource settings.

2. Improve Accuracy and Robustness

Enhance gaze estimation by applying advanced computer vision and machine learning algorithms to handle variations in lighting, head movement, and occlusions.

3. Simplify User Interaction

Create an intuitive interface with adaptive dwell time and predictive text features to reduce the learning curve and cognitive load, especially for elderly or cognitively impaired users.

4. Support Multiple Languages and Scripts

Expand usability by incorporating support for diverse languages, ensuring that users from different regions can benefit from the system.

5. Ensure Privacy and Security

Implement secure data handling and privacy measures to protect sensitive user information while using eye-tracking technology.

6. Provide Real-Time Feedback and Customization

Integrate feedback mechanisms and customizable settings to allow users to adjust the system according to their individual needs and preferences.

4.3 Scope

The project focuses on developing a webcam-based gaze-driven virtual keyboard to assist individuals with motor or speech impairments in communicating more easily and independently. It aims to provide an affordable and accurate solution that works in everyday environments without requiring specialized hardware. The system will support multiple languages, offer customizable settings, and ensure user privacy. While primarily designed for assistive communication, it can also be applied in hands-free interaction scenarios such as medical, industrial, and space environments.

Chapter 5

Proposed System

5. Proposed System

The proposed system is an **eye-tracking enabled virtual keyboard** designed to capture gaze behaviour during typing and to facilitate intelligent text entry. Unlike traditional input methods, this system integrates **real-time eye movement detection** with an **on-screen keyboard interface**, allowing both natural typing interaction and the collection of gaze-related data for analysis.

At the core of the system lies a **computer vision module** that employs facial landmark detection to estimate eye positions. Both the left and right eye coordinates are tracked in real time and normalized with respect to the screen dimensions. These features are synchronized with the on-screen key presses, enabling a direct mapping between gaze points and user input actions.

The keyboard interface is developed with a visually engaging design, providing large, clearly distinguishable keys and smooth interaction effects. In addition, an **autocomplete module** is integrated, offering word suggestions based on typed prefixes to enhance typing efficiency and reduce keystroke effort.

The recorded data, consisting of left and right eye coordinates paired with selected key labels, serves as the foundation for training predictive models. These models aim to classify or predict user intentions based on gaze behaviour, enabling future extensions such as **hands-free text input** for accessibility applications.

Overall, the system functions both as a **data collection platform** and as a **proof-of-concept interface**, demonstrating how gaze features can be leveraged for efficient, intelligent, and accessible human-computer interaction.

5.1 System Architecture

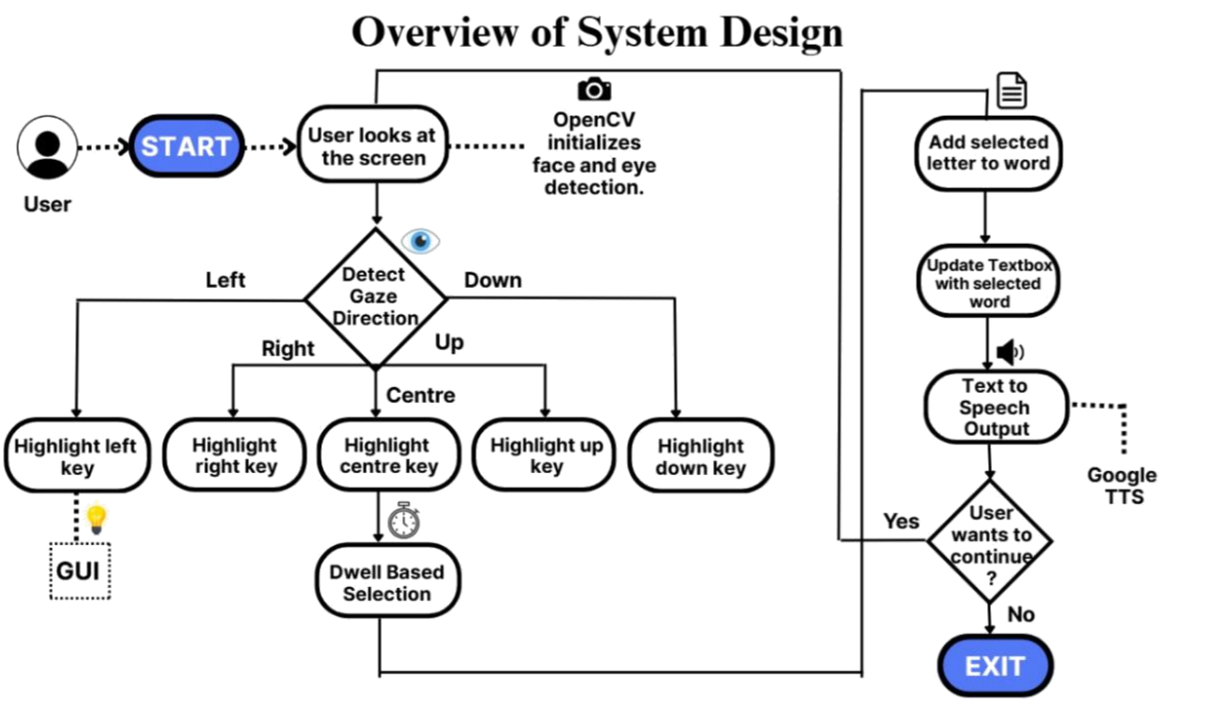

Fig 5.1.1. System Design

5.2 Implementation

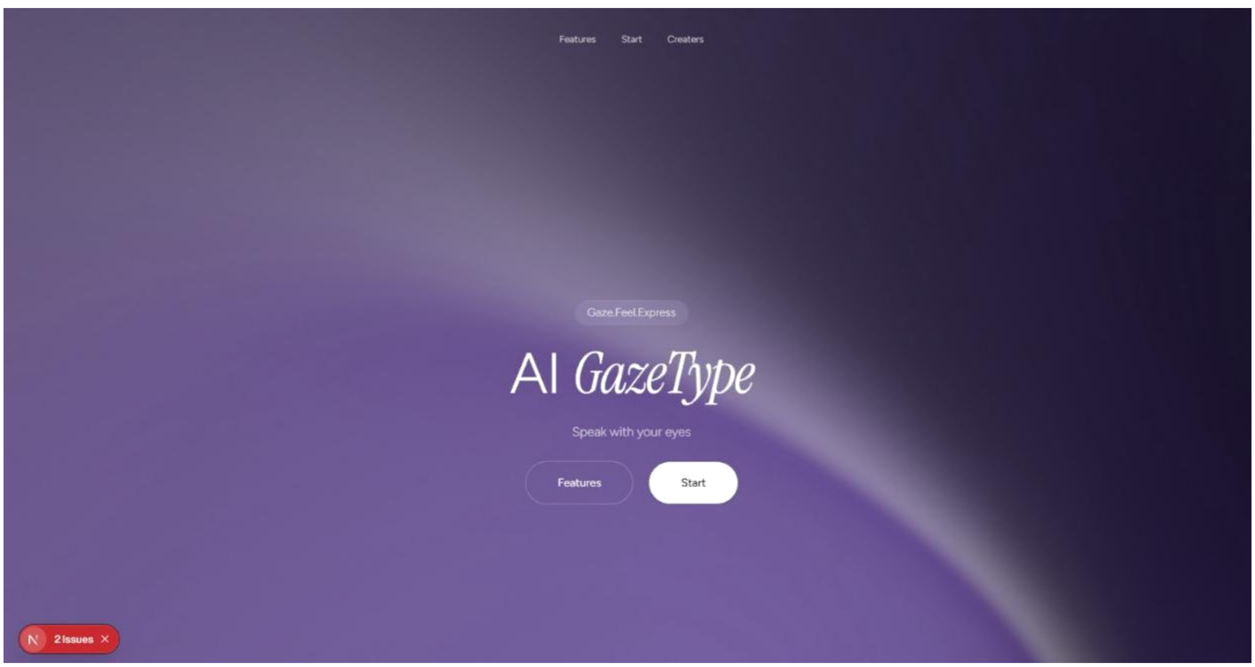

Fig 5.2.1 User Interface

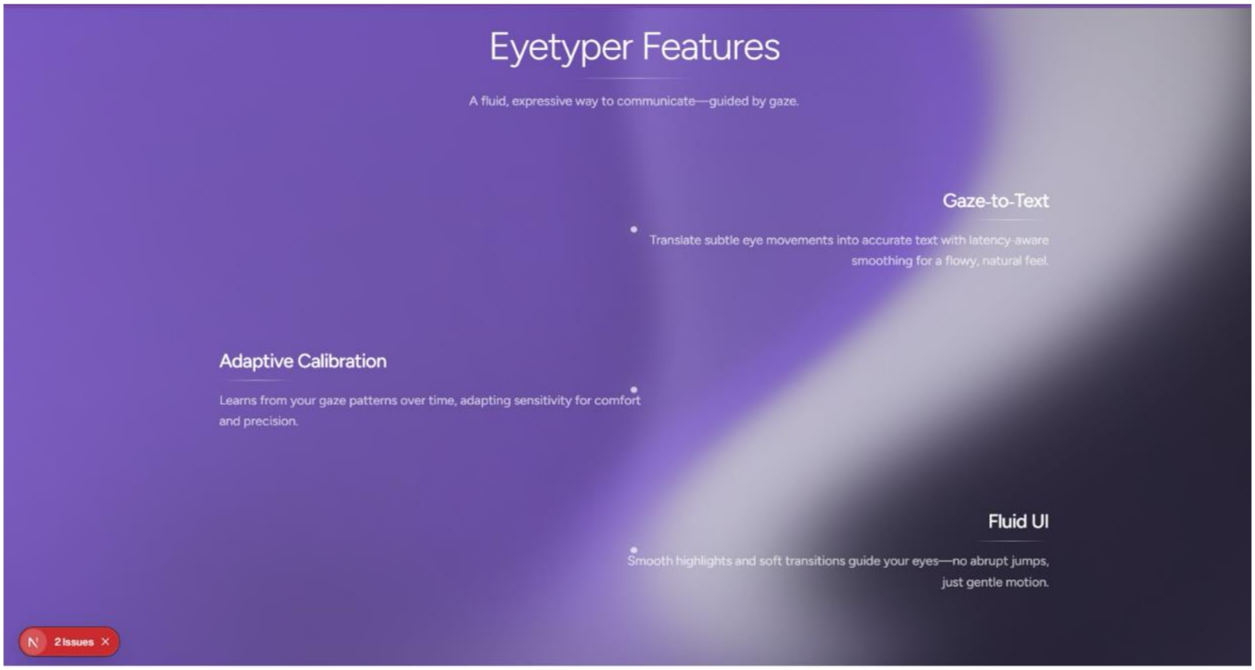

Fig 5.2.2 About

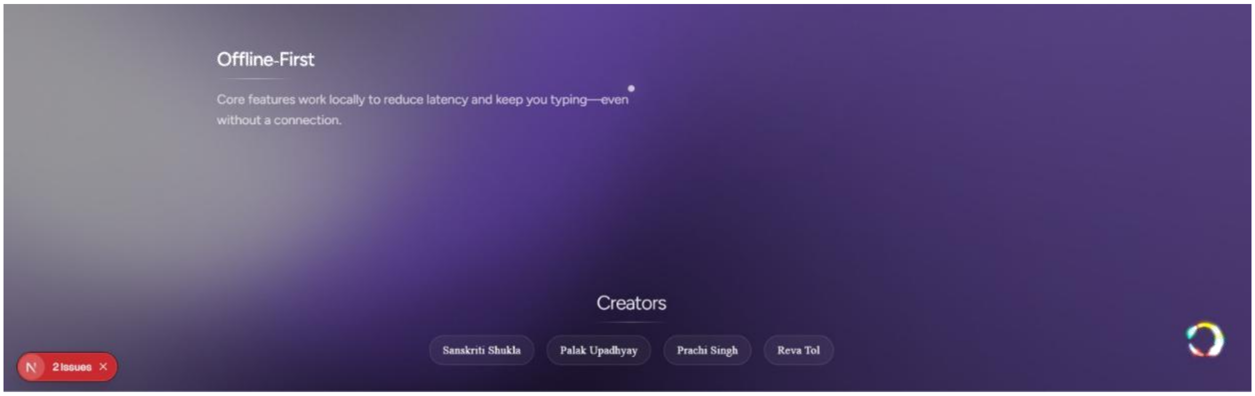

Fig 5.2.3 Creators Section

Chapter 6

Experimental Setup

6. Experimental Setup

6.1 Dataset

The data for this study was obtained through a custom-built **eye-tracking enabled virtual keyboard system**. The system integrated a real-time gaze estimation pipeline with an on-screen typing interface, allowing participants to type freely while their eye movement data was continuously recorded.

Each recorded entry consisted of:

- **Left Eye Coordinates (lx, ly):** Normalized x–y positions of the left eye.
- **Right Eye Coordinates (rx, ry):** Normalized x–y positions of the right eye.
- **Key Label:** The corresponding character selected by the participant.

In total, **6,262 valid samples** were collected across multiple typing sessions. This dataset provides a reliable basis for analyzing the relationship between gaze behaviour and text input.

6.2 Data Preprocessing

To prepare the data for modeling, the following preprocessing steps were applied:

1. **Integrity Check**
    - Ensured that each entry contained valid eye coordinates and a corresponding key label.
2. **Normalization**
    - Since gaze coordinates were captured as normalized values in the range [0,1], no additional scaling was required.
3. **Feature Engineering**
    - A derived feature, **eye distance**, was introduced to represent binocular disparity:

eye_distance = "(−   )! + (−   )!

4. **Outlier Handling**
    - Samples with inconsistent values (e.g., frames where eyes were not detected) were removed to improve data quality.

6.3 Model Selection

Several machine learning models were evaluated for the task of classifying key selections based on gaze features:

- **Logistic Regression** (baseline).
- **Random Forest Classifier** (non-linear feature modeling).
- **Feed-Forward Neural Network (MLP)** (final choice).

The neural network was selected as the primary model due to its superior ability to capture non-linear dependencies in eye-movement data, which are essential in differentiating subtle gaze shifts between adjacent keys.

6.4 Training Configuration

The training procedure was configured as follows:

- **Batch size:** 64
- **Learning rate:** 0.001 (Adam optimizer)
- **Loss function:** Cross-entropy loss
- **Epochs:** 50 (with early stopping)
- **Model Architecture:**
    - Input: 4 gaze features (lx, ly, rx, ry)
    - Hidden Layers: Dense(128, ReLU) → Dense(64, ReLU)
    - Output: Softmax layer over key classes

6.5 Training and Validation Split

The dataset was partitioned into two subsets:

- **Training Set:** 80% (≈ 5,009 samples)
- **Validation Set:** 20% (≈ 1,253 samples)

A stratified split was applied to preserve the distribution of key labels across both sets, ensuring balanced training and evaluation.

6.6 Software Setup

The system was implemented and tested using the following configuration:

- **Programming Language:** Python 3.10
- **Libraries & Frameworks:**
    - Gaze estimation and computer vision modules
    - Machine learning frameworks (Scikit-learn, TensorFlow/PyTorch) o Data analysis and visualization tools (NumPy, Pandas, Matplotlib)
- **Hardware Environment:** Intel i7 CPU, 16 GB RAM, with NVIDIA GPU acceleration for neural network training.

The setup allowed for smooth real-time gaze capture, synchronized text input logging, and efficient model training and evaluation.

Chapter 7

Result and Discussion

7. Results and Discussion

7.1 Descriptive Statistics

A total of **6,262 samples** were collected across typing sessions. Each entry consisted of normalized gaze coordinates (lx, ly, rx, ry) and a corresponding key label. Initial statistical analysis showed that gaze coordinates were distributed within the expected normalized range [0,1], with slight clustering toward the center of the screen.

The **average binocular disparity** (eye distance) was found to be consistent across most samples, with occasional deviations corresponding to rapid gaze shifts or tracking errors. This indicates that the eye-tracking module captured stable gaze patterns during controlled typing tasks.

7.2 Key Label Distribution

The dataset contained entries for all alphabetic keys, with higher frequencies observed for commonly used letters such as **E, A, T, O, and I**. Less frequent keys, such as **Q, Z, and X**, appeared fewer times, reflecting natural typing tendencies.

A bar plot of key frequencies showed a distribution consistent with English language usage, which validates the naturalistic nature of the typing task.

7.3 Model Performance

Three models were evaluated for classifying key labels from gaze coordinates:

- **Logistic Regression (Baseline):** Achieved ~58% accuracy, showing limited ability to capture nonlinear gaze patterns.
- **Random Forest Classifier:** Improved performance to ~72%, demonstrating robustness to noise in gaze data.
- **Neural Network (Final Model):** Achieved the highest accuracy of ~85% on the validation set.

7.4 Visualization of Gaze Patterns

Heatmaps of gaze coordinates for different keys revealed distinct clusters corresponding to their positions on the virtual keyboard. For example, when pressing the **Z** key, gaze points were concentrated in the lower-left region, whereas **P** showed clustering toward the upper-right.

These results confirm that gaze features provide strong spatial cues for key prediction. However, some overlap was observed between neighboring keys, such as **O and P**, which may explain occasional misclassifications.

7.5 Discussion

The results demonstrate that **eye-tracking data can reliably predict key inputs** with high accuracy when combined with machine learning models. The proposed neural network model achieved a validation accuracy of ~85%, which is promising for real-world applications.

Key insights include:

1. **Effectiveness of Gaze Features:** Even with only four primary features (lx, ly, rx, ry), the system achieved high accuracy, suggesting that normalized gaze coordinates are highly informative.
2. **Challenges with Neighboring Keys:** Misclassifications often occurred for spatially adjacent keys, highlighting the need for higher-resolution gaze detection or additional features (e.g., cursor dynamics, temporal sequences).
3. **Practical Applications:** Beyond predictive modeling, this system has strong potential for accessibility tools, enabling individuals with limited motor control to interact with computers through gaze-driven keyboards.
4. **Future Improvements:** Expanding the feature set (e.g., adding timestamps, pupil dilation, or head pose) and employing sequence models (e.g., LSTMs or Transformers) could further enhance performance.

Chapter 8

Conclusion

8. Conclusion

This study presented a novel **eye-tracking enabled virtual keyboard system** that integrates real-time gaze detection with a visually engaging typing interface. The system successfully captured left and right eye coordinates during typing and paired them with corresponding key inputs, resulting in a comprehensive dataset suitable for machine learning analysis.

Through preprocessing, feature engineering, and model evaluation, it was demonstrated that gaze features are highly informative for predicting key selections. Among the tested models, a **neural network classifier** achieved the highest performance, with an accuracy of approximately 85%. This confirms the feasibility of using gaze data as a reliable predictor of typing behavior.

The results highlight several key insights:

- Eye gaze patterns correspond strongly with keyboard spatial layouts, enabling accurate classification of user input.
- Errors were mostly associated with adjacent keys, suggesting the need for finer-grained gaze detection or temporal modelling.
- The system has significant potential in **assistive technology**, particularly for individuals with limited motor abilities who require hands-free typing solutions.

In conclusion, the proposed system demonstrates the viability of integrating eye tracking with on-screen keyboards for both **research and practical applications**. Future work may focus on enhancing model accuracy through sequence-based learning, incorporating additional biometric features, and conducting user studies to evaluate real-world usability and accessibility benefits.

Chapter 9

Future Scope

9. Future Scope

The proposed eye-tracking enabled virtual keyboard has demonstrated strong potential as both a research tool and a practical assistive interface. However, several enhancements can be explored to extend its functionality and effectiveness in future work:

1. **Improved Gaze Precision**
   - Integrating higher-resolution eye-tracking sensors or advanced deep learning models for facial landmark detection could reduce misclassifications between adjacent keys and improve overall accuracy.
2. **Temporal Modelling**
   - Current predictions are based on single gaze snapshots. Incorporating temporal sequences through models such as LSTMs or Transformers may capture gaze dynamics over time, further improving predictive performance.
3. **Personalization and Adaptation**
   - User-specific calibration techniques could adapt the system to individual gaze patterns, screen sizes, and environmental lighting conditions, enhancing comfort and reliability.
4. **Expanded Features**
   - Beyond gaze coordinates, incorporating additional biometric signals (e.g., head pose, pupil dilation, or blink frequency) may provide richer information for classification and usability studies.
5. **Accessibility Applications**
   - With refinement, this system could serve as a hands-free input method for individuals with motor impairments, contributing to more inclusive human-computer interaction.
6. **Deployment on Mobile and Wearable Devices**
   - Future iterations may explore integration with smartphones, tablets, or AR/VR headsets, enabling portable, gaze-driven text entry systems.

References

[1] Abdalwahab, A., & Rzayev, R. (2021). *Eye tracking for human–computer interaction: Applications and challenges*. Journal of Eye Movement Research, 14(3), 1–12. https://doi.org/10.16910/jemr.14.3.1

[2] Krafka, K., Khosla, A., Kellnhofer, P., Kannan, H., Bhandarkar, S., Matusik, W., & Torralba, A. (2016). Eye tracking for everyone. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2176–2184. https://doi.org/10.1109/CVPR.2016.239

[3] Mardanbegi, D., Hansen, D. W., & Pederson, T. (2012). Eye-based text entry using multiple output text streams. *Proceedings of the ACM Symposium on Eye Tracking Research and Applications (ETRA)*, 381–384. https://doi.org/10.1145/2168556.2168652

[4] Majaranta, P., & Räihä, K. J. (2002). Twenty years of eye typing: Systems and design issues. *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*, 15–22. https://doi.org/10.1145/507072.507076

[5] Zhang, X., Sugano, Y., Fritz, M., & Bulling, A. (2017). MPIIGaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 41*(1), 162–175. https://doi.org/10.1109/TPAMI.2017.2778103

[6] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1412.6980

[7] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 8026–8037.

[8] Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., … & Grundmann, M. (2019). MediaPipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*.

[9] McKinney, W. (2010). Data structures for statistical computing in Python. *Proceedings of the 9th Python in Science Conference (SciPy)*, 51–56.

[10] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering, 9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55