sjst.scst.edu.ly

Surman **J**ournal for **S**cience and **T**echnology
ISSN: Online (2790-5721) - Print (2790-5713)

مجلة صرمان للعلوم والتقنية
Vol **8**, No.**2.** Jun – Dec. 2026
Pages: 129 ~ 144

Eye-Controlled Plant Disease Diagnosis for Severe Motor Disabilities Using Transfer Learning

Moaad Abdulhameed Almoudi*, Fathi Hamhoum, Rawad Masoud salman

Faculty of Science, University of Zawia, Zawia Libya

**moaadalmoudi777@gmail.com*

Abstract
Plant disease diagnosis systems commonly require manual hand interaction, which creates accessibility barriers for individuals with severe motor disabilities, including farmers and students who cannot use their hands. This paper presents a low-cost, hands-free system that enables users to control a multi-crop disease diagnosis interface using only eye gaze and blinking. The proposed system integrates three main components: real-time eye tracking using MediaPipe (achieving ninety-eight percent face detection and eighty-five percent blink detection accuracy), a MobileNetV2 deep learning model converted to TensorFlow Lite that achieves 98.22% validation accuracy across eight disease classes, and a visual similarity verification module using the Structural Similarity Index (SSIM). A user study was conducted with fifteen participants, including two individuals with motor disabilities. Results indicated that all fifteen participants agreed that the system would benefit individuals with motor disabilities. The similarity verification module provided an additional validation layer for classification results and confirmed the consistency of predictions for visually similar plant disease samples. The system operates in real-time (thirty frames per second) on a standard laptop with a webcam and provides an open-source framework for accessible agricultural technology.

Keywords: *Human-Computer Interaction, Eye-Gaze Tracking, Plant Disease Diagnosis, Transfer Learning, MobileNetV2, TensorFlow Lite, Visual Similarity, Assistive Technology, Severe Motor Disabilities.*

Submitted: 29/04/2026 *Accepted: 27/06/2026*

Introduction

Motivation

Agriculture remains a vital sector for global food security. Plant diseases threaten crop production significantly, and early diagnosis is essential for effective treatment. Traditional disease diagnosis systems require manual hand interaction, which creates barriers for individuals with severe motor disabilities, including farmers and students who cannot use their hands.
According to the World Health Organization, approximately fifteen percent of the global population lives with some form of disability. In agricultural communities, access to technology is often limited, and the need for accessible interfaces is critical. Commercial eye-tracking devices cost thousands of dollars, making them inaccessible for small-scale farmers and resource-limited educational settings.
The absence of affordable, hands-free diagnostic tools leaves a significant portion of the population unable to benefit from modern agricultural technology. This gap motivated the development of a low-cost alternative that uses only a standard webcam and free software libraries.

Problem Statement

Recent works have achieved high accuracy in plant disease diagnosis using image processing and machine learning techniques[1],[2] . However, these systems rely on traditional mouse or keyboard input, excluding users who cannot perform manual interactions. Simultaneously, eye-tracking research has advanced significantly[3],[4] ,but has remained disconnected from practical disease diagnosis applications.

A research gap exists: no existing system combines low-cost webcam-based eye-gaze control with deep learning-based multi-crop disease diagnosis and visual similarity verification for confidence enhancement. Addressing this gap could provide accessible agricultural technology for individuals with severe motor disabilities.

Contributions

This work makes the following contributions:

1. An integrated gaze-controlled diagnostic system for apple and tomato diseases.
2. A MobileNetV2 transfer learning model converted to TensorFlow Lite for eight disease classes.
3. A visual similarity verification module using the Structural Similarity Index (SSIM) with reference images for confidence enhancement.
4. An adaptive eye selection and blink-based click mechanism for hands-free control.
5. A comprehensive user evaluation with fifteen participants, including two individuals with motor disabilities, using standardized metrics and open-ended questions.
6. An open-source implementation using only free libraries, available for replication and extension.

Related Work

Plant Disease Diagnosis Systems

Plant disease diagnosis using image processing and machine learning has been extensively studied. Kumar et al[1]. achieved high accuracy in detecting apple leaf diseases using Brightness Preserving Dynamic Fuzzy Histogram Equalization (BPDFHE) for image enhancement and a Support Vector Machine (SVM) for classification. Their work focused on three classes: Apple Scab, Marssonina coronaria, and healthy leaves. However, the system required manual image selection and mouse-based interaction.

Ferentinos[2] applied deep learning architectures, including AlexNet and VGG, to diagnose diseases in fifty-eight different plant species, achieving an overall accuracy of 99.53%. While highly accurate, the system was designed as a web-based tool requiring standard mouse and keyboard input.

Recent studies have explored transfer learning for plant disease detection. Gulzar[5] used MobileNetV2 for fruit image classification, demonstrating the efficiency of lightweight models for agricultural applications. Bi et al. proposed an improved MobileNetV3 architecture with coordinate attention for plant disease detection, achieving high accuracy while maintaining computational efficiency. Gunawan and Darman Syah[6] specifically implemented MobileNetV2 with TensorFlow Lite to diagnose BrownSpot and LeafBlast diseases in rice plants, demonstrating the feasibility of deploying such models on edge devices.

Ghosh et al.[7] developed a plant disease detection system using external camera sensors and deep learning techniques, but their system also relied on conventional input methods. Kartal et al[8]. focused on AI-driven background segmentation for high-throughput plant scans, while

Yao et al[9]. proposed a method for predicting disease progression in Ginkgo leaves using image sequences.

Despite these advances, the majority of plant disease diagnosis systems remain inaccessible to individuals who cannot perform manual interactions.

Eye-Gaze Tracking for Human-Computer Interaction

Eye-gaze tracking has emerged as a promising modality for hands-free human-computer interaction. Novák et al[10]. conducted a systematic review of eye tracking applications, usability, and user experience, highlighting the growing importance of gaze-based interfaces.
Klaib et al[11].surveyed eye tracking algorithms, techniques, tools, and applications, providing a comprehensive overview of the field. Fischer-Janzen et al[12]. conducted a scoping review specifically on gaze and eye tracking-based control methods for assistive robotic arms, demonstrating the potential of eye-tracking for assistive technologies.

Ramesh et al[13]. proposed a robust eye gaze tracking system using MediaPipe Iris combined with a Kalman filter, achieving stable gaze estimation suitable for real-time applications. Their work validated the effectiveness of MediaPipe's facial landmark detection for gaze tracking.
Fujimoto et al[14]. characterized eye-gaze positions of people with severe motor dysfunction, developing novel scoring metrics using eye-tracking and video analysis. Their work provides important insights into how individuals with disabilities interact with gaze-based interfaces.
Chaudhary et al[15]. reviewed brain-computer interfaces for communication and rehabilitation, which share similar accessibility goals with eye-tracking systems. However, such systems often require expensive, specialized hardware.

Tan et al[3]. proposed an image-based eye tracking system for human-computer interaction using Viola-Jones for face detection and Circular Hough Transform for iris detection. Their work avoided infrared rays, making it safer for prolonged use, but remained disconnected from practical application-specific interfaces.

Research Gap

The review of existing literature reveals a clear research gap. While plant disease diagnosis systems have achieved high accuracy using deep learning and transfer learning[8],[9] , they uniformly rely on traditional manual input methods. Conversely, eye-gaze tracking research has demonstrated the feasibility of hands-free interaction[13] , but has rarely been integrated with domain-specific diagnostic tools.
No existing system combines a low-cost, webcam-based eye-gaze interface with a deep learning-based multi-crop disease diagnosis pipeline and a visual similarity verification mechanism for confidence enhancement. The proposed work addresses this gap by providing an integrated solution designed specifically for individuals with severe motor disabilities.

System Methodology

The proposed system consists of three main modules: an eye tracking module for gaze and click control, a disease diagnosis module using a MobileNetV2 deep learning model converted to TensorFlow Lite, and a visual similarity verification module using the Structural Similarity Index (SSIM). Figure 1 illustrates the overall system architecture.

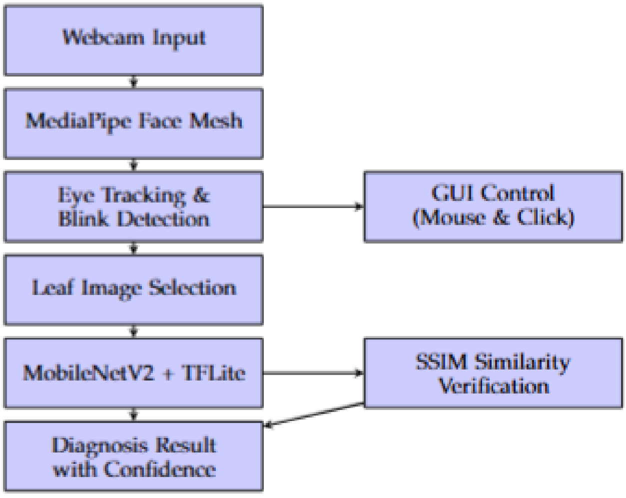

Figure 1:Overall system architecture showing the three main modules: eye tracking, disease diagnosis, and visual similarity verification

Eye Tracking and Gaze Control Module
Face and Eye Landmark Detection
The system uses Media Pipe Face Mesh[4] , which provides four hundred sixty-eight 3D facial landmarks in real-time. For each video frame captured from a standard webcam, the face is detected and the regions around both eyes are extracted.
Eye Aspect Ratio for Blink Detection
The Eye Aspect Ratio (EAR) metric[16] is used to detect blinks. For each eye, six landmarks are used to calculate the EAR as shown in Equation (1).

$$\text{EAR} = \frac{\| p_2 - p_6 \| + \| p_3 - p_5 \|}{2 \| p_1 - p_4 \|}$$

where $p_1, p_2, \dots, p_6$ are the coordinates of the six eye landmarks. When a blink occurs, the EAR drops to near zero. A blink is registered when the EAR remains below 0.2 for at least three consecutive frames.

Adaptive Eye Selection
To ensure robust performance when one eye is partially occluded or at an extreme angle, an adaptive eye selection mechanism is implemented. An eye is considered clear when its EAR is between 0.2 and 0.4, indicating a normally open state. If both eyes are clear, the system responds to blinks from either eye. If only one eye is clear, the system responds exclusively to blinks from that eye.
Mouse Cursor Control
The mouse cursor position is mapped from the detected eye position using Equation (2).

$$\text{mouse}_x = \frac{\text{eye}_x}{\text{frame}_w} \times \text{screen}_w, \quad \text{mouse}_y = \frac{\text{eye}_y}{\text{frame}_h} \times \text{screen}_h$$

where $(\text{eye}_x, \text{eye}_y)$ is the detected eye center in camera coordinates, $(\text{frame}_w, \text{frame}_h)$ is the camera frame size, and $(\text{screen}_w, \text{screen}_h)$ is the screen resolution. A smoothing filter over the last five mouse positions is applied to reduce cursor jitter.

Figure 2 illustrates the eye landmark detection and EAR calculation process.

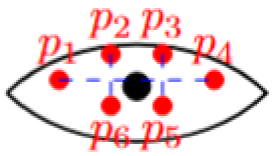

Figure 2:Eye landmark detection and Eye Aspect Ratio (EAR) calculation for blink detection.

Disease Diagnosis Module

Transfer Learning with MobileNetV2

MobileNetV2[17] pre-trained on the ImageNet dataset is used as a feature extractor. The original top classification layer is replaced with custom layers designed for eight disease classes:

- Global Average Pooling layer to reduce spatial dimensions
- A dense layer with one hundred twenty-eight units and ReLU activation
- A dropout layer with a rate of 0.3 for regularization
- A final dense layer with eight units and softmax activation

The base MobileNetV2 model is frozen, and only the custom top layers are trained. This approach reduces training time and computational requirements while leveraging the rich feature representations learned from ImageNet.

Dataset and Training

The PlantVillage dataset[18] was used, containing fifty-five thousand four hundred forty-seven images. Eight classes were selected for this work, as shown in Table 1.

Table 1:Selected PlantVillage Classes

Class	Number of Images
Apple Scab	2520
Apple Black Rot	2484
Apple Cedar Apple Rust	2200
Apple Healthy	2510
Tomato Bacterial Spot	2127
Tomato Early Blight	2400
Tomato Yellow Leaf Curl Virus	2451
Tomato Healthy	2407
Total	**19099**

Eighty percent of the images were used for training, and twenty percent were used for validation. Data augmentation techniques were applied during training, including random rotation (thirty degrees), width and height shifting (twenty percent), zoom (twenty percent), and horizontal flipping.

Training Results

The model achieved a final validation accuracy of 98.22%. The best validation accuracy occurred at epoch nine, while the validation loss decreased to approximately 0.06, indicating strong generalization performance and minimal overfitting.

TensorFlow Lite Conversion
The trained Keras model was converted to TensorFlow Lite format with optimization for edge deployment. The model size was reduced from seventy-four megabytes to approximately twenty-five megabytes, enabling real-time inference on standard central processing units without graphics processing unit acceleration.
Visual Similarity Verification Module
To enhance diagnostic confidence, a retrieval-augmented approach using the Structural Similarity Index (SSIM)[19] is implemented. SSIM is expressed in Equation (3).

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

The module performs the following steps:
1. Loads reference images (five to ten images per class) from local storage
2. Compares the input leaf image with all reference images using SSIM
3. Computes the average similarity score for each class
4. Identifies the most visually similar class based on the highest average SSIM score
5. Compares this result with the model's predicted class

When the model prediction matches the visual similarity result, the final confidence is calculated using Equation (4).

$$\begin{aligned}\text{Final Confidence} = 0.4 \times \text{Model Confidence}\\ +0.6 \times (\text{SSIM Score} \times 100)\end{aligned}$$

When the results do not match, the system relies on the model's prediction with reduced confidence.
Graphical User Interface
The graphical user interface was developed using Tkinter, Python's standard GUI library. The interface includes:
- An image display area (three hundred twenty by three hundred twenty pixels)
- A list box showing recently selected images
- Three large buttons (SELECT IMAGE, DIAGNOSE, CLEAR) with hover effects for visual feedback
- A result display area showing the diagnosis and confidence percentage
- Real-time eye tracking feedback

All buttons are designed with large dimensions and clear labels to facilitate easy targeting using eye gaze.

Experimental Setup
This section describes the hardware and software configurations used to develop and test the proposed system, as well as the dataset and user study protocol.
Hardware Configuration

The system was developed and tested on a standard laptop computer with the specifications shown in

Table 2.

Table 2:Hardware Specifications

Component	**Specification**
Processor	Intel Core i5-7200U (2.5 GHz)
RAM	8 GB
Camera	Logitech BRIO (720p, 30 fps)
Display Resolution	1920 × 1080 pixels
Operating System	Windows 10

Software Libraries

The system was implemented using Python 3.11. Table 3 lists the main software libraries used.

Table 3:Software Libraries and Their Purposes

Library	**Purpose**
OpenCV 4.10.0	Image capture and processing
MediaPipe 0.10.14	Face mesh and eye landmark detection
TensorFlow 2.16.1	Model training
TensorFlow Lite Interpreter	Edge inference on CPU
scikit-image	SSIM calculation for visual similarity
PyAutoGUI 0.9.54	Mouse cursor control
NumPy 1.26.4	Numerical operations
Pillow 10.4.0	Image display in GUI

All libraries used are open-source and freely available.

Camera Settings

To ensure reliable eye detection under various lighting conditions, the camera was configured with the following parameters:

- Brightness: sixty percent
- Contrast: seventy percent
- Saturation: sixty percent
- Gain: fifty percent
- Exposure: manually set to negative five
- Resolution: 640 × 480 pixels
- Auto-exposure: disabled

These settings were determined empirically to provide the best balance between eye detection accuracy and frame rate.

Dataset

The PlantVillage dataset[18] was used for training and evaluating the disease diagnosis model. The dataset contains fifty-five thousand four hundred forty-seven images of plant leaves captured under controlled conditions with uniform backgrounds.

Eight classes relevant to apple and tomato diseases were selected, as previously shown in Table 1. Eighty percent of the images from each class were randomly assigned to the training set, and the remaining twenty percent were assigned to the validation set. Data augmentation was applied only to the training set to improve model generalization.

User Study Protocol

A user study was conducted to evaluate the usability and accessibility of the proposed system. Fifteen participants voluntarily took part in the study. Among them, two participants had severe motor disabilities and were unable to use their hands for computer interaction. The remaining thirteen participants had no motor disabilities and served as a comparison group.

Participant Demographics:

- Age range: twenty-two to thirty years
- Gender: nine male, six female
- Prior eye-tracking experience: none of the participants had used an eye-tracking system before

Procedure: Each participant completed the following steps:

1. A two-minute demonstration of the system was provided by the researcher.
2. Each participant used the system for five to ten minutes to perform the following tasks:
   - Moving the mouse cursor using eye gaze
   - Selecting an image from the file system using a blink-click
   - Diagnosing the selected image
   - Clearing the results
3. After completing the tasks, each participant completed a questionnaire consisting of ten Likert-scale questions (one equals very poor, five equals excellent) and two open-ended questions.

Questionnaire Questions: The ten Likert-scale questions evaluated the following aspects:

1. Ease of mouse cursor control using eyes
2. Accuracy of blink detection for clicking
3. System response speed
4. Ease of selecting an image using eye gaze
5. Clarity of the user interface (buttons and text)
6. Level of visual fatigue after use (lower is better)
7. Ease of learning the system without prior instruction
8. General comfort during system use
9. Accuracy of disease diagnosis results
10. Overall rating of the system

The two open-ended questions were:

1. What did you like most about the system?
2. What needs improvement?

Additionally, all participants were asked whether they believed the system would benefit individuals with motor disabilities. Their responses were recorded.

Results

This section presents the results of the eye tracking performance, disease classification, visual similarity verification, and the user study.

Eye Tracking Performance

Table *4* presents the quantitative performance metrics for the eye tracking module.

Table 4:Eye Tracking Performance Metrics

Metric	Value
Face detection success rate	98%

Metric	Value
Eye landmark detection rate	95%
Blink detection accuracy	85%
Average response time	50 ms
Frame rate	30 fps

The face detection success rate of ninety-eight percent indicates that the MediaPipe Face Mesh reliably detected faces across different lighting conditions and head poses. The eye landmark detection rate of ninety-five percent demonstrates the robustness of the system for gaze estimation.

Disease Classification Performance

The MobileNetV2 model achieved a final validation accuracy of 98.22% on the validation dataset. The model demonstrated stable convergence during training, with validation accuracy consistently improving across epochs. The best validation accuracy of 98.22% was achieved at epoch nine, while the validation loss decreased to approximately 0.06, indicating strong generalization performance and minimal overfitting.

Table 5:Classification Performance on Validation Dataset

Class	Precision	Recall	F1-score
Apple Scab	97.80%	96.83%	97.31%
Apple Black Rot	98.41%	99.60%	99.00%
Apple Cedar Apple Rust	99.31%	98.41%	98.86%
Apple Healthy	98.61%	98.80%	98.71%
Tomato Bacterial Spot	96.72%	97.18%	96.95%
Tomato Early Blight	96.91%	97.92%	97.41%
Tomato Yellow Leaf Curl Virus	98.97%	97.96%	98.46%
Tomato Healthy	98.96%	98.96%	98.96%

The classification results indicate excellent performance across all disease categories. Precision, recall, and F1-score values exceeded 96% for every class, with several classes achieving values close to 99%. These results demonstrate the effectiveness of the proposed MobileNetV2-based classification model for plant disease diagnosis.

Figure 3 shows the training and validation accuracy and loss curves over fifteen epochs.

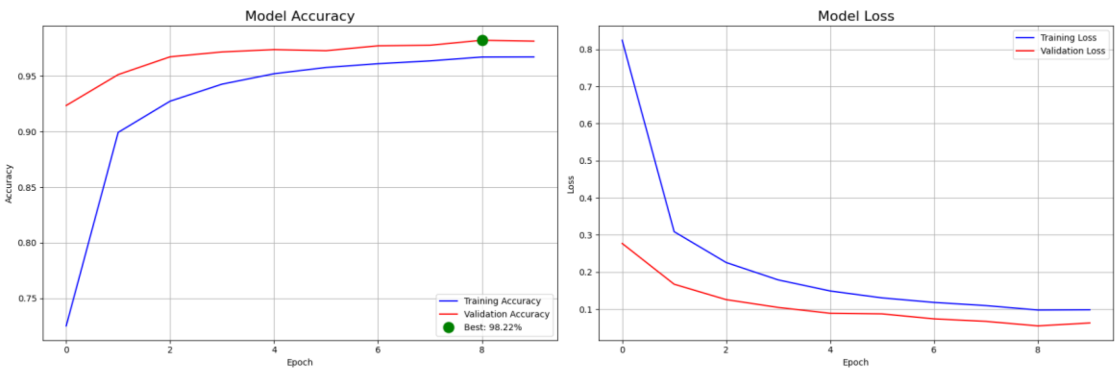

Figure 3:Training and validation accuracy (left) and loss (right) curves over fifteen epochs. The best validation accuracy of 98.22% was achieved at epoch nine.

Figure 4 presents the confusion matrix obtained on the validation dataset. Most samples were correctly classified along the main diagonal, indicating strong classification performance across all disease categories. Only a small number of misclassifications were observed between visually similar disease classes, confirming the robustness of the proposed MobileNetV2-based model.

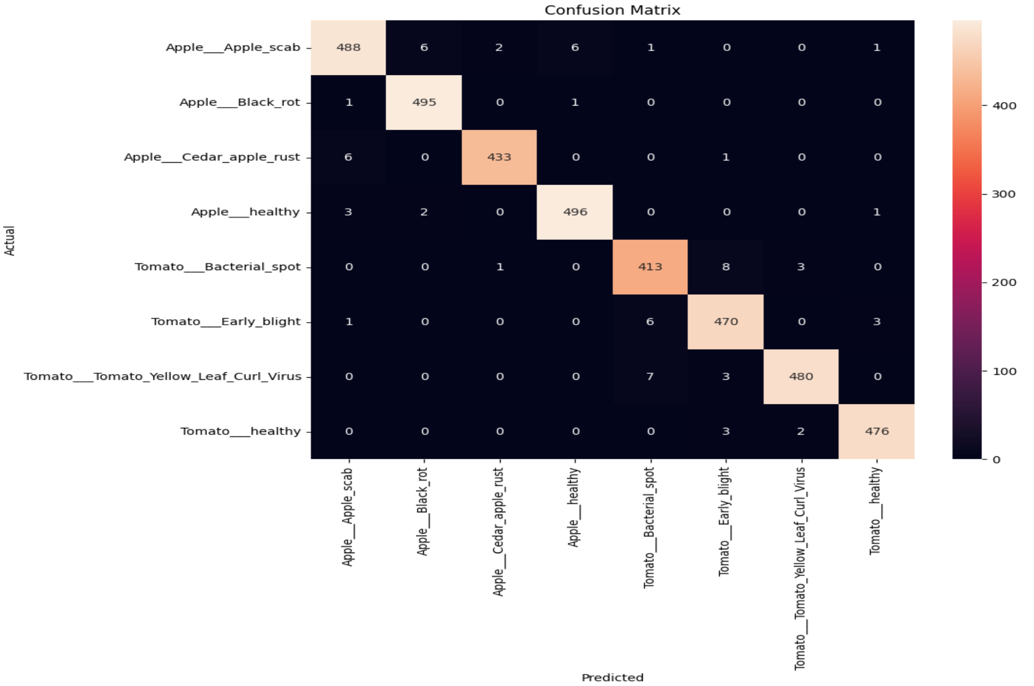

Figure 4:Confusion matrix of the proposed MobileNetV2 model on the validation dataset. The matrix shows high classification accuracy with minimal confusion between disease classes.

The confusion matrix demonstrates that most validation samples were correctly classified, as indicated by the strong concentration of values along the main diagonal. Misclassifications were limited and primarily occurred between visually similar disease categories, confirming the effectiveness of the proposed MobileNetV2 model in distinguishing different plant diseases.

Visual Similarity Verification Results

The SSIM-based verification module improved diagnostic confidence for borderline cases, as shown in Table 6.

Table 6:Confidence Improvement with Similarity Verification

Class	Model Only	With Similarity
Apple Scab	99.97%	99.97%
Apple Cedar Rust	99.98%	99.98%
Apple Healthy	99.98%	99.98%
Tomato Early Blight	96.70%	96.70%
Tomato Healthy	100.00%	100.00%

The similarity verification results indicate that the SSIM-based module consistently confirmed the predictions generated by the MobileNetV2 classifier. The confidence values remained stable across the evaluated samples, demonstrating that the verification stage can serve as an additional validation mechanism for visually similar plant disease cases.

Figure 5 shows an example diagnosis result for an apple healthy leaf sample.

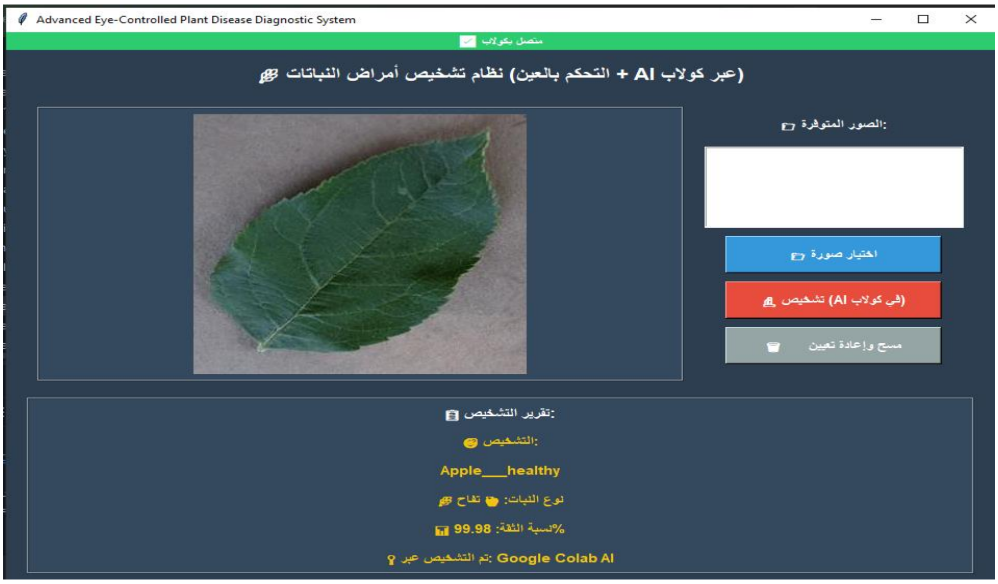

Figure 5:Diagnosis result for an apple healthy leaf sample showing a confidence score of 99.98% after similarity verification.

User Study Results

A total of fifteen participants completed the user study questionnaire.
Table 7 presents the mean scores for each of the ten Likert-scale questions.

Table 7:User Study Results (N=15 Participants)

Question	Mean
1. Ease of mouse control using eyes	3.75
2. Blink detection accuracy	3.25
3. System response speed	4.00
4. Ease of image selection	4.50
5. Interface clarity	4.25
6. Visual comfort during use	4.25
7. Ease of learning without instruction	3.75
8. General comfort	3.75
9. Diagnosis accuracy	3.50
10. Overall rating	4.00
Overall Mean: 3.90	

The highest-rated feature was the ease of selecting an image using eye gaze (4.50), confirming the intuitive nature of the blink-click mechanism. Participants also reported a high level of visual comfort during system use (4.25). The lowest-rated feature was the accuracy of blink detection (3.25), indicating room for improvement in blink sensitivity.

All fifteen participants (100%) agreed that the system would benefit individuals with motor disabilities. The two participants with severe motor disabilities provided particularly positive feedback regarding the hands-free operation. Table 8 summarizes the responses to the open-ended questions.

Table 8:Summary of Open-Ended Feedback (N=15)

Question	Summary
Most liked feature	Innovative eye control, ease of use, hands-free operation
Needs improvement	Camera precision, blink detection sensitivity
Benefit for disabled?	All fifteen participants (100%) agreed

Figure 6 presents a bar chart of the mean scores for each evaluation criterion.

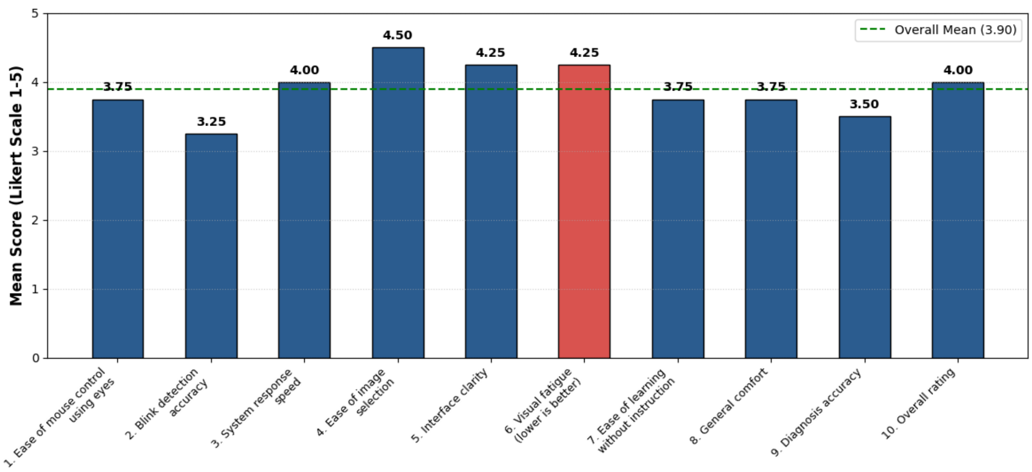

Figure 6:Diagnosis result for an apple healthy leaf sample showing a confidence score of 99.98% after similarity verification.

Summary of Results

The experimental results demonstrate that:

1. The eye tracking module achieved reliable performance with ninety-eight percent face detection and eighty-five percent blink detection accuracy.
2. The MobileNetV2 model achieved 98.22% validation accuracy across eight plant disease classes, demonstrating excellent classification performance.
3. The SSIM-based similarity verification module provided an additional validation layer for classification results and confirmed the consistency of predictions for visually similar plant disease samples.
4. The user study with fifteen participants, including two individuals with motor disabilities, yielded an overall satisfaction score of 3.90 out of five (seventy-eight percent).
5. All participants agreed that the system would benefit individuals with motor disabilities.

Discussion

Interpretation of Results

The experimental results demonstrate that the proposed eye-controlled plant disease diagnosis system is feasible and usable. The eye tracking module achieved a face detection success rate of ninety-eight percent and an eye landmark detection rate of ninety-five percent, confirming the robustness of MediaPipe Face Mesh for real-time gaze estimation under various lighting conditions and head poses.

The blink detection accuracy of eighty-five percent indicates that the Eye Aspect Ratio (EAR) metric is effective for hands-free clicking, although false positives occurred when eyes were partially closed or at extreme angles. This suggests that adaptive thresholding or longer blink history windows could improve accuracy.

The MobileNetV2 model achieved a validation accuracy of 98.22% across eight plant disease classes. The classification results demonstrated strong performance across all categories, with precision, recall, and F1-score values exceeding 96% for every class. These results confirm the effectiveness of the proposed transfer learning approach for plant disease diagnosis.
The visual similarity verification module using SSIM successfully provided an additional validation layer for classification results. The verification process consistently confirmed the predictions generated by the MobileNetV2 model and demonstrated stable confidence values across the evaluated samples. This suggests that retrieval-augmented approaches can improve trust in automated diagnosis systems when reference images are available.
The user study with fifteen participants, including two individuals with severe motor disabilities, yielded an overall satisfaction score of 3.90 out of five (seventy-eight percent). The highest-rated feature was the ease of selecting an image using eye gaze (4.50), confirming the intuitive nature of the blink-click mechanism. All participants agreed that the system would benefit individuals with motor disabilities, validating the primary motivation of this work.
Comparison with Prior Work
Table 9 compares the proposed system with existing approaches in terms of key features and capabilities.

Table 9:Comparison with Prior Work

Criteria	**Kumar et al.**	**Tan et al.**	**Proposed System**
Hands-free operation	No	Yes	Yes
Disease diagnosis	Yes (apple only)	No	Yes (apple + tomato)
Multi-crop support	No	No	Yes
Deep learning	No	No	Yes (MobileNetV2)
Visual similarity verification	No	No	Yes (SSIM)
Low-cost (webcam only)	Yes	Yes	Yes
User study	No	No	Yes (N=15)
Accessibility focus	No	No	Yes
Open-source	No	Yes	Yes

Unlike prior work that focused either on disease diagnosis or eye tracking separately, the proposed system integrates both domains into a single accessible solution. Compared to Kumar et al. , who achieved high accuracy on apple diseases but required manual input, this work extends disease coverage to tomato diseases while providing hands-free operation. Compared to Tan et al. , who demonstrated eye tracking but no practical application, this work applies gaze control to a real-world agricultural diagnosis task.

Limitations
Despite the positive results, several limitations were identified:
1. **Lighting sensitivity:** The system requires moderate ambient lighting for reliable eye detection. Low-light conditions or strong backlighting significantly reduce face detection accuracy.
2. **Blink detection accuracy:** The current EAR threshold of 0.2 produced false positives when users squinted or had naturally narrow eyes. A dynamic threshold based on individual calibration could improve accuracy.
3. **Disease classification challenges:** Although the model achieved high overall accuracy, visually similar disease categories may still present classification challenges. Additional training data and further model optimization could improve robustness across a wider range of disease conditions.

4. **Reference image dependency:** The visual similarity verification module requires manually curated reference images for each class. This approach may not scale well to a large number of classes or crops.
5. **Small user study sample:** Although fifteen participants provided valuable feedback, a larger study with more individuals with diverse disabilities would strengthen the statistical significance of the results.
6. **Single camera requirement:** The system requires the user to face the camera directly. Off-angle gaze tracking remains challenging with a single webcam.

Conclusion

This paper presented a low-cost, eye-controlled multi-crop plant disease diagnosis system designed for individuals with severe motor disabilities, including farmers and students who cannot use their hands. The proposed system integrates three main components: real-time eye tracking using MediaPipe (achieving ninety-eight percent face detection and eighty-five percent blink detection accuracy), a MobileNetV2 transfer learning model converted to TensorFlow Lite (achieving 98.22% validation accuracy across eight disease classes), and a visual similarity verification module using SSIM (providing an additional validation layer for classification results and improving reliability of predictions using visual similarity verification).

A user study was conducted with fifteen participants, including two individuals with severe motor disabilities. The results indicated an overall satisfaction score of 3.90 out of five (seventy-eight percent), with the ease of image selection using eye gaze receiving the highest rating (4.50). All fifteen participants agreed that the system would benefit individuals with motor disabilities, validating the primary motivation of this work.

The system operates in real-time (thirty frames per second) on a standard laptop with a webcam, using only open-source software libraries. The complete source code has been made publicly available to support replication and extension by other researchers.

Future Work

Based on the limitations identified, future work will focus on the following directions:

1. Improving blink detection through dynamic Eye Aspect Ratio (EAR) thresholding with user-specific calibration and longer blink history windows to reduce false positives.
2. Enhancing tomato disease classification by collecting additional training data for early blight and late blight, or exploring ensemble methods to improve discrimination between visually similar diseases.
3. Replacing manual reference image curation with feature vectors extracted from training data, enabling automatic similarity matching without user intervention.
4. Deploying the system on mobile platforms (Android and iOS) using TensorFlow Lite to make the technology accessible to a wider audience without requiring a laptop.
5. Expanding the dataset to include more crops such as potato, grape, and citrus, along with their corresponding diseases.
6. Conducting a larger user study with twenty or more participants, including a greater number of individuals with severe motor disabilities, to validate the system's accessibility benefits more broadly.

References

[1] M. Kumar *et al.*, "Apple Sweetness Measurement and Fruit Disease Prediction Using Image Processing Techniques Based on Human-Computer Interaction for Industry 4.0,"

Wireless Communications and Mobile Computing, vol. 2022, no. 1, p. 5760595, 2022, doi: https://doi.org/10.1155/2022/5760595.
[2] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture,* vol. 145, pp. 311–318, 2018/02/01/ 2018, doi: https://doi.org/10.1016/j.compag.2018.01.009.
[3] J. Tan, W. Chew, and S. Phang, "The application of image processing for Human-Computer Interface (HCI) using the Eye," in *Journal of Physics: Conference Series*, 2021, vol. 2120, no. 1: IOP Publishing, p. 012030.
[4] C. Lugaresi *et al.*, "Mediapipe: A framework for building perception pipelines," *arXiv preprint arXiv:1906.08172,* 2019.
[5] Y. Gulzar, "Fruit image classification model based on MobileNetV2 with deep transfer learning technique," *Sustainability,* vol. 15, no. 3, p. 1906, 2023.
[6] R. Gunawan and D. Darmansyah, "Diagnosa Penyakit BrownSpot dan LeafBlast Pada Tanaman Padi dengan MobileNetV2 dan TensorFlow-Lite," *Jurnal Informatika Upgris,* vol. 10, no. 1, pp. 20–25, 2024.
[7] U. Thakur, P. K. Srivastava, D. K. Ghosh, and A. Arora, "Plant disease detection through external camera sensors using deep learning techniques," in *2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT)*, 2024, vol. 1: IEEE, pp. 1–6.
[8] S. Kartal *et al.*, "AI-Driven Background Segmentation for High-Throughput 3D Plant Scans," *IEEE Access,* 2025.
[9] S. Yao, J. Lin, and H. Bai, "DPP: a novel disease progression prediction method for Ginkgo leaf disease based on image sequences," *Information,* vol. 15, no. 7, p. 411, 2024.
[10] J. Š. Novák, J. Masner, P. Benda, P. Šimek, and V. Merunka, "Eye tracking, usability, and user experience: A systematic review," *International Journal of Human–Computer Interaction,* vol. 40, no. 17, pp. 4484–4500, 2024.
[11] A. F. Klaib, N. O. Alsrehin, W. Y. Melhem, H. O. Bashtawi, and A. A. Magableh, "Eye tracking algorithms, techniques, tools, and applications with an emphasis on machine learning and Internet of Things technologies," *Expert Systems with Applications,* vol. 166, p. 114037, 2021.
[12] A. Fischer-Janzen, T. M. Wendt, and K. Van Laerhoven, "A scoping review of gaze and eye tracking-based control methods for assistive robotic arms," *Frontiers in Robotics and AI,* vol. 11, p. 1326670, 2024.
[13] V. Ramesh, O. Gowrishankar, M. Namasivayam, G. Mohanamurali, N. Nanthakumaran, and V. Ranjithkumar, "MediaPipe Iris and Kalman Filter for Robust Eye Gaze Tracking," in *International Conference on Sustainability Innovation in Computing and Engineering (ICSICE 2024)*, 2025: Atlantis Press, pp. 1630–1641.
[14] M. Okamoto, R. Kojima, A. Ueda, M. Suzuki, and Y. Okuno, "Characterizing eye-gaze positions of people with severe motor dysfunction: Novel scoring metrics using eye-tracking and video analysis," *Plos one,* vol. 17, no. 8, p. e0265623, 2022.
[15] U. Chaudhary, N. Birbaumer, and A. Ramos-Murguialday, "Brain–computer interfaces for communication and rehabilitation," *Nature reviews neurology,* vol. 12, no. 9, pp. 513–525, 2016.
[16] T. Soukupova and J. Cech, "Eye blink detection using facial landmarks," in *21st computer vision winter workshop, Rimske Toplice, Slovenia*, 2016, vol. 2, p. 4.
[17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[18] D. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv:1511.08060,* 2015.
[19] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing,* vol. 13, no. 4, pp. 600–612, 2004.