BOSTON UNIVERSITY

FACULTY OF COMPUTING AND DATA SCIENCES

Thesis

ADVANCING HUMAN-AI COLLABORATION: MULTIMODAL SYSTEMS FOR ACCESSIBILITY, RESEARCH, AND LEARNING

by

FARID KARIMLI

B.A., Boston University, 2023

Submitted in partial fulfillment of the

requirements for the degree of

Master of Science

2025

Approved by

First Reader

Dr. Thomas Gardos
Associate Professor of the Practice of Computing & Data Sciences Director of MS in Data Science Program

Second Reader

Dr. Margrit Betke
Professor and Director of Masters Admissions and Director of Masters in AI

Third Reader

Dr. Jeffrey Considine
Associate Professor of the Practice of Computing & Data Sciences and Computer Science

Nil sine labore.

Acknowledgments

I would like to express my deepest gratitude to Professor Margrit Betke and Professor Thomas Gardos for their support and guidance over the past year. Their mentorship has been instrumental in shaping my academic growth and helping me navigate the challenges of this work.

I also would like to thank Alex Burnham and the residents of The Boston Home in Dorchester, MA for their support in testing CameraMouseAI and KeyGlide. I also thank the college students who tested CameraMouseAI, and the DS701 students who participated in the pilot deployment of the Tools4DS tutor.

I would also like to thank my family and friends for their support and encouragement throughout this process.

ADVANCING HUMAN-AI COLLABORATION: MULTIMODAL SYSTEMS FOR ACCESSIBILITY, RESEARCH, AND LEARNING

FARID KARIMLI

ABSTRACT

This thesis explores domain-specific AI applications in assistive technology, botany and education. First, CameraMouse, an AI-powered system, enables computer use for individuals with severe motor impairments by converting head movements into cursor control. User studies highlight its effectiveness in enhancing accessibility. Next, the thesis addresses species recognition and analysis in herbarium research, using fine-grained image classification, zero-shot learning, and conversational models. Early experiments show promising results in identifying subtle species differences. Lastly, a suite of tools and modules to enable AI-powered academic assistants was developed. These contributions demonstrate AI's impact on accessibility, botany, and education.

Contents

List of Tables

List of Figures

List of Abbreviations

FGVC	.....	Fine-grained Visual Classification
LLM	.....	Large Language Model
VQA	.....	Visual Question Answering

Chapter 1

Introduction

The intersection of artificial intelligence (AI) and human-computer interaction (HCI) has transformed the way people interact with technology, enhancing personal workflows, interaction with devices and productivity. AI-driven systems have played a pivotal role in improving accessibility through tools like voice-controlled assistants and speech-to-text software, enhancing education with intelligent tutoring systems, and advancing scientific research through automated image analysis in fields such as medical diagnostics and environmental monitoring. By leveraging the synergy between multimodal AI and user-centric design, these systems address critical societal needs, paving the way for inclusive and efficient workflows.

This thesis presents contributions in three interrelated areas: assistive technology for accessible device control, academic engagement through AI tutors, and large-scale image classification for herbarium image analysis. Each project integrates state-of-the-art AI models to solve unique challenges while contributing to the broader goal of designing intelligent systems that enhance access, productivity, and collaboration.

1.1 Empowering Accessible Personal Device Control

Assistive technologies are critical for individuals with severe motor impairments, who often face significant barriers when interacting with traditional input devices. To address this need, this thesis introduces two complementary systems: CameraMouseAI and KeyGlide.

1. CameraMouse^{AI} is a head-controlled mouse replacement system for individuals with severe motor impairments that enhances traditional dwell-time mechanisms by incorporating customizable facial gestures. Its customization allows for easy adaptation to individual user needs, ensuring flexibility across diverse motor impairments. By leveraging AI-based facial feature tracking, the system empowers users to interact with personal devices independently and intuitively. This work was published in ASSETS '24 (Karimli et al., 2024).

2. KeyGlide is an on-screen text input interface that is purely motion-based, meaning it enables users to select keys and words by mouse pointer movement only. Users do not need precise control of the mouse pointer location since no pointing and clicking activities are required for the selection operation. Users can select groups of keys and keys within these groups by simply moving the mouse pointer sideways when the desired group or key becomes highlighted by an automated gliding-through-choices process. Word completion, prediction and spell-check are integrated into the interface to support the user in fast and accurate text input.

Together, these systems aim to provide greater autonomy to users, advancing accessibility in personal computing and communication. This thesis describes extensive user experiments with CameraMouseAI and KeyGlide, involving users with severe motion impairments.

1.2 Driving Student Engagement using Academic AI Assistants

The rapid growth of online education, student enrollment and complex course content necessitates intelligent tools that support both students and educators. This

thesis introduces Edubotics.ai, a platform designed to enhance academic engagement through AI-powered conversational assistants.

1. The platform includes intelligent data extraction pipelines capable of processing diverse content formats such as PDFs, Markdown files, Jupyter notebooks, and GitHub repositories. These pipelines not only capture complex visual elements but also extract semantic metadata to ensure contextually accurate responses.

2. A robust retrieval system, designed to handle diverse course content, enables dynamic adaptation to different learning environments.

The work described in this thesis is a step toward the goals of the platform's developers, to seamlessly adapt to new courses and diverse content formats, helping instructors easily deploy assistants that provide personalized support for students, reduce staff workload and provide insight into students' learning progress.

1.3 Towards a Foundation Model for Herbarium Analysis

Herbarium specimens play a pivotal role in understanding plant morphology, taxonomy, and ecological trends. However, their manual curation remains labor-intensive and prone to human error. This thesis contributes to the development of a multimodal foundational model for herbarium specimen analysis, emphasizing the integration of computer vision and natural language processing for zero-shot classification tasks.

1. Leveraging the SWIN Transformer architecture (Liu et al., 2021a), this work seeks to replicate and improve upon previous research that demonstrated strong performance in large-scale herbarium species recognition, where a team achieved 87% accuracy on NAFlora-1M, a dataset of over 15,000 herbarium species (Park et al., 2024).

2. This work seeks to scale the classification by combining SWIN with CLIP (Radford et al., 2021) - a model capable of recognizing categories it was not explicity trained on by understanding the semantics of text (species names) and images.

These are steps towards developing a collection of multimodal foundation models tailored for large-scale herbarium analysis. The ultimate goal is to streamline botanical research by providing scalable and accurate tools that reduce the burden of manual specimen annotation and can understand and inform visual differences between species images.

This thesis underscores the transformative potential of AI when applied to diverse domains such as accessibility, education, and scientific research. By building on established methodologies and exploring novel solutions, the work presented here provides a foundation for future advancements in assistive technologies, academic tools, and fine-grained classification models. These projects are unified by a common goal: towards AI-powered intelligent systems that enhance productivity, learning, and access.

Chapter 2

Empowering Accessible Personal Device Control using Facial Feature Tracking and Gesture Recognition

2.1 Introduction

Recent advancements in assistive technologies for individuals with severe motor disabilities such as Amyotrophic Lateral Sclerosis (ALS) and Multiple Sclerosis (MS) have led to the development of several mouse-replacement solutions. Earlier solutions used classic computer vision techniques such as template matching for facial feature tracking and detection, translating their location to mouse pointer coordinates (Betke et al., 2002; Su et al., 2005). Other solutions rely on deep learning techniques such as object detection to track facial features (Kalabarige et al., 2023), or record head (Fu and Huang, 2007; Kurauchi et al., 2015; Waber et al., 2005) and eye gaze direction (Alagarsamy et al., 2022; Feng et al., 2021; Vazquez-Li et al., 2016; Kurauchi et al., 2015), and use input from external sensors and devices to move the mouse cursor (Huang et al., 2006).

Existing technologies face significant challenges, including limited interface control, high costs, reliance on external devices, and inadequate customization options. These limitations underscore the need for innovative solutions that offer greater precision, ease of use, and cater to the diverse needs and conditions of users with movement disabilities (Feng et al., 2018). Current solutions also tend to be self-contained and

do not easily allow for updates or integration with new components. The lack of modularity prevents researchers and developers from upgrading certain aspects of available systems, such as implementing better head movement trackers or selection mechanisms, without having to replace the entire system.

To address these challenges, an AI-based mouse-replacement interface, called CameraMouse^{AI}, is proposed. This interface enables users to customize features according to their preferences and needs. Every aspect of its operation is customizable: mouse pointer movement sensitivity, gestures for clicking and their sensitivities, and screen exclusion. These adaptive capabilities, paired with state-of-the-art tracking technologies and a novel method to map a user's nose tip to mouse pointer coordinates, significantly reduce the cognitive effort required for interaction (Magee et al., 2011). Using our interface, a computer user can navigate their device screen with head movements and facial gestures, and can thus browse the web. To enable textual input, a no-click, movement-based keyboard interface was developed and can be used with CameraMouse^{AI}.

We designed CameraMouse^{AI} with a modular architecture that allows for seamless replacement or augmentation of AI models that interpret the user's intent, thereby extending the useful life of the technology without requiring users and researchers to invest in entirely new systems. For example, researchers can update face or facial landmark detection models, or add new selection mechanisms based on users' requirements. The modularity of CameraMouse^{AI} thus supports the design of tailored research tools that can adapt to diverse and evolving patient needs. The main contributions of this work are to provide

- a mouse-control interface for people with severe motion disabilities that is based on real-time artificial intelligence and has extensive customization options, thereby offering a more personal device interaction experience on case by

case basis,

- a modular architecture and open-source code that enable researchers and developers to experiment with and enhance the system,
- a pilot user study that involved individuals with and without disabilities.

The code for CameraMouse^{AI} can be downloaded at
https://github.com/hicsail/cameramousejs

In the following sections, I describe our design, discussing our user-centric approach, the integration of AI technologies, and the significance of the modular design in the broader context of assistive technologies. We also describe a small user study. My specific contributions are integrating facial gestures for clicking, adding customization options such as facial gesture thresholds, general code infrastructure, and the KeyGlide text input interface.

2.2 CameraMouse^{AI} Interface

2.2.1 Interface

The Interface component is responsible for managing user interactions and system configurations. It features a minimalist graphical user interface (GUI) with two primary tabs: "Home" and "Settings." The Home tab, shown in Fig. 2·1, provides the user with real-time feedback on the location of their tracked facial feature, the nose, within an operative window of possible movements. The Home tab also enables users to initiate or stop tracking via keyboard input. The Settings tab, shown in Fig. 2·2, offers customization options such as click mechanisms (dwell time, mouth opening, eyebrow raising), sensitivity adjustments, and screen exclusion settings. Users can fine-tune parameters like gesture thresholds and mouse pointer sensitivity to suit their individual needs, enhancing usability and reducing accidental inputs.

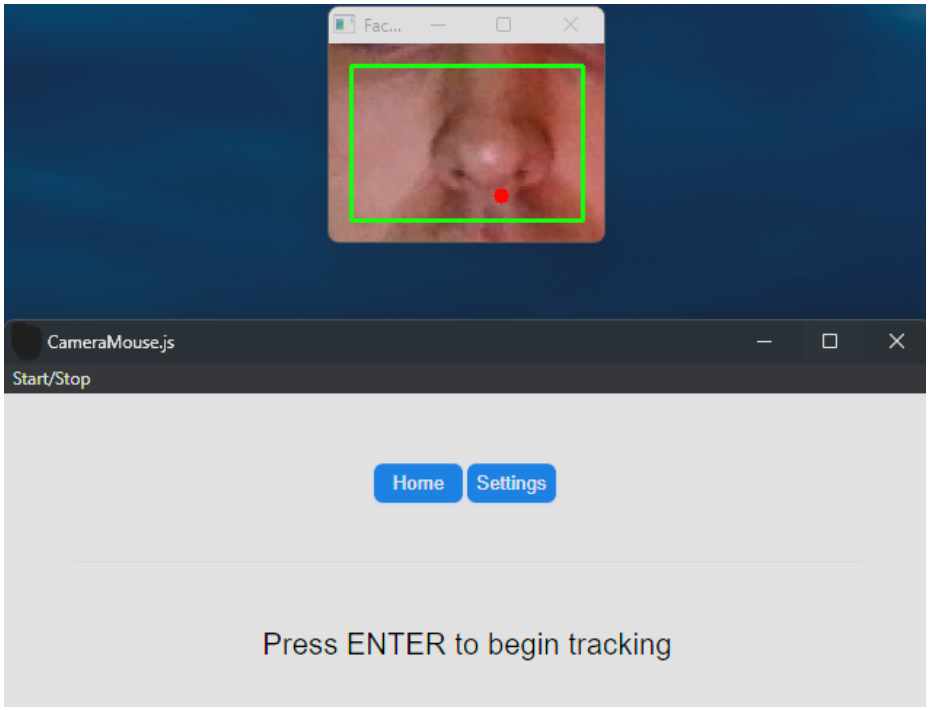

Figure 2·1: Graphical User Interface of the CameraMouse^{AI}: Home tab and webcam image showing the operative window (green box) and the tracked facial feature (red dot).

2.2.2 Architecture and Modular Design

An overview of the architecture of CameraMouseAI is provided in Fig. 2·3. The application consists of two com- ponents: the Interface (green box) and the Video Processor (purple box). The interface handles the interaction of the user with the system, enabling the user to start or stop the tracking, switch between clicking methods, and customize settings. The video processor performs face and facial landmark detection, followed by tracking and command interpretation. The two components communicate in real time to ensure uninterrupted, smooth operation. The interface receives screen coordinates and mouse commands from the video professor. User config- uration parameters, e.g., pointer sensitivity or gesture thresholds, are sent to the video processor. The design of the video processor enables researchers and developers to replace face and facial landmark recognition models, change which facial

feature to track, or add gestures to enhance interaction functionality. Components shown in yellow in Fig.2·3 can be easily exchanged, components in orange provide the backbone of the system. In the following subsections, we discuss these components in detail.

Enabling Modularity

CameraMouse^{AI} employs a modular architecture that supports flexible integration of facial landmark detection and tracking models. This design facilitates easy replacement or enhancement of components to accommodate advancements in computer vision technologies. The system's robustness is underscored by its ability to adapt to different facial feature tracking methods and detection models, ensuring continuous innovation and improvement.

We here explain the developmental framework that enables this modularity. To use a different facial landmark model, developers only need to modify a few lines of code, as the mapping of facial landmarks is decoupled from the model output. This flexibility extends to selecting which facial features to track by specifying the appropriate index in the model's output.

The video processor maintains a configuration object, *trackerState*, which stores all user-defined settings and communicates them to the interface via HTTP. On the interface side, visual components (Fig. 2·1 and 2·2) are implemented as modular objects using ElectronJS, a desktop framework leveraging HTML, CSS, and JavaScript. This design simplifies customization and benefits from strong community support.

Adding new features requires minimal effort:

- In the interface, developers add a list item for the new clicking mechanism (e.g., Fig. 2·2) and replicate an existing component, such as a slider for thresholds.

- Add a corresponding field is added to the *appConfig* object, ensuring the up-

dated settings are sent to the video processor.

- The video processor then handles the new mechanism with either a function to detect clicks or a boolean flag for model-based triggers.

This modular design positions CameraMouseAI as a flexible, open-source platform that encourages collaboration and continuous innovation in assistive technologies. Researchers and developers can seamlessly integrate new AI models, enhancing system capabilities to meet diverse user needs.

2.2.3 Mapping Visually Tracked Feature to Mouse Pointer Coordinate

A critical aspect of CameraMouse^{AI} is its mapping mechanism, which translates the coordinates of visually tracked facial features into precise mouse pointer movements on the computer screen. Users can adjust the size and position of the "operative window," a focused area within the camera view that dictates mouse pointer movement, thereby optimizing control and responsiveness. This means that the movement of the mouse pointer is not absolute, but relative to the operative window. The edges of the window correspond to the edges of the screen, meaning moving the mouse pointer to an edge of the window will move the pointer to that edge of the screen.

Figure 2·4 provides a visual illustration of our approach. To map the tracked facial feature coordinates (c_x, c_y) from the camera view to mouse pointer coordinates (s_x, s_y) on the computer screen, we address two key challenges: 1. The *camera view* and *screen dimensions* $(c_{\mathrm{dim}_x}, c_{\mathrm{dim}_y})$ and $(s_{\mathrm{dim}_x}, s_{\mathrm{dim}_y})$ differ in resolution and aspect ratio. 2. Users cannot move facial features across the entire camera view naturally; instead, movement is concentrated within a central *operative window* (Fig. **??**(a)).

The *operative window*, adjustable via the user interface ("sensitivity" in Fig. 2·2(b)), defines a smaller subregion $(o_{\mathrm{dim}_x}, o_{\mathrm{dim}_y})$ centered in the camera view. Reducing its dimensions proportionally decreases physical movement required to control the mouse

across the screen.

The mapping process (Fig. 2·4) involves three steps:

1. **Normalize Camera Coordinates**: Convert the pixel coordinate (c_x, c_y) into a ratio within the camera view:

$$(c_{r_x}, c_{r_y}) = \left(\frac{c_x}{c_{\dim_x}}, \frac{c_y}{c_{\dim_y}} \right), \quad c_{r_x}, c_{r_y} \in [0, 1]. \tag{2.1}$$

2. **Scale to Operative Window**: Map the normalized coordinates to screen ratios (s_{r_x}, s_{r_y}) using a scaling function:

$$\text{Scale}(c_{r_x}) = \begin{cases} 0 & c_{r_x} \leq 0.5 - \frac{o_{\dim_x}}{2c_{\dim_x}} \\ 1 & c_{r_x} \geq 0.5 + \frac{o_{\dim_x}}{2c_{\dim_x}} \\ 0.5 - \frac{0.5 - c_{r_x}}{\frac{o_{\dim_x}}{c_{\dim_x}}} & \text{otherwise.} \end{cases} \tag{2.2}$$

   Points outside the operative window are mapped to the screen boundaries, while points inside are scaled proportionally.

3. **Convert to Screen Coordinates**: Finally, compute the mouse pointer coordinates (s_x, s_y):

$$(s_x, s_y) = (s_{r_x} \cdot s_{\dim_x}, s_{r_y} \cdot s_{\dim_y}). \tag{2.3}$$

This approach ensures smooth and precise mapping of facial movements to screen interactions while minimizing unnecessary physical effort for users.

2.2.4 Video Processor

The Video Processor component integrates real-time face detection and facial landmark identification using a MobileNetV2-inspired convolutional neural network provided by Mediapipe (Lugaresi et al., 2019), specifically modified for real-time usage.

Given a video frame, the model predicts 478 facial features, ranging from the eyebrows to the chin. CameraMouse^{AI}'s Video Processor specifically tracks features around the nostrils to determine mouse pointer coordinates accurately. Unlike traditional methods that directly translate facial landmark locations into mouse movements, CameraMouse^{AI} combines feature detection with template-based tracking for smoother and more efficient pointer control. Automatic re-initialization mechanisms ensure tracking continuity even during fast movements or occlusions. The model also outputs predictions on facial expressions, called "blendshapes." These are extensive, ranging from mouth opening to winking. CameraMouse^{AI} uses these "blendshapes" from the model along with additional facial landmark processing to determine whether a gesture has been made.

Tracking of Facial Feature

When the facial landmarks are detected, a square is drawn around the location of the midpoint of the nostrils. A sub-image within this square is cropped and used as a "template" for tracking the position of the nostrils in the subsequent frame. The area is chosen because the nostrils provide distinctive features which present notable changes in intensity, leading to improved performance in template-based tracking. Additionally, the shape of the nostrils remains relatively stable compared to other facial features, such as the mouth, which can vary significantly especially during speech or expressions, ensuring consistency of the template. We selected nostrils as the feature to be tracked also because we considered the perspective of a user interfacing with the computer – nostrils offer the user an intuitive and natural point of reference, given the human tendency to utilize the nose for pointing and indicating direction.

Our method searches for the nostril sub-image in the current video frame within a "search window" centered at the position of the sub-image in the previous frame. The

template is successively shifted through this search window and correlated with the underlying sub-images. The normalized correlation coefficient between the template T and the sub-image S in the search window is computed as

$$R(T,S) = \frac{\sum_{x,y}(T(x,y) - \mu_T) \cdot (S(x,y) - \mu_S)}{n\sigma_T\sigma_S}, \tag{2.4}$$

where μ_T and σ_T are the mean and standard deviation of all pixels in the template, and μ_S and σ_S are the mean and standard deviation of all pixels in the sub-image. n is the number of pixels in the template and the sub-image. The sub-image with the highest correlation coefficient is determined as the tracked feature in the current frame. The location of this sub-image is mapped to the actual mouse location on the screen. In addition, we update the template using the sub-image for tracking in the next frame.

2.2.5 Clicking Mechanisms: Dwell Time and Facial Gestures

There are three ways to click using CameraMouse^{AI}: dwell time, opening mouth and raising eyebrows. Dwell time is a classic interaction technique where a click is administered when the mouse pointer stays in the same approximate location for a set amount of time (Feng et al., 2021). That time can be customized (Figure 2·2). The nature of dwell time requires rest areas on the screen, otherwise everywhere the user looks can be clicked (Jacob, 1990). Gestures do not have this problem, and thus offer more control. We have implemented two gestures for issuing mouse commands, mouth-opening and eyebrow-raising, and provide a choice for the user in the CameraMouse^{AI} GUI (Fig. **??**(b)). Since everybody's mouth and eyebrows sizes, and distance to the screen could be different, the application allows the user to change the threshold for gesture detection. So, if the user believes that they are making enough effort in making the gesture but the application is not issuing clicks,

they can lower the threshold. Any of the three can set to left, right and double clicks.

CameraMouse^{AI} recognizes a user's facial gestures based on MediaPipe's real-time facial gesture detector (Lugaresi et al., 2019). The detector identifies the occurrence and intensity of 52 facial gestures based on a 3-dimensional face mesh model.

2.3 KeyGlide: No-Click, Low-Cognitive-Load Text Input

During our experiments and user studies, we found it challenging to use CameraMouse^{AI} with on-screen keyboards, as it required high precision even for users without disabilities. Therefore, we developed KeyGlide, a no-click text input interface that enables typing with minimal accuracy. The user selects the letter group first, then the letter, by moving the mouse pointer into an area at the right time as the system cycles through the letters and groups. Typing is further boosted by word prediction, completion and spell check. This low-cognitive-load design ensures users can input text accurately with limited precision in movements.

The layout consists of a central letter grid, a rest area, a word prediction section, and control buttons. The letter grid organizes the alphabet into groups, such as "A-D" or "E-H," allowing users to quickly navigate to specific clusters. Adjacent to the grid, the rest area serves as a neutral space for pausing input, helping users avoid accidental selections. Above the grid, the word prediction section dynamically suggests words based on typed letters, enabling faster input. Control buttons at the top of the interface, including options like Save, Delete, Copy, and Clear All, provide essential text management functionality.

To type, users navigate their mouse pointer or motion controller to the into the 'key area' (left side) of the interface, at the right time. Once inside the group, letters are highlighted sequentially, allowing the user to select the intended letter by moving back into the same area. So, to pick the letter 'F', the user would wait till the orange

band reaches the second row in the key area, move into that area, move back into the rest area. Then, the user would wait till the letter 'F' is highlighted, and move back into the key area to pick the letter. The prediction feature further accelerates typing by allowing users to select suggested words instead of typing each letter. This intuitive design minimizes effort and cognitive load, making KeyGlide effective for users with limited motor function.

Note that the two applications (CameraMouse^{AI} and KeyGlide) do not have to be used together. Users can opt to use KeyGlide with any other mouse-replacement interface, or a physical device like a roller mouse.

2.4 User Studies

Two studies were conducted: one evaluating the CameraMouse^{AI} interface, and one evaluating the KeyGlide text input interface. In the CameraMouse^{AI} study, all users achieved near-100% accuracy in a target selection task and success in navigating the web in a browser.

2.4.1 CameraMouse^{AI} Study

Twelve college students without disabilities and four individuals with advanced multiple sclerosis tested CameraMouse^{AI}, performing a multi-directional target selection and web browsing tasks.

Target Selection

The target selection task involved selecting 10 rectangular targets arranged in a circular manner (see Figure 2·6), deliberately numbered to make the user move across the circle center from target to target. Targets disappeared once clicked. Each user without a disability completed 5 blocks for each of the three clicking mechanism, resulting in 15 blocks per user. This task measured how many targets were clicked, the

time to move to a target (ballistic time), the time to select the target (select time), and the overall time to complete the task. Ballistic time refers to the time taken for the initial movement towards a target, and selection time is the time required to finalize the selection of a target once inside the target area.

Every user without a disability was able to successfully complete five blocks of the target selection task for each clicking mechanism, except two that were not able to use the eyebrow raise gesture due to low contrast around their eyebrows. Users demonstrated 100% accuracy in selecting targets in the specified order. The total, ballistic, and selection times are shown in Figure 2·7. Each clicking mechanism exhibited comparable completion times, showing that the facial gestures were at least as effective as the traditional dwell time technique. Analysis of the ballistic and selection times showed higher average ballistic times and a larger range of selection times for the dwell time mechanism. This was due to accidental clicks, which were predictably frequent with the dwell time mechanism, but not with the facial gestures. Facial gestures not only reduced the total completion time but also exhibited a smaller range in ballistic and selection times, indicating greater consistency across users. Notably when it comes to selection time, performed the best, achieving a median selection time of approximately 0.8 seconds with minimal spread, suggesting its efficacy as a reliable alternative for target selection. This highlights the potential of facial gestures to enhance accessibility tools for individuals with motor impairments.

Overall, the eyebrow raise gesture proved to be the most stable across all metrics as evidenced by the low means and small range of time metrics.

While our experiments with individuals with severe motor impairments were less structured than traditional user studies, this was an intentional decision. Working with participants who face significant motor challenges requires flexibility to accommodate their needs and ensure their comfort during the testing process. Instead of

Table 2.1: Summary of target selection task with users with motor impairments. "Block" refers to a successfully clicking on all targets; *7/10 targets on one block; **8/10 targets.

	Dwell Time	Open Mouth	Raise Eyebrows
User 1	3 blocks	2 blocks	2 blocks
User 2	3 blocks*	3 blocks	0 blocks
User 3	2 blocks	1 block**	2 blocks
User 4	2 blocks	5 blocks	2 blocks

adhering strictly to a predefined protocol, we focused on observing how participants interacted with CameraMouseAI in real-world scenarios. The numbers of experimental blocks users with advanced multiple sclerosis were able to perform with the three selection mechanisms are reported in Table 2.1. In all cases but 2, the users were able to click on all 10 targets, achieving a near 100% success rate.

The target selection results for users with motor impairments showed promising performance, with participants achieving largely consistent performance across blocks, with notable improvement in certain cases (Fig. 2·8). Dwell-time again emerged as the most successful clicking method across participants, with faster selection and ballistic times compared to the open-mouth and eyebrow gestures. Performance variability across participants highlighted individual differences in motor control and learning rates. For instance, User 1 demonstrated some improvement in selection time, while User 2 showed fluctuations likely due to fatigue or adjustment to the task. These results underscore the importance of customizable sensitivity and gesture thresholds to accommodate diverse user abilities.

One user was not able to use the eyebrow raise gesture due to their motor impairment and facial attributes (low contrast between eyebrows and the rest of the face).

Table 2.2: Summary of target selection task with randomized arrangement with users with disabilities. "Block" refers to a successfully clicking on all targets.

	Dwell Time	Open Mouth	Raise Eyebrows
User 4	2 blocks	5 blocks	2 blocks
User 2	3 blocks	2 blocks	2 blocks
User 3	6 blocks	2 blocks	0 blocks

Adaptation to Randomized Tasks

To test adaptability, we introduced a randomized rotation of target arrangements for users with motor impairments, randomly rotating the target arrangement after every block while keeping the relative positioning the same. We show how many blocks each of the three participants successfully completed in this study in Table 2.2 and the (normalized) progression of ballistic and selection time at every block in Fig. 2·9. Despite the change in target positioning, participants showed comparable performance to the fixed-arrangement task (Fig. 2·8). The average time to complete a block remained largely stable across gestures and successive blocks. Notably, User 4 showed remarkable improvement in times while using dwell time, and User 2 showed consistency in successive blocks while using the open mouth gesture.

These results suggests that users can successfully generalize their learned interaction patterns to new spatial arrangements, highlighting the robustness of the CameraMouseAI interface.

Movement patterns: Overshooting

A recurring trend of users with motion impairments was overshooting targets upon the initial approach and then readjusting to move back into the target to click. This overshooting pattern is typical among people with multiple sclerosis (Nij Bijvank et al., 2022). In our experiments, it happened with targets at the vertical extremities of the screen (and not for the other targets).

For targets at the top of the screen, instead of entering the target from the bottom (as is the natural approach), the mouse pointer would enter from the top. Vice versa, for the targets at the bottom of the screen, the user's mouse pointer would enter the target from the bottom instead of the top.

We illustrate examples of this phenomenon for one of the users with motor impairments in Figure 2·10.

Browser Navigation Task

The browser task involved navigating through seven pages on Wikipedia, starting from the computer mouse page

`https://en.wikipedia.org/wiki/Computer_mouse`. The users were asked to click on hyperlinks, buttons, move from one side of the browser window to the other, and scroll. To scroll, users clicked on the scroll bar. Four of the seven browser navigation tasks are shown in Fig. 2·11. Since some of the users with multiple sclerosis also had visual impairments, we provided visual aids. In particular, we increased the resolution of the web page to 200% and installed the Custom Scroll Bar extension of Google Chrome (Branton, 2024), which allowed us to increase the contrast around the scroll bar.

The timing results for the browser tasks with users without disabilities for each clicking mechanism are summarized in Figure 2·12. The mean time to complete the browser task, which involved navigating through seven web pages, had a range from 40 s to under 60 s, with dwell time clicking being the fastest. Three of the four users with MS were able to complete the browser task successfully at least twice for each gesture.

Phrase to be Written	User	SD	Time (sec)	Predictions
HAPPY TO SEE YOU	User 1	4	177	2
GLAD YOU COULD VISIT	User 1	3	269	3
I APPRECIATE YOUR TIME	User 2	6	608	2
ITS A GOOD DAY	User 2	0	174	3
JOIN ME FOR LUNCH	User 2	3	502	2
KILL BILL	User 3	1	172	0
CALL ME WHEN YOU CAN	User 3	1	323	3
I LOVE TALKING WITH YOU	User 3	0	297	2
GLAD YOU COULD VISIT	User 3	0	146	4

Table 2.3: Performance metrics for KeyGlide user study. Users were tasked to input different phrases of 3-5 words, and metrics such as difference from the correct string (String Distance), deletions, and predictions used are shown.

2.4.2 KeyGlide Study

The KeyGlide user study involved three participants with severe motor impairments, testing the system across a series of phrases designed to simulate real-world text input scenarios. Two of the users used CameraMouse^{AI} and one used a roller mouse. Part of the study was to show that KeyGlide does not need to be used with CameraMouse^{AI}, but can be used with any other mouse-replacement interface, or even a physical device like a roller mouse.

They were tasked to input different phrases of 3-5 words, which included phrases like "HAPPY TO SEE YOU" and "GLAD YOU COULD VISIT". These phrases are phrases that people are likely to use in daily communication with friends and family. The results highlight the potential of KeyGlide to facilitate accesible text input. Results are summarized in 2.3.

Performance varied between users based on their motor abilities and prior experience with assistive devices. User 1, a CameraMouse user, demonstrated steady performance with an average WPM (words per minute) of 4.6. Prediction features contributed significantly to this user's efficiency, with examples such as completing the word "GLAD" in 43 seconds—a substantial improvement over non-predicted in-

put times. MSD values for User 1 ranged between 3 and 4, reflecting moderate errors that were easily corrected using prediction. In contrast, User 2, who faced more severe impairments, achieved the lowest typing speeds, with an average WPM of 2.6. While prediction improved their WPM to 4.2, detection inconsistencies and slower reaction times limited its full effectiveness. User 3, a roller mouse user, achieved the highest performance, with an average WPM of 5.33 and nearly perfect accuracy (MSD values of 0-1) across most phrases. They completed the phrases efficiently, like phrase "GLAD YOU COULD VISIT" in 146 seconds without any mistakes.

The impact of prediction was most pronounced in terms of efficiency. Prediction also facilitated more consistent typing speeds across phrases, reducing user effort and increasing task completion rates. Using prediction, typing speed improved by an average of 103% letters per second (LPS), effectively doubling users' typing efficiency. While this is promising, the small sample size of three users limits the generalizability of these findings. A larger, more diverse sample is needed to determine the robustness of this effect across different users and impairments. Prediction's effectiveness varied across participants. While Users 1 and 3 demonstrated significant benefits from prediction, User 2 struggled due to input inconsistencies caused by motor impairments. This disparity highlights the need for further development to make prediction features more adaptable and accessible to users with varying levels of mobility.

Error patterns revealed common challenges, including insertions, deletions, and substitutions. Insertions often occurred during pauses or when re-entering selection zones, as seen in outputs like "GGLAD" instead of "GLAD." Participants also spent a lot of time simply waiting for the system to get to the group/letter they wanted to select. Often, they thought they had to move towards the letter they wanted to select, when in fact they simply had to move into an area.

While these results are promising, the challenges faced by User 2 highlight op-

portunities for improvement in system responsiveness and adaptability. With enhancements in detection and error correction mechanisms, KeyGlide could become a transformative solution for accessible text input, bridging critical gaps in assistive technology for individuals with diverse motor abilities.

2.5 Conclusion and Future Work

In this paper, I introduced two new assistive platforms, CameraMouseAI and KeyGlide, designed to empower individuals with severe motor impairments by enabling access to personal devices through innovative, customizable interfaces. CameraMouseAI provides a head-controlled mouse replacement interface, extending traditional mouse replacement systems by integrating deep-learned facial landmark detection with a template-based tracking mechanism. This allows users to select gestures tailored to their specific motor abilities, offering an alternative to the conventional dwell-time selection mechanism. Similarly, KeyGlide reimagines text input by introducing a purely motion-based interface that minimizes cognitive load and eliminates the need for precise movements or clicks. Through its prediction-enhanced keyboard, KeyGlide enables users to enter text efficiently, with a design focused on adaptability and accessibility.

The empirical evaluation of both systems underscores their potential to transform how users with motor impairments interact with digital content. CameraMouseAI was tested with both individuals with and without motor impairments, demonstrating its flexibility and usability across varied user profiles. Participants appreciated its gesture-based customization options and the precise control it offered over the mouse pointer. However, pilot studies revealed limitations in text entry using generic on-screen keyboards, where closely packed keys hindered dependable selection. Building on these findings, we developed KeyGlide to address this gap, introducing a motion-

based keyboard tailored to the needs of users with motor impairments. KeyGlide's prediction feature significantly improved typing efficiency, as demonstrated in our user studies, which showed marked reductions in typing effort and time, particularly for longer phrases. The results highlight KeyGlide's potential as an accessible text input solution, capable of adapting to diverse user abilities.

The broader significance of CameraMouseAI and KeyGlide lies in their shared mission to provide user-centric, modular assistive technologies that prioritize accessibility and adaptability. Both platforms lay a foundation for future advancements in assistive technology, spanning fields such as human-computer interaction, artificial intelligence, and rehabilitation sciences. Their modular designs not only ensure flexibility for users but also offer researchers a platform for further innovation. Future research could explore integrating more advanced machine learning models, such as personalized calibration mechanisms that adapt to individual user movement patterns and capabilities, to enhance usability further. Additionally, incorporating domain-specific computer vision techniques could improve the precision and responsiveness of both interfaces.

Looking ahead, we see CameraMouseAI and KeyGlide not only as practical tools for improving accessibility but also as platforms for interdisciplinary research. By making these systems open source and modular, we encourage the research community to build upon this work, exploring new algorithms and interaction techniques to address the diverse and evolving needs of users with motor impairments. Together, these systems represent a significant step forward in assistive technology, paving the way for more inclusive and empowering digital interactions.

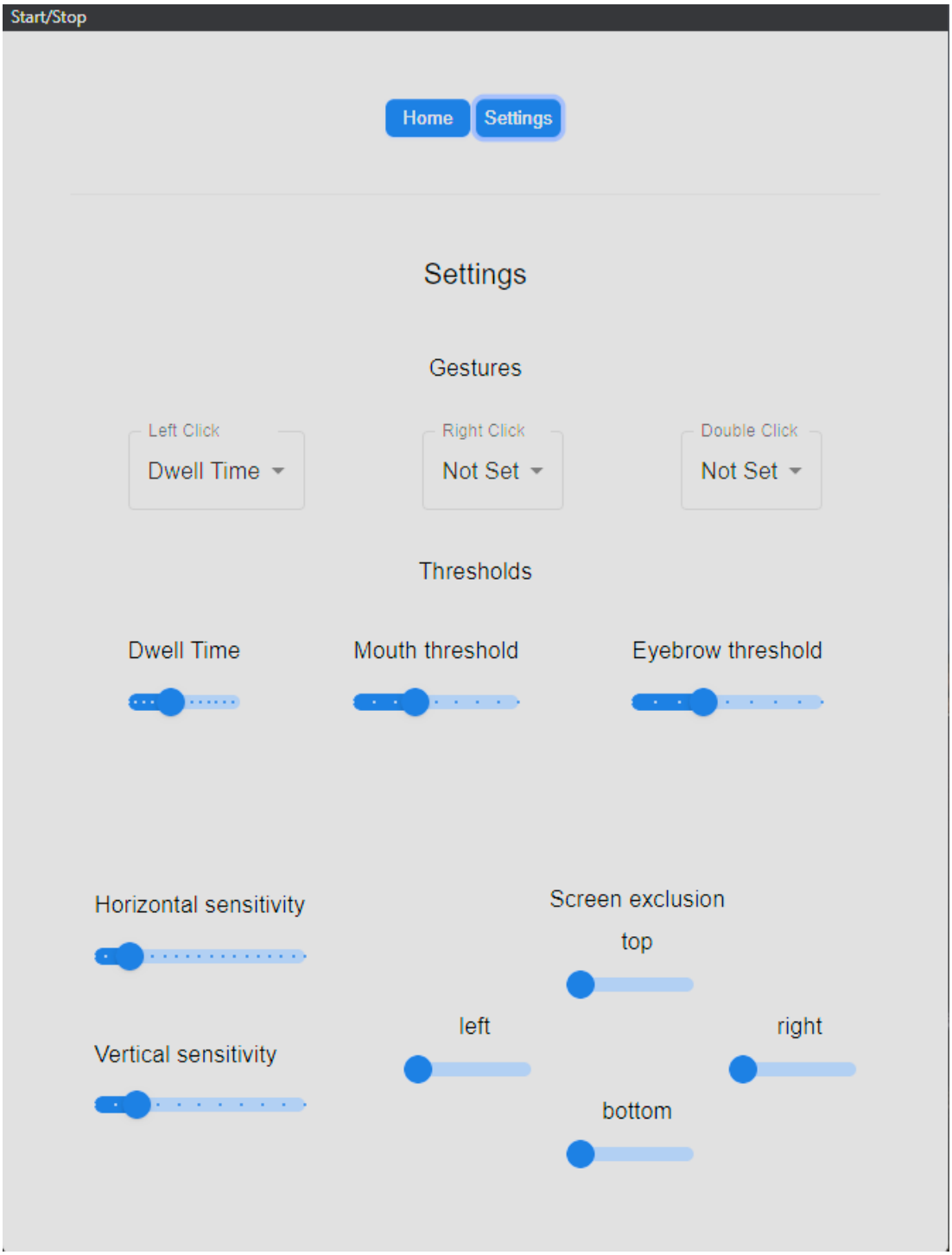

Figure 2·2: Graphical User Interface of the CameraMouse^{AI}: Settings tab where the user can customize parameters of the application.

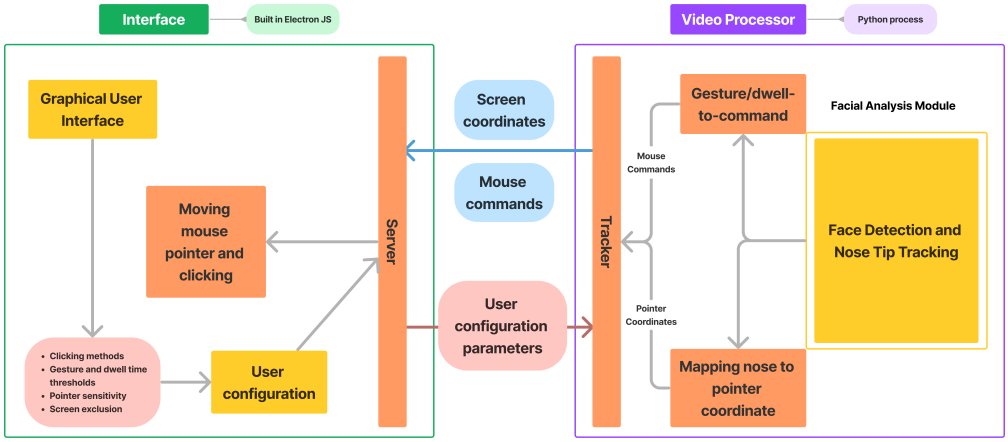

Figure 2·3: Architecture of the CameraMouse^{AI}: The interface and the video processor are separate components that can be replaced or upgraded independently.

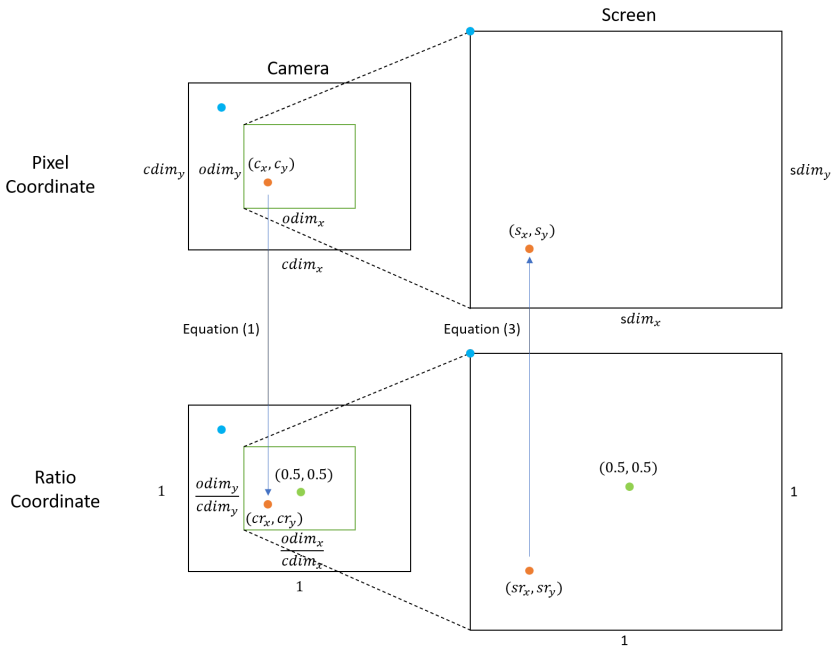

Figure 2·4: Mapping from the camera view to the computer screen.

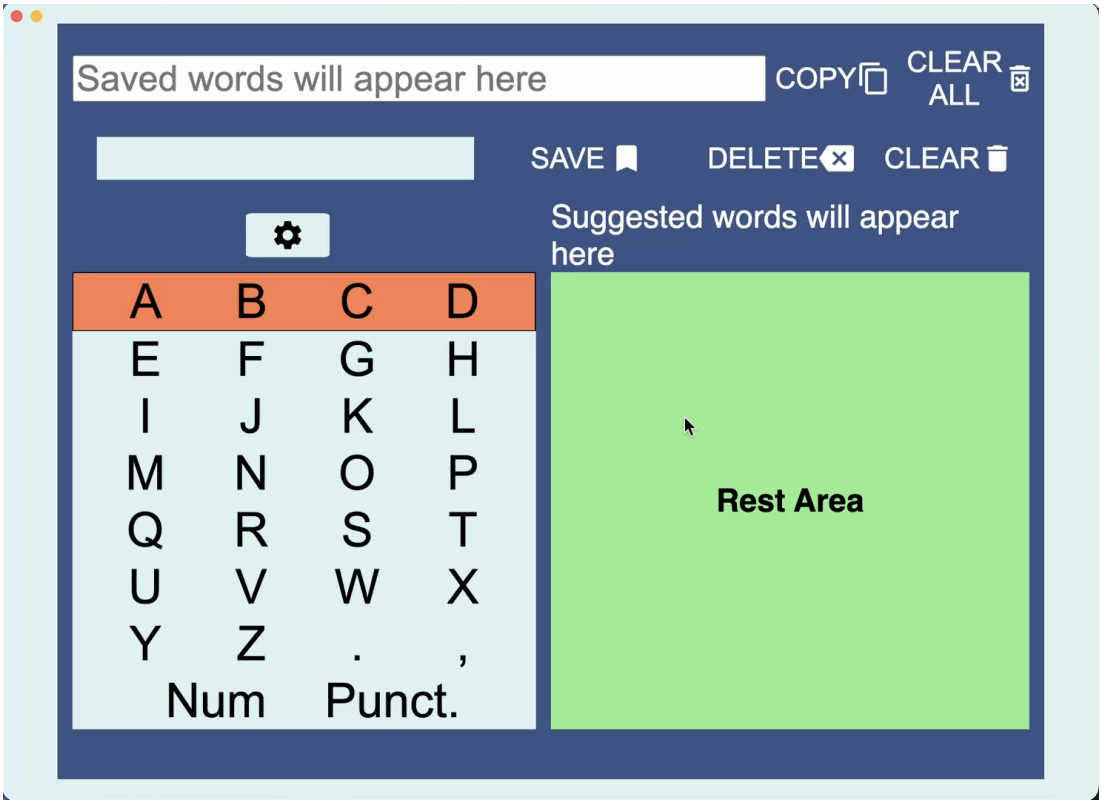

Figure 2·5: KeyGlide: No-click text input interface. User selects the letter group first, then the letter, by moving the mouse pointer into an area at the right time as the system cycles through the letters and groups.

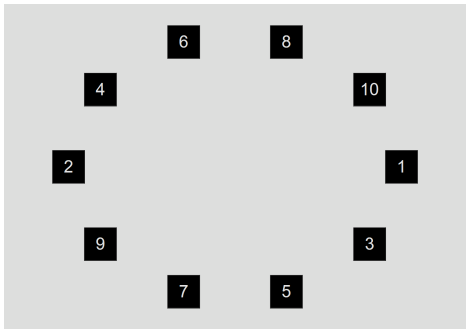

Figure 2·6: First study target arrangement of the testing interface: The user was asked to click on all targets in order. Targets disappear once clicked.

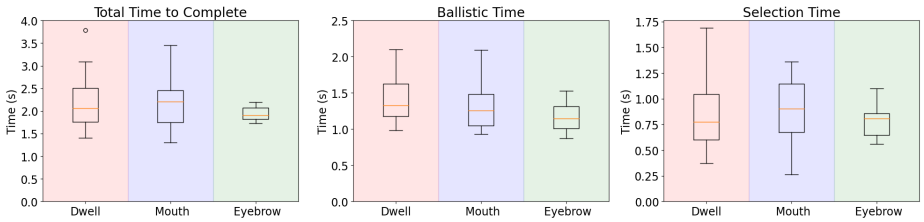

Figure 2·7: Experimental total, ballistic and selection time results with people without disabilities on the target selection task. Each box is defined by the first and third quartile of the data, shows the median time in red, and has whiskers that indicate the shortest and longest measured times.

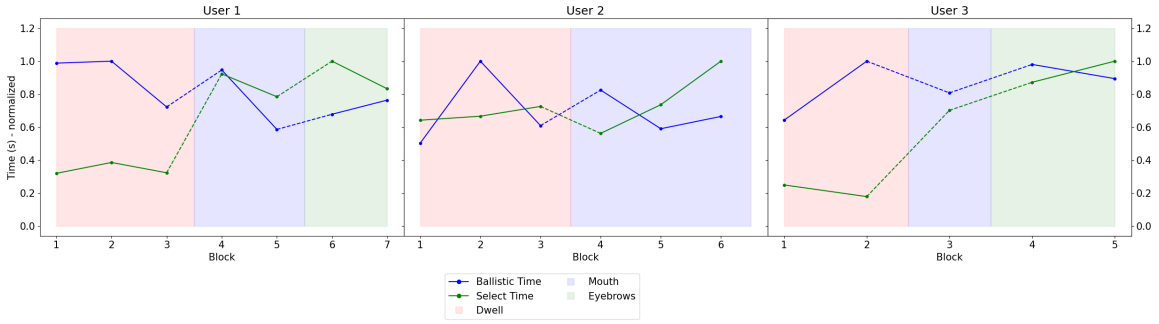

Figure 2·8: Normalized progression of ballistic time, and selection time in experiments involving with participants with motor impairments. User 2 was not able to use the "Eyebrow Raise" gesture due to physical constraints.

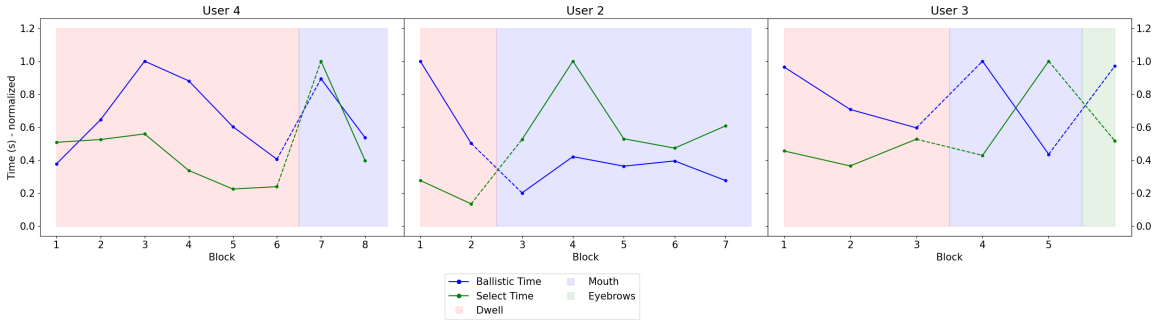

Figure 2·9: Normalized progression of ballistic time and selection time with randomized target arrangement in experiments with participants with motor impairments.

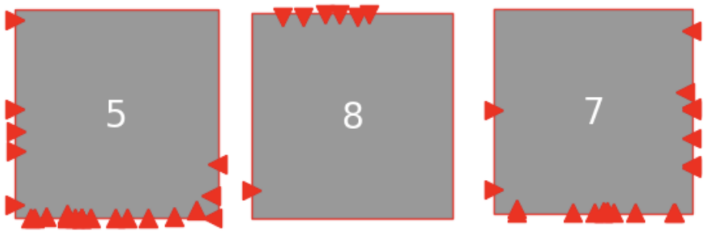

Figure 2·10: Mouse pointer entry points (red markers) into targets in extreme vertical positions for a user with motion impairments (all blocks of study 1). The user worked with the default target arrangement (Fig. 2·6). The user was asked to move the mouse pointer from target 4 at the top left of the target circle to target 5 at the bottom right of the target circle. The markers show that the user tended to enter target 5 mostly from the bottom. Similarly, target 7, which is located at the bottom left of the target circle, has entry points mostly on the bottom. Notably, for both targets 5 and 7, every side has entry points except the top, which is the natural direction of entry. Target 8, which is at the top right of the target circle, exclusively has entry points at its top, indicating that the user must have overshot target 8 during the ballistic movement from target 7 to 8.

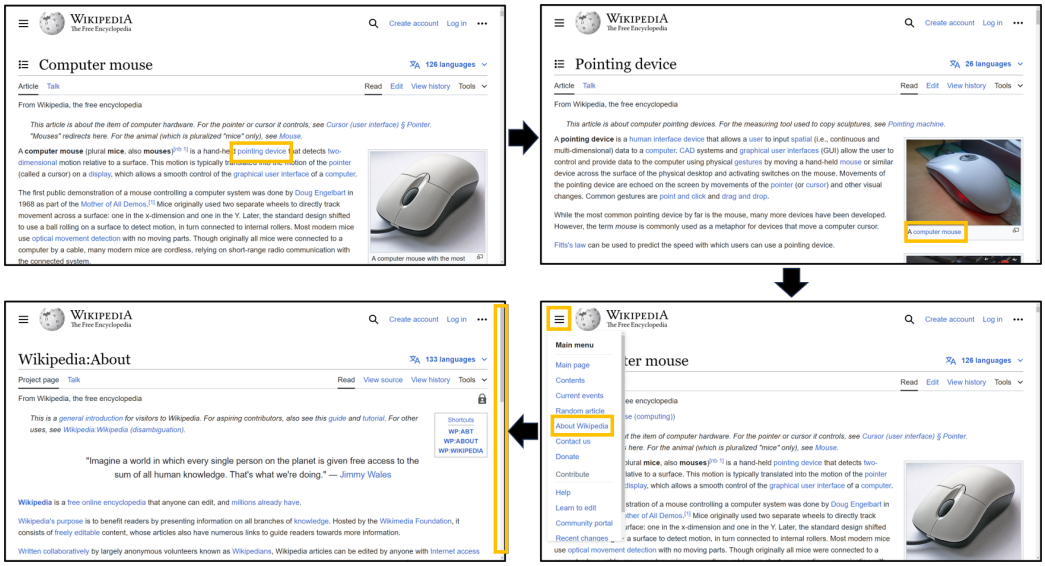

Figure 2·11: Four of the seven steps of the browser navigation task: clicking on the links "pointing device" and "A computer mouse," pulling down a menu and clicking on the link "About Wikipedia," and clicking on the scroll bar.

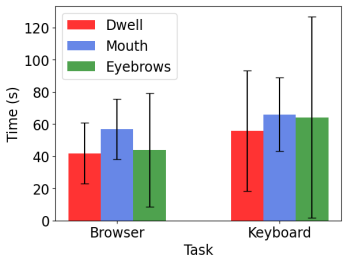

Figure 2·12: The time (average and standard deviation) that users without disabilities took to complete browser and typing tasks.

Chapter 3

Driving student engagement, learning and course development using personalized AI assistants.

3.1 Introduction

In an era where artificial intelligence is increasingly integrated into diverse sectors, education remains one of the most promising fields for conversational AI applications (Holmes and Tuomi, 2022). Higher education demands a nuanced approach due to the vast and diverse nature of its content, spanning general knowledge to highly specialized technical information. Traditional methods of student support, including office hours, forums, and tutoring, have proven valuable; however, they lack the scalability and real-time responsiveness that today's students expect. The need for advanced, personalized AI systems to bridge these gaps has never been more pressing, especially as universities worldwide continue to expand digital and hybrid learning models. An always-on, always-available assistant can provide real-time support to students, enabling them to get immediate answers to their questions and receive personalized feedback on their work. It could thus reduce the load on instructors and teaching assistants, enabling them to focus on more complex questions that could require human interaction. There have been attempts to integrate conversational assistants into learning environments (Hwang and Chang, 2021; Goel and Joyner, 2017; Hussain, 2017). With the advent of newer frontier conversational models (LLMs and

LMMs), and agentic frameworks, there is an opportunity to create more sophisticated assistants that can provide more personalized and helpful support to students.

Large language models (LLMs) and Large Multimodal Models (LMMs) are demonstrating incredible potential to function as assistants in many domains and applications. Through pre-training on massive internet-scale datasets they exhibit the capability to answer questions and solve problems. In some cases, they can approach or exceed human capability on the same task. Through supervised fine-tuning they can better align to users' expectations of helpful answers and reduce toxicity and bias. Through Retrieval Augmented Generation (RAG) techniques they base their answers and cite specific content from a knowledge base supplied by the developers. More recently, agentic architectures enable them to be used to perform more complex tasks such as multi-step planning, decision-making, and self-correction. There is a growing software ecosystem of products and services in support of these techniques. As indicated by the number of papers published on these topics, as well as the number of platforms and products being developed in industry, research in this area is evolving very rapidly.

This chapter discusses contributions to Edubotics.ai, an open-source library of tools and modules for building and deploying multi-purpose AI assistants. These assistants offer 24/7 support to students, providing them with immediate help and personalized feedback on their work. They can also be used to support instructors and teaching assistants, who can monitor the assistant's interactions with students to identify areas where additional support is needed. Edubotics.ai is designed to redefine student engagement and learning support by deploying conversational AI assistants tailored to academic contexts. In particular, three contributions are discussed: intelligent data extraction, advanced retrieval of technical course content, and seamless adaptation to different courses. Intelligent data extraction preserves the semantic

structure and integrity of source materials regardless of the format, ensuring that responses not only match the original documents in content but also reflect them accurately in context. This high-quality extraction process is fundamental to providing students with information that feels as though it's directly pulled from the course materials, enhancing the assistant's credibility and effectiveness in supporting diverse learning needs. Advanced Retrieval Augmented Generation (RAG) techniques enable assistants to flexibly engage with a broad spectrum of course material, from assignments and lectures to complex technical documentation. These two are integral to seamless adaptation to different courses by the same platform. Together, they are key to the platform's ability to create an adaptable, course-specific AI support system and to foster meaningful, personalized interactions that deepen student understanding and engagement with their coursework.

An early prototype of the Edubotics.ai platform has been piloted this Fall 2024 semester in DS701: Master's Tools for Data Science at Boston University. Future work will focus on the integration of the intelligent data extraction and retrieval techniques with more sophisticated agentic architectures to enable more fine-grained and context-aware interactions, like query rewriting and clarification questions, Socratic assistance, and multi-step planning.

3.2 Contributions to Edubotics.ai

3.2.1 Intelligent Data Extraction

Today's courses are increasingly digital and use a variety of formats for course materials (Haleem et al., 2022). These include PDFs, Word documents, HTML pages and video content (like lecture recordings). Each may include complex visual elements, like graphs, charts and tables. Even pure text can be formatted in a variety of ways, including Markdown, LaTeX, code, and more. University-level course material, espe-

cially in STEM fields, It is important that the AI assistant is able to understand and extract information from all of these different formats, and preserve their semantic structure and integrity. This section discusses the implementation of parsers for the different formats.

The intelligent data extraction system for Edubotics.ai processes various academic content formats into a structured, accessible representation. A custom PDF parser converts each page into an image, processed by the GPT4o model. This extracts textual and visual content, converting it into Markdown format. Mathematical formulas are accurately extracted and formatted in LaTeX. Through specific prompting instructors can instruct the system to extract visual elements like graphs, charts, and figures, and replace them with detailed descriptions, enabling comprehensive query responses. This directly affects the quality of the responses, as shown in Figure 3·1.

The data extraction pipeline also now supports Markdown files, Jupyter notebooks, and entire GitHub repositories. This broadens compatibility with educational resources, making it adaptable to various courses and materials.

Finally, metadata for each document is automatically generated, aided by LLMs. When processing course materials, the pipeline retrieves metadata directly from the source, such as the course website. For instance, when an assignment is encountered, the system identifies relevant metadata: assignment title, numbering, release date, and due date. This metadata is then appended to the document before it is passed into the vector store for later retrieval. By embedding this additional layer of structured information, the system ensures that the conversational assistant can provide contextually rich and specific responses.

This combination of automated content parsing, multi-format compatibility, and metadata extraction creates a robust and scalable foundation for data processing, enabling Edubotics.ai to support diverse content types.

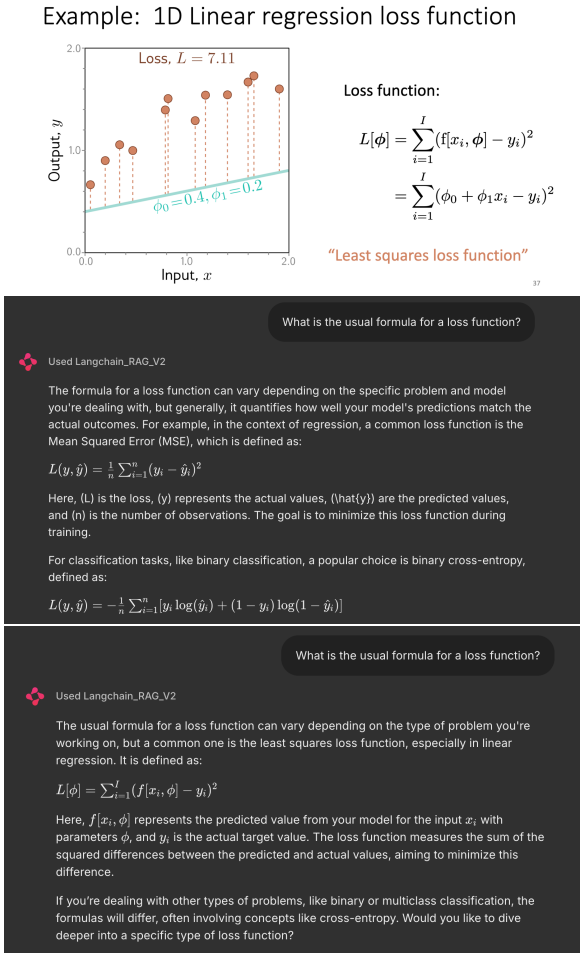

Figure 3·1: Example of an improved, more grounded response from the assistant due the correct extraction of LaTeX equations from a PDF. The source PDF (top) shows how the relevant math equation is written in the lecture material. The old response (middle), does give the correct answer, but in it the formula does not match the formula in the source, because math equations were not captured properly. After they were processed by the LLM-powered pipeline, the new response (bottom) follows the same mathematical notation as the original source PDF.

3.2.2 Effective Retrieval of Technical Course Content

Retrieval Augmented Generation (RAG) is a technique that enables an AI assistant to retrieve and use information from a knowledge base of documents (Lewis et al.,

2021). Document content is embedded using a large language model and stored in a vector database. When a user asks a question, the question is embedded and the most semantically similar texts (document chunks) to that question are retrieved. The content of these chunks is then added into the prompt to the LLM (in addition to the user's question) to provide necessary context in the generation of a response. For example, if the user asks a question about the earnings of a stock XXX, the AI assistant can retrieve the document that mentions the earnings of stock XXX and reference it in the generation of a response. Today's embeddings models are very effective at capturing the semantic meaning of words, and so the correct passage can be found even if the user's question does not use the exact same wording.

What we found is that the current prototype struggles with the retrieval of certain content, particularly with assignments. In the case of this class, the assignments are written by different people and do not follow a consistent format, like titling and annotation of equations. The course also has course website pages for assignments, which confuses the retrieval system. I set out to set up a RAG system that can handle a wide variety of course content without the need to impose a strict format.

A RAG-system generally consists of four main components: document chunks, embedding, vectorization, and retrieval. Document chunks refer to the process of dividing large documents (our source content) into smaller, more manageable pieces for processing. Embedding involves converting these chunks into dense vector representations that capture their semantic meaning. Vectorization is the process of converting these embeddings into a format suitable for storage and querying. Finally, retrieval is the process of searching for and returning the most relevant documents based on a user's query.

Experiments with different embedding models and vectorstores showed that the difference in responses was trivial. OpenAI's embeddings model "text-embedding-3-

large" consistently provided good responses. For the vectorstore, I picked FAISS, as it is a fast and efficient vectorstore that is easy to set up.

I experimented with different chunk sizes, reranking, and the use of multiple vectorstores (MVS).

MVS is a technique that enables the assistant to dynamically route queries to the most relevant vectorstore, improving precision by ensuring that retrieval focuses on contextually appropriate content. Each content type is stored in a separate vectorstore, and the assistant is able to determine which vectorstore to use based on the user's query.This is an 'agentic' approach, since the LLM assistant needs to make a decision. In the case of DS701, there are four types of content: assignments, lecture slides, discussions, and course website pages. If the student asks a question on an assignment, the assistant knows to retrieve from the assignments vectorstore, narrowing the scope of the retrieval and improving precision. The primary goal was to assess the improvements provided by the MVS approach in both quantitative retrieval performance and qualitative flexibility.

Combinations of these approaches were rigourously evaluated on a set of queries, for which the most relevant ('golden') chunks and documents were identified manually. These queries ranged from asking about general lecture material, help on assignments and course logistics, i.e. targeted at certain content types.

The naming of the scenarios reflects the configuration of the retrieval system, combining key elements of chunk size, reranking, and the use of multiple vectorstores (MVS). "OG" refers to the original setup with default chunk sizes (300 tokens) and a single vectorstore, where all source content is stored. "C-1K" denote configurations with document chunks of 1000 tokens. The number 1000 is chosen because it creates a good balance between the number and size of document chunks. The suffix "R" indicates the inclusion of a reranking step to refine retrieved results. Cohere's rerank-

Metric	OG	OG-R	OG-MVS	OG-MVS-R	C-1K	C-1K-R	C-1K-MVS	C-1K-MVS-R
Success@1	0.233	0.233	0.067	0.200	0.467	0.467	0.567	**0.633**
Success@3	0.433	0.433	0.233	0.267	**0.733**	**0.733**	**0.733**	**0.733**
Success@5	0.567	0.567	0.233	0.267	0.767	0.767	0.733	**0.833**
Recall@1	0.233	0.233	0.067	0.200	0.467	0.467	0.567	**0.633**
Recall@3	0.200	0.200	0.133	0.133	**0.452**	**0.452**	0.435	0.435
Recall@5	0.280	0.280	0.147	0.160	0.516	0.516	0.484	**0.548**

Table 3.1: Comparison of retrieval performance metrics for different configurations of the retrieval system, on a manually curated set of queries. Success@n is the percentage of queries for which at least one relevant ('golden') chunk is retrieved within the top n results. Recall@k is the average percentage of relevant ('golden') chunks (out of all relevant chunks for a query) that are retrieved within the top k results. The best performing configuration in each row is highlighted in bold.

v3.5 (Sun et al., 2023) was used for reranking. After the documents are retrieved, they are ranked by the embedding model and the vectostore. A reranking step is used to refine the ranking of the documents according to the user's query.

"MVS" specifies the use of multiple vectorstores, with one dedicated to each content type (assignments, lecture slides, discussions, etc.), and "MVS-R" combines MVS with reranking. Only documents that are of the same content type are put into the same vectorstore. When a query is made, the assistant determines which vectorstore to use based on the user's query. If the question is about an assignment, the assistant will retrieve from the assignments vectorstore.

Results of the evaluation are shown in Table 3.1. Success@k is the percentage of queries for which at least one relevant ('golden') chunk is retrieved within the top k results. Recall@k is the average percentage of relevant ('golden') chunks (out of all relevant chunks for a query) that are retrieved within the top k results. Success and recall metrics were calculated across scenarios, with notable improvements observed when using MVS combined with reranking. In particular, the C-1K-MVS-R configuration, which employed 1000-token chunks, reranking, and MVS, achieved the highest performance across several metrics. Success@1 improved significantly compared to baseline configurations, reaching 0.633—a nearly threefold increase over the

original single- retriever setup (OG, 0.233). Recall@5, which measures the model's ability to retrieve all relevant golden chunks within the top five results, demonstrated the most substantial improvement, rising to 0.548 from the baseline 0.280.

The agentic nature of MVS provides the system with a scalable framework for handling diverse and complex queries, through techniques like query rewriting, and interventions like query clarification. Integrating more agents could enable these interactions and more, like multi-step planning for Socratic assistance, where the system leads the student towards the correct answer through a series of questions.

3.3 Conclusion

In this chapter, I have described my contributions to Edubotics.ai, an open-source library of tools and modules for building and deploying multi-purpose AI assistants. I developed a data extraction pipeline that can process a wide variety of course content formats into a structured, accessible representation. Visual elements from formats like complex PDFs are properly processed, enabling more grounded and context-aware responses. The pipeline also extracts metadata from the source content, which is used to create more context-aware responses. These are key to the assistant's credibility and effectiveness in supporting the source content.

I also created a retrieval system (MVS) that showed vastly improved accuracy of document retrieval for context-requiring and technical queries, like assignments. The agentic nature of the MVS approach also supports the addition of new content categories without retraining or major reconfiguration, making it highly scalable and adaptable. This ensures the system remains robust as course materials evolve, offering long-term value. Finally, combining MVS with reranking refined results further, prioritizing the most relevant chunks within the returned documents and reducing the need for users to sift through irrelevant or less important content.

These results and discussions highlight the potential of LLM-powered pipeilnes and agentic orchestration in bridging gaps between disparate content types and aligning retrieval outputs with user intent. By integrating both quantitative metrics and qualitative insights, the chapter showed the value of agentic implementations in educational contexts.

Chapter 4

Towards a Foundational Model for Analyzing Herbarium Specimens

4.1 Introduction

Herbaria specimens are a vital resource for botanical research, specifically for studying plant morphology and phenology over space and time (Davis et al., 2015). Growing global awareness of climate change has sparked renewed interest in using herbarium-derived data to study climate change. However, the manual curation of herbarium specimens is a time-consuming and labor-intensive process, limiting the scalability of traditional methods. A significant challenge remains in mislabeled specimens, which can lead to incorrect conclusions in research (Fujii, 2019).

This has led to the development of highly accurate, automated image analysis techniques powered by deep learning and artificial intelligence.

Deep learning approaches for herbarium classification range from traditional image classification using Convolutional Neural Networks (CNNs) (Šulc and Matas, 2017; Younis et al., 2018), a CNN, called YOLO ("You Only Look Once"), previously used for object detection (Redmon et al., 2016), now adapted for plant part detection (Thompson et al., 2023), to a recent method (Stevens et al., 2024) that applies CLIP ("Contrastive Language Image Pretraining") (Radford et al., 2021), to plant species classification. . Given the widespread adoption of CNNs, YOLO, and CLIP models in computer vision and their extensive documentation across the academic literature

and online resources, this section will focus on their application to plant classification rather than providing an in-depth technical overview over these general models.

Unlike traditional image classification tasks, fine-grained Visual Classification (FGVC) focuses on distinguishing among a high number (over a thousand) of categories - species in this case - that often exhibit only subtle visual differences (Xu et al., 2023). This challenge is particularly relevant in herbaria, where dried, pressed specimens present unique visual complexities and degradation over time (Swain and Chakraborty, 2024). Images of preserved herbarium specimens also exhibit a high degree of similarity, making FGVC a critical field within computational botany, offering powerful tools to support taxonomy, phenology, and biodiversity research.

Research in the area has accelerated in recent years. (Shirai et al., 2022) developed 96.4% classification accuracy on over 2,000+ of Japanese herbarium species using the popular Inception-ResNet-v2 model (Szegedy et al., 2016). However, 2,000 species is a relatively small dataset. Kaggle competitions such as the FGVC9 (Hogan et al., 2022; Park et al., 2024) in 2022 have also been organized to benchmark models on the task of FGVC in the herbarium domain, using a dataset of over 1,000,000 images of 15,501 unique species. The dominant types of models used were Swin-Transformer (Liu et al., 2021b), DeiT (Touvron et al., 2021), and Meta-Transformers (Tan and Le, 2020). Top 5 performing teams were able to achieve F1 scores of at least 85% on the test set. The sheer size of the models used in these competitions is also noteworthy - parameter counts go up to the billions. While the results are significant, the model weights were not made available publicly, and the training details are not extensively documented.

Zero-shot classification refers to the ability of a model to identify categories or classes that it has not explicitly seen during training. This is significant for large-scale image classification tasks, where the sheer number of possible categories — such

as biological species — may make it infeasible to gather labeled data for every class. The ability of a model to generalize to new species is crucial for scalability. The hierarchical nature of species names ('taxonomic names') provide semantic information that models can leverage for better generalization. To address this challenge, BioCLIP (Stevens et al., 2024) adapted the CLIP (Radford et al., 2021) framework to classify over 40,000 biological species, spanning plants, animals, and other organisms. By leveraging CLIP's text encoder, BioCLIP demonstrated that the model inherently learns relationships between taxonomic labels, such as genus and species hierarchies, in the biological domain. This capability allows the model to generalize its understanding and classify species it has never seen before, enabling zero-shot predictions. BioCLIP achieved impressive results, with 91% zero-shot accuracy on the PlantNet dataset (1081 species) and 38.6% on the Medicinal Leaf dataset (40 species)(Royal Botanic Gardens and Domain Trust, 2024; Roopashree and Anitha, 2020) While these results highlight the potential of zero-shot classification for large-scale biological image recognition, they focus primarily on images of plants and animals 'in the wild'. Images of herbaria are much more homogenous - they involve the same background and orientation all the time.

The focus of this chapter is to:

1. Attempt to reproduce the results of the top performing team in the FGVC9 2022 competition. The results of this fine-tuning are expected to serve as a strong baseline for further explorations, such as with more specialized FGVC models. Multiple training strategies are explored to find the most effective approach.

2. Further, inspired by the success of BioCLIP, this chapter explores the potential of a hybrid model that combines the strengths of CLIP with the strengths of SWIN-Transformer. I aim to investigate whether replacing CLIP's Vision Transformer (ViT) (Dosovitskiy et al., 2021) backbone with a FGVC-specific

vision backbone would improve its performance and generalization on herbarium species.

Another one of my contributions is a parallelized data collection pipeline, which was used to collect over 5.5 million herbarium specimen images spanning over 20,000 species. These efforts are part of a larger vision to develop a collection of foundation models for herbarium analysis, aimed at enabling botanists to recognize species from images, analyze subtle visual differences, and access critical information about species origin, location, migration, morphology, and more. This work takes incremental but essential steps toward that goal, building the foundational blocks necessary for scaling herbarium specimen classification and analysis.

At the end, the chapter proposes a novel conversational assistant for herbarium specimen analysis, and discusses the potential of this approach to support the manual curation process.

4.2 Contributions

4.2.1 Dataset

The dataset used in this experiment is the NAFlora-1M dataset (Hogan et al., 2022), which contains over 1.2M images of 15,501 unique species. Each specimen includes metadata like species name, taxonomy, and collector details. Designed for large-scale fine-grained classification, the dataset highlights subtle interspecies variations, enabling advancements in biodiversity research and herbarium digitization. Exhibiting a long-tailed distribution, the dataset is well-suited for addressing the challenges of imbalanced distributions and class rarity.

4.2.2 SWIN-Transformer

The SWIN (Shifted WINdow) Transformer is a hierarchical vision transformer that processes images using shifted windows of varying sizes (Liu et al., 2021b). It contains four transformer (Vaswani et al., 2017) blocks, each operating at a different resolution. Unlike traditional vision transformers that maintain a fixed resolution throughout their layers, SWIN progressively merges image patches to create a hierarchical representation. This design allows the model to efficiently process high-resolution images while maintaining computational efficiency.

The key innovation of SWIN is its shifted window partitioning approach. In each layer, the image is first divided into non-overlapping windows where self-attention is computed locally. In subsequent layers, the window partition is shifted, allowing for cross-window information exchange. This shifting mechanism enables the model to capture both fine-grained local features and broader contextual information, which is particularly crucial for fine-grained visual classification tasks.

SWIN's architecture consists of several stages, each operating at a different resolution. As the network deepens, adjacent patches are merged, reducing spatial resolution while increasing the channel dimension. This hierarchical design mirrors the feature pyramid commonly found in convolutional neural networks, enabling the model to capture multi-scale features effectively. The computation of self-attention within local windows, rather than globally, results in linear computational complexity with respect to image size, making SWIN more scalable than traditional vision transformers.

The model also incorporates relative position encoding within each window, allowing it to better understand spatial relationships between patches. This localized approach to position encoding, combined with the shifted window mechanism, helps SWIN maintain translation invariance while being sensitive to local spatial structures

- properties that are essential for identifying subtle morphological differences between plant species in herbarium specimens.

The original SWIN-Transformer (Liu et al., 2021a) model weights are publicly available, and the model was trained on popular datasets like ImageNet-1K (1,000 classes) and ImageNet-22K (21,841 classes) (Russakovsky et al., 2015). There are multiple versions, including SWIN-Tiny, SWIN-Small, SWIN-Base, and SWIN-Large. This following experiments will focus on the SWIN-Base model, as the smaller models were found to underperform and finetuning the larger models was found to be resource intensive, specifically memory.

4.2.3 CLIP

Traditional classification models are limited to predicting a fixed set of predefined labels because they learn a direct mapping between images and those labels during training (Stevens et al., 2024). This limitation prevents them from performing zero-shot classification, where the goal is to identify categories the model has never encountered before. Zero-shot classification is especially valuable in domains like biological species recognition, where the sheer number of categories—such as plant species—makes it infeasible to label data for every class. Herbarium labels are also inherently hierarhical - or 'taxonomic' organized into structured relationships, such as genus, family, and order. This hierarchical nature introduces additional complexity but also provides semantic relationships that models can leverage for better generalization.

CLIP (Contrastive Language-Image Pretraining), developed by OpenAI, addresses this limitation by learning to associate images and textual descriptions based on their similarity (Radford et al., 2021). CLIP is trained on a massive dataset of image-caption pairs using two encoders: one for images and one for text. Through a process called contrastive learning, the model is trained to increase the similarity between

an image and its correct caption while decreasing the similarity between the image and unrelated captions. This approach allows CLIP to learn rich visual and linguistic patterns that generalize beyond its training data.

For example, during training, CLIP might be shown an image of a red fox alongside the caption 'a red fox in the forest.' The image encoder processes the image to capture its visual features, while the text encoder processes the caption to capture its semantic meaning. The model learns to associate (or increase the similarity between) this image and caption pair. Over time, it develops the ability to generalize to new pairs, such as associating the description 'a fox in the woods' with visually similar images, even if those images or descriptions were not in the training set.

In a zero-shot classification scenario, CLIP can be used to classify a completely new image by comparing its similarity to a set of textual descriptions. For instance, when presented with an image of a plant it has never seen before, CLIP can assess the similarity between the image and a set of candidate labels, such as 'a leaf from a maple tree' or 'a leaf from an oak tree.' The model selects the label with the highest similarity to the image, accurately predicting the category (e.g., 'maple tree') without requiring labeled examples for that species during training. This ability to infer from similarity makes CLIP a powerful tool for tasks with large, fine-grained taxonomies.

BioCLIP (Stevens et al., 2024) extends this framework to biological domains, demonstrating that CLIP's text encoder naturally learns hierarchical structures, such as genus-species relationships, in taxonomic data. By leveraging the similarity between visual features and descriptive labels, BioCLIP enables zero-shot classification for complex biological datasets. It has shown strong performance in large-scale tasks like plant species recognition, providing a scalable solution for domains where collecting labeled data is challenging. This makes models like CLIP and BioCLIP invaluable for advancing research in areas such as herbarium specimen analysis.

4.3 Experiments and Results

4.3.1 Recreating the FGVC9 2022 winning model

The finetuning conducted in this study was designed to attempt to recreate the results of the top-performing team in the FGVC9 Kaggle competition, which achieved an accuracy of 87.5%. Their approach utilized an ensemble of models including SWIN-Base and SWIN-Large models, with training conducted on 8 NVIDIA A100 GPUs. Top teams also employed more sophisticated training techniques, like including advanced loss functions and extensive data augmentation methods. The present study deliberately employed a more bare-bones methodology, focusing on more standard finetuning techniques. This decision was made to establish a clear, reproducible baseline for fine-grained herbarium classification and to better understand the capabilities of the SWIN-Transformer without introducing additional complexities that could obscure the analysis.

Due to resource constraints, this work focused on finetuning a single SWIN-Base model using 4 NVIDIA L40 GPUs. Despite these limitations, the results reaffirm the SWIN-Transformer's effectiveness for fine-grained visual classification (FGVC) tasks in the herbarium domain.

From here on, the original SWIN-Transformer trained on ImageNet-22K will be referred to as SWIN-Pretrained. For the purposes of finetuning on the NAFlora-1M dataset, the ImageNet-22k weights were used, and the model was modified to output 15,501 classes instead. A widely accepted approach to finetuning is to partially freeze the weights of the pre-trained model, and only update the weights of the last few layers. This allows the model to retain the knowledge it learned during pre-training and adapt it to a new task (called downstream task). Multiple freezing strategies were tested, including not freezing any layers. These finetuned SWIN models (starting from SWIN-Pretrained) will be referred to as SWIN-Finetuned.

Two following two approaches were found to be effective for SWIN-Finetuned:

- Unfreezing all layers, and updating the weights of all layers.
- Starting by unfreezing the last transformer block, then incrementally unfreezing more layers as training progressed.

These will be referred to as SWIN-Finetuned-All and SWIN-Finetuned-Incremental, respectively.

Model	**Accuracy**	**F1 Score**	**Epochs**
SWIN-Finetuned-All	70.78%	70.60%	100
SWIN-Finetuned-Incremenetal	70.62%	70.44%	100

Table 4.1: Results of the SWIN-Transformer finetuning experiments.

Both models were finetuned using Hugging Face's Transformers library. SWIN-Finetuned-All was finetuned with a batch size of 512, while SWIN-Finetuned-Partial was finetuned with a starting batch size of 128, which was decreased as training progressed. SWIN-Finetuned-All was finetuned with a Cosine Annealing learning rate scheduler, while SWIN-Finetuned-Partial was finetuned with a standard linear learning rate scheduler and an AdamW optimizer, with a weight decay of 0.05. Both models were finetuned with 4 NVIDIA L40S GPUs.

So far, both approaches have yielded an accuracy of 70% and an F1 score of 70% on the test set.

These results, while lower than those reported by the Kaggle competition team, were achieved using significantly fewer computational resources and a simpler single-model approach, underscoring the potential of SWIN-Base to serve as a strong baseline for FGVC tasks in the herbarium domain. The insights gained from this work provide a stepping stone for future explorations into larger models or ensembles, such as SWIN-Large, coupled with methods to fit these models into smaller GPU mem-

ory like quantization (Jacob et al., 2018). These include integrating domain-specific enhancements or leveraging recent advancements in foundational vision models.

Although the different finetuning approaches did not yield significant improvements, they both validate the SWIN-Transformer as a highly capable model for fine-grained herbarium classification tasks, setting the stage for further advancements in FGVC methodologies.

By confirming SWIN's strong performance and identifying its potential in the herbarium domain, this study paves the way for future research to explore larger, more specialized models. These could include multi-modal approaches, where SWIN serves as a backbone for systems integrating text, images, and metadata to expand the scope of herbarium specimen analysis. While this work provides a foundation, the possibility of exceeding 70% accuracy and achieving state-of-the-art performance will likely require additional resources and more sophisticated modeling techniques.

4.3.2 SWIN-CLIP

Since SWIN-Transformer has proven effective for herbarium specimen classification, and CLIP is well-regarded for its zero-shot classification capabilities, this section explores the potential of combining both worlds. The core strength of CLIP lies not in its specific model architecture but in its contrastive learning framework, which trains the model to align image and text embeddings by maximizing their cosine similarity. Prior research (Zhai et al., 2022) has demonstrated that mixing and matching components of the contrastive learning framework, including vision backbones and text encoders, can yield strong results across diverse tasks, outperforming original CLIP. This work builds on that insight by integrating SWIN as a vision backbone into CLIP, leveraging its hierarchical feature extraction capabilities to enhance representation learning for fine-grained herbarium species classification.

Unlike the Vision Transformer (ViT), which processes global image patches, SWIN

utilizes a hierarchical structure with shifted windows, allowing it to capture both local and global image context more effectively. This structure is especially advantageous for fine-grained classification, where subtle and localized visual features differentiate classes—such as species with minor morphological differences. In addition, by integrating SWIN as the visual backbone to CLIP, the model retains zero-shot prediction capability, allowing it to generalize to new categories without additional or with minimal training. This makes it highly suitable for complex and taxonomically diverse datasets in herbaria research.

Due to resource constraints, SWIN-CLIP was trained on subsets of the NAFlora-1M dataset, first with 100, then with 1000 classes. The model is also substantially larger than the original CLIP model, so the SWIN vision backbone was partially frozen during training. The transformer text encoder was trained fully from the "openai/clip-vit-base-patch32" checkpoint (Radford et al., 2021). The freezing strategy for the SWIN backbone is denoted by a suffix, e.g. "Finetuned vX". Here, "vX" refers to the freezing of the last X transformer blocks and beyond. So, "Finetuned v2" refers to the training of the last 2 SWIN transformer blocks and beyond, and keeping the first 2 transformer blocks frozen. In addition, two scenarios for the SWIN vision backbone were explored:

- SWIN-Finetuned ("finetuned"): The SWIN vision backbone was finetuned on the herbarium dataset. The same checkpoint used for SWIN-Finetuned in the previous section.

- SWIN-Base ("base"): The SWIN vision backbone is pretrained on ImageNet-22k, and was not finetuned on the herbarium dataset.

To form a baseline, the original CLIP model ('baseline') was also finetuned on the same set of labels. All models were trained with a batch size of 128, a learning

rate of 0.0001, and a weight decay of 0.05. The loss function used was standard Cross-Entropy. The models were trained for 40 epochs on 2 NVIDIA L40S GPUs.

Results of training the SWIN-CLIP model on a subset of 100 labels of the NAFlora-1M dataset are shown in Figure 4·2. Across all SWIN-backbone configurations ("finetuned" or "base"), the final validation accuracy during training (blue bars) demonstrates performance comparable to the baseline CLIP model ("baseline"). However, differences in performance emerge when specific configurations are compared. For instance, in the "base v2" and "finetuned v2" configurations, where the last two transformer blocks are unfrozen, SWIN-CLIP with the ImageNet-22k pretrained backbone outperforms the finetuned SWIN backbone. By contrast, unfreezing an additional transformer block ("v3" configurations) leads to the opposite result: the finetuned backbone achieves the highest accuracy, slightly outperforming the baseline CLIP model.

Interestingly, SWIN-CLIP consistently outperforms baseline CLIP in zero-shot accuracy (red bars), underscoring the robustness of the SWIN backbone for generalization tasks. The "base v3" configuration achieves the highest zero-shot accuracy among all models, with a top-1 accuracy of 20%, compared to 4% for baseline CLIP. This substantial improvement suggests that the SWIN backbone enhances CLIP's capacity for feature representation in fine-grained, domain-specific categories, even without labeled training data.

These results highlight several key trends. First, the performance trade-offs between pretrained and finetuned backbones suggest that domain-specific fine-tuning can significantly enhance task-specific accuracy, especially when sufficient transformer blocks are unfrozen. Second, the significant improvements in zero-shot accuracy achieved by SWIN-CLIP indicate that integrating SWIN into the CLIP architecture may address inherent limitations in CLIP's feature representation for fine-grained

categories.

While these findings are promising, they are based on a small subset of the NAFlora-1M dataset and may not fully reflect performance on the entire dataset. Future work should validate these results on larger sets of species (labels), explore additional SWIN configurations, and investigate the effects of advanced training techniques, such as domain-specific augmentations or specialized loss functions. Further scaling of the dataset and computational resources (such as training the SWIN backbone fully) may unlock greater performance.

4.4 Future Directions

The SWIN-CLIP model holds substantial promise for advancing zero-shot classification accuracy across the extensive herbarium species dataset, encompassing 15,501 classes. While finetuning the SWIN model on herbarium images has already yielded notable improvements in fine-grained classification, preliminary experiments integrating SWIN-CLIP demonstrate that its zero-shot capabilities outperform those of conventional CLIP. This underscores the potential for SWIN-CLIP to better generalize to unseen species, an essential feature given the frequent inclusion of rare or newly discovered specimens in herbaria. Future work will focus on further evaluating and refining this integration, such as with other text decoder models like BERT.

Building on these advancements, I propose the development of a multimodal conversational assistant tailored for herbarium specimen analysis, inspired by LLaVa-Med (Li et al., 2024). This system will leverage the visual understanding capabilities of fine-tuned SWIN-based models alongside large language models to provide researchers and botanists with a seamless tool for querying specimen data. A critical component of this effort involves generating high-quality visual descriptions of herbarium species to enhance the model's training data. By combining visual and textual data, the as-

sistant will be equipped to answer complex queries, provide detailed descriptions, and support taxonomic research. This novel direction represents a significant step toward bringing cutting-edge AI capabilities into the field of botanical science, fostering new discoveries and improving accessibility to herbaria collections worldwide.

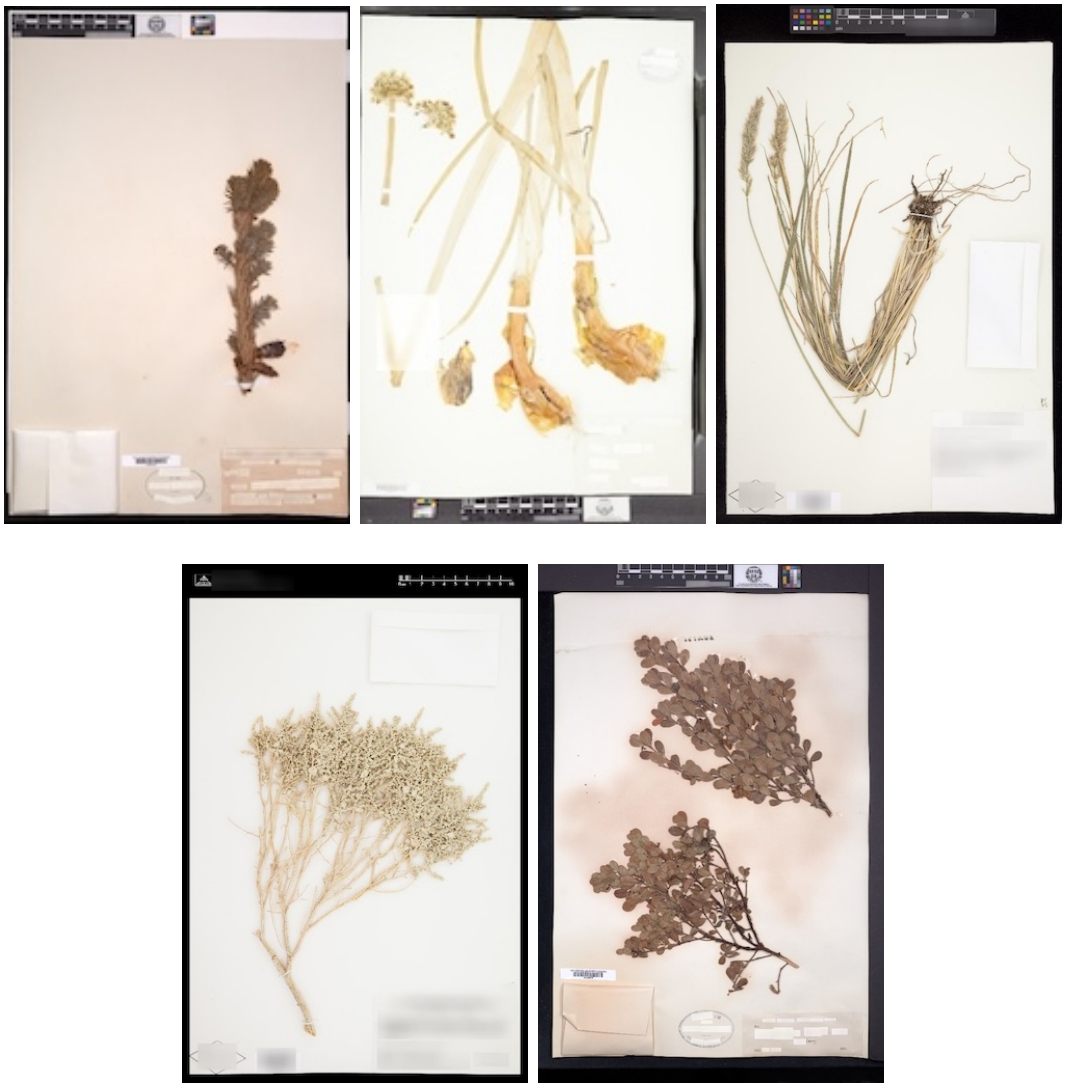

Figure 4·1: Example images from the NAFlora-1M dataset showing the diversity and complexity of herbarium specimens.

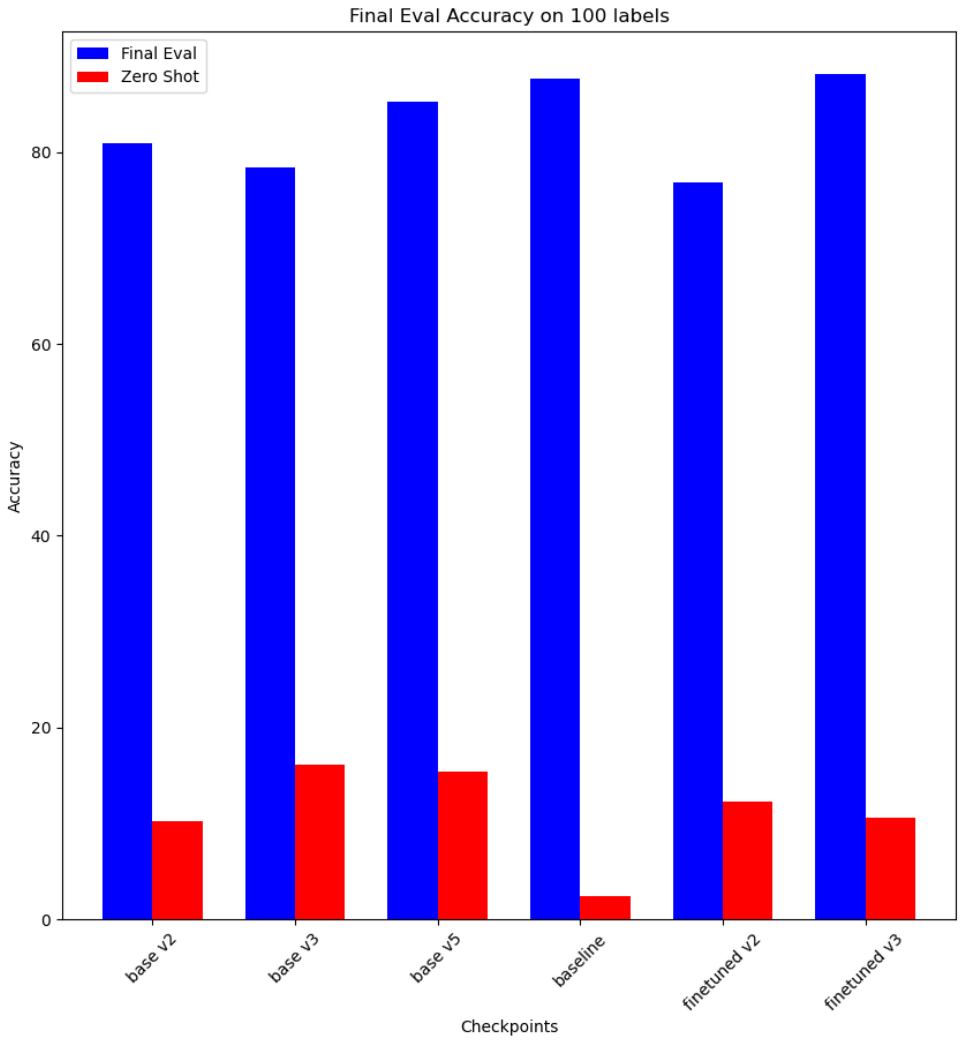

Figure 4·2: Comparison of training validation (blue) and zero-shot (red) accuracies on 100 labels of the NAFlora-1M dataset between multiple configurations of SWIN-CLIP ('finetuned ...' and 'base ...'), and original CLIP ('baseline')

Chapter 5

Conclusions

This thesis explored the development of AI systems across three distinct domains: assistive technology, botanical research, and academic engagement, each aimed at addressing unique challenges while advancing the integration of AI into specialized applications. Through the design and implementation of practical, user-focused solutions, this work contributes to the broader goal of creating accessible, scalable, and impactful AI systems.

CameraMouseAI and KeyGlide—were developed to empower individuals with severe motor impairments. CameraMouseAI extended traditional head-controlled mouse replacement systems by incorporating customizable gesture-based selection mechanisms, tailored to users' needs. KeyGlide introduced a motion-based text input interface that leveraged predictive text to enhance typing speed and reduce cognitive effort. Empirical evaluations of both systems demonstrated their usability and effectiveness, highlighting their potential to improve digital accessibility.

For botanical research, the thesis focused on fine-grained classification tasks in the herbarium domain. Using the SWIN-Transformer architecture, models were fine-tuned on a dataset containing over 15,000 species, achieving significant accuracy in species identification. Building on this, a hybrid model—SWIN-CLIP—was investigated, integrating SWIN's fine-grained visual capabilities with CLIP's zero-shot learning framework. Results showed that SWIN-CLIP outperformed baseline models in zero-shot tasks, underscoring its potential for scaling herbarium specimen analysis

without requiring extensive labeled data. These advancements provide a foundation for developing a broader, multimodal herbarium model capable of supporting tasks such as morphological analysis, species migration tracking, and ecological studies.

In education, the thesis introduced Edubotics.ai, a platform for deploying academic conversational assistants. This system combines intelligent data extraction pipelines, advanced retrieval techniques, and long-term personalization to adapt seamlessly to diverse course content. The platform processes a variety of content formats, such as PDFs, Jupyter notebooks, and GitHub repositories, enabling the generation of contextually aware and accurate responses. The results demonstrated the platform's scalability and effectiveness in academic environments, with promising applications in providing personalized student support and improving engagement.

Looking forward, several directions emerge for future research. For CameraMouseAI and KeyGlide, incorporating adaptive calibration during the initial setup could enhance usability by tailoring functionality to individual users' movement patterns, while exploring more intuitive key arrangements may further accelerate letter acquisition in text input tasks. In herbarium recognition, developing a LLaVa-powered conversational assistant would bridge image recognition with detailed contextual and ecological analysis. For Edubotics.ai, advanced agentic orchestration for retrieval, query processing, and data collection could enable multi-step, and more context-aware interactions, like Socratic assistance.

Overall, the contributions of this thesis demonstrate the versatility and potential of AI in addressing diverse and impactful challenges. By balancing technical rigor with practical applications, this work lays a foundation for future advancements that can further integrate AI into accessibility, research, and education to deliver meaningful societal benefits.

Appendix A

Proof of xyz

Nothing here yet.

References

Alagarsamy, S., Reddy, S. A., Reddy, V. V., Reddy, V. B., and Reddy, Y. V. P. (2022). Control the movement of mouse using computer vision technique. In *2022 6th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 395–399, Tamil Nadu, India. IEEE, ICECA.

Betke, M., Gips, J., and Fleming, P. (2002). The camera mouse: visual tracking of body features to provide computer access for people with severe disabilities. *IEEE Transactions on neural systems and Rehabilitation Engineering*, 10(1):1–10.

Branton, W. (2024). Custom scroll bar extension.

Davis, C. C., Willis, C. G., Connolly, B., Kelly, C., and Ellison, A. M. (2015). Herbarium records are reliable sources of phenological change driven by climate and provide novel insights into species' phenological cueing mechanisms. *American Journal of Botany*, 102(10):1599–1609.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale.

Feng, W., Sameki, M., and Betke, M. (2018). Exploration of assistive technologies used by people with quadriplegia caused by degenerative neurological diseases. *International Journal of Human–Computer Interaction*, 34(9):834–844.

Feng, W., Zou, J., Kurauchi, A., Morimoto, C. H., and Betke, M. (2021). Hgaze typing: Head-gesture assisted gaze typing. In *ACM Symposium on Eye Tracking Research and Applications*, ETRA '21 Full Papers, New York, NY, USA. Association for Computing Machinery.

Fu, Y. and Huang, T. S. (2007). hmouse: Head tracking driven virtual computer mouse. In *2007 IEEE Workshop on Applications of Computer Vision (WACV '07)*, pages 30–30, Austin, Texas. WACV.

Fujii, S. (2019). An examination of confidence in open data of specimens: Cuscuta australis (convolvulaceae). *Journal of the Botanical Research Institute of Texas*.

Goel, A. K. and Joyner, D. A. (2017). Using ai to teach ai: Lessons from an online ai class. *AI Magazine*, 38(2):48–58.

Haleem, A., Javaid, M., Qadri, M. A., and Suman, R. (2022). Understanding the role of digital technologies in education: A review. *Sustainable Operations and Computers*, 3:275–285.

Hogan, B., damon, inversion, Park, J., and de Lutio, R. (2022). Herbarium 2022 - fgvc9. https://kaggle.com/competitions/herbarium-2022-fgvc9. Kaggle.

Holmes, W. and Tuomi, I. (2022). State of the art and practice in ai in education. *European Journal of Education*, 57(4):542–570.

Huang, C.-N., Chen, C.-H., and Chung, H.-Y. (2006). Application of facial electromyography in computer mouse access for people with disabilities. *Disability and Rehabilitation*, 28(4):231–237. PMID: 16467058.

Hussain, A. (2017). Ada-Bolton College's latest digital assistant. *Blogtext*.

Hwang, G.-J. and Chang, C.-Y. (2021). A review of opportunities and challenges of chatbots in education. *Interactive Learning Environments*.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., and Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713.

Jacob, R. J. K. (1990). What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, page 11–18, New York, NY, USA. Association for Computing Machinery.

Kalabarige, L. R., Abhilash, K. A., Trivedi, K. A., and Dathatreya, M. (2023). Facial landmark-based cursor control and speech-to-text system for paralyzed individuals. In *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, pages 849–856, Tamil Nadu, India. IEEE, ICSCDS.

Karimli, F., Yu, H., Jain, S., Akosah, E. S., Betke, M., and Feng, W. (2024). Demonstration of cameramouseai: A head-based mouse-control system for people with severe motor disabilities. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '24, New York, NY, USA. Association for Computing Machinery.

Kurauchi, A., Feng, W., Morimoto, C., and Betke, M. (2015). Hmagic: head movement and gaze input cascaded pointing. In *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '15, New York, NY, USA. Association for Computing Machinery.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. (2021). Retrieval-augmented generation for knowledge-intensive nlp tasks.

Li, C., Wong, C., Zhang, S., Usuyama, N., Liu, H., Yang, J., Naumann, T., Poon, H., and Gao, J. (2024). Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021a). Swin transformer: Hierarchical vision transformer using shifted windows.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021b). Swin transformer: Hierarchical vision transformer using shifted windows.

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M., Lee, J., Chang, W.-T., Hua, W., Georg, M., and Grundmann, M. (2019). Mediapipe: A framework for perceiving and processing reality. In *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019*, Long Beach, CA. CVPR.

Magee, J. J., Epstein, S., Missimer, E. S., Kwan, C., and Betke, M. (2011). Adaptive mouse-replacement interface control functions for users with disabilities. In Stephanidis, C., editor, *Universal Access in Human-Computer Interaction. Users Diversity. UAHCI 2011. Lecture Notes in Computer Science*, volume 6766, pages 332—-341. Springer, Berlin, Heidelberg.

Nij Bijvank, J. A., Hof, S. N., Prouskas, S. E., Schoonheim, M. M., Uitdehaag, B. M. J., van Rijn, L. J., and Petzold, A. (2022). A novel eye-movement impairment in multiple sclerosis indicating widespread cortical damage. *Brain*, 146(6):2476–2488.

Park, J., de Lutio, R., Rappazzo, B., Ambrose, B., Michelangeli, F., Watson, K., Belongie, S., and Little, D. (2024). NAFlora-1m: Continental-scale high-resolution fine-grained plant classification dataset. *Journal of Data-centric Machine Learning Research*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Huang, P., and SQA, e. a. (2021). Learning transferable visual models from natural language supervision. *arXiv*, 21(2).

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection.

Roopashree, S. and Anitha, J. (2020). Medicinal Leaf Dataset.

Royal Botanic Gardens and Domain Trust (2024). Plantnet (the nsw plant information network system).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Shirai, M., Takano, A., Kurosawa, T., Inoue, M., Tagane, S., Tanimoto, T., Koganeyama, T., Sato, H., Terasawa, T., Horie, T., Mandai, I., and Akihiro, T. (2022). Development of a system for the automated identification of herbarium specimens with high accuracy. *Scientific Reports*, 12(1):8066.

Stevens, S., Wu, J., Thompson, M. J., Campolongo, E. G., Song, C. H., Carlyn, D. E., Dong, L., Dahdul, W. M., Stewart, C., Berger-Wolf, T., Chao, W.-L., and Su, Y. (2024). Bioclip: A vision foundation model for the tree of life.

Su, M.-C., Su, S.-Y., and Chen, G.-D. (2005). A low-cost vision-based human-computer interface for people with severe disabilities. *Biomedical Engineering: Applications, Basis and Communications*, 17(06):284–292.

Šulc, M. and Matas, J. (2017). Fine-grained recognition of plants from images. *Plant Methods*, 13(1):115.

Sun, W., Chen, Z., Ma, X., Yan, L., Wang, S., Ren, P., Chen, Z., Yin, D., and Ren, Z. (2023). Instruction distillation makes large language models efficient zero-shot rankers. *arXiv preprint arXiv:2311.01555*.

Swain, H. and Chakraborty, K. (2024). Science behind herbarium and its importance in recent years. *Nordic Journal of Botany*, n/a(n/a):e04499.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning.

Tan, M. and Le, Q. V. (2020). Efficientnet: Rethinking model scaling for convolutional neural networks.

Thompson, K. M., Turnbull, R., Fitzgerald, E., and Birch, J. L. (2023). Identification of herbarium specimen sheet components from high-resolution images using deep learning. *Ecology and Evolution*, 13(8):e10395. e10395 ECE-2023-05-00833.R1.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. (2021). Training data-efficient image transformers distillation through attention.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Vazquez-Li, J., Pierson Stachecki, L., and Magee, J. (2016). Eye-gaze with predictive link following improves accessibility as a mouse pointing interface. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '16, page 297–298, New York, NY, USA. Association for Computing Machinery.

Waber, B. N., Magee, J. J., and Betke, M. (2005). Fast head tilt detection for human-computer interaction. In Sebe, N., Lew, M., and Huang, T. S., editors, *Computer Vision in Human-Computer Interaction*, pages 90–99, Berlin, Heidelberg. Springer Berlin Heidelberg.

Xu, Q., Wang, J., Jiang, B., and Luo, B. (2023). Fine-grained visual classification via internal ensemble learning transformer. *IEEE Transactions on Multimedia*, 25:9015–9028.

Younis, S., Weiland, C., Hoehndorf, R., Dressler, S., Hickler, T., Seeger, B., and Schmidt, M. (2018). Taxon and trait recognition from digitized herbarium specimens using deep convolutional neural networks. *Botany Letters*, 165(3-4):377–383.

Zhai, X., Wang, X., Mustafa, B., Steiner, A., Keysers, D., Kolesnikov, A., and Beyer, L. (2022). LiT: Zero-Shot Transfer with Locked-image text Tuning.

CURRICULUM VITAE

Farid Karimli

Coming soon.