A DESKTOP APPLICATION TO ASSIST IN DETECTING AND PREDICTING ALZHEIMER'S DISEASE USING FACIAL RECOGNITION AND BRAINWAVE MONITORING

Haokun Chen [1], Rodrigo Onate [2]

[1] Stanford Online High School, 415 Broadway Academy Hall, Floor 2, 8853, 415 Broadway, Redwood City, CA 94063
[2] California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

The project aims to address the lack of convenient tools to identify early signs of Alzheimer's disease [4]. Current tools are challenging to utilize, costly, and inaccessible to many. Our proposed solution is a straightforward computer program utilizing face recognition and brainwave data to detect potential early signs of Alzheimer's. The solution utilizes MediaPipe to analyze facial movements, a Muse2 EEG headband to record data, and OpenAI's API to analyze data using prompts [5]. We encountered challenges with low-quality EEG data and required improved prompts and stable face tracking. We conducted two experiments to test how effective the prompts were and how well MediaPipe performed, and both demonstrated that the system is responsive to a set of words and the user's states. Our proposal is affordable, easily scalable, and user-friendly for non-tech users relative to alternative services. The program assists in keeping track of cognitive functions, making screening for Alzheimer's more accessible using readily available technology.

KEYWORDS

Alzheimer's Disease, Face Recognition, Brainwave Analysis, Prompt Engineering

1. INTRODUCTION

Alzheimer's disease is the most prevalent form of dementia, affecting over 55 million people globally. Typically, the onset occurs at age 65 or older, although sometimes people younger than 65 years old can be affected. The unpredictability contributes to the concern of potentially developing Alzheimer's disease.

It is crucial to predict when one might develop Alzheimer's disease because early awareness allows them to take steps to delay it from happening by changing their life styles [6]. Many people in our community have Alzheimer's disease or worry about developing it. However, there is no easily accessible approach to predict when a person will develop Alzheimer's disease.

Selkoe's strategy uses biomarkers, i.e., amyloid-beta deposits and tau tangles, revealed by PET scans and analysis of cerebrospinal fluid, to diagnose Alzheimer's at a preclinical stage. Highly accurate in a clinical set-up, these methods are, however, costly, invasive, and require special

David C. Wyld et al. (Eds): SIGI, CSTY, AI, NMOCT, BIOS, AIMLNET, MaVaS, BINLP – 2025
pp. 275-285, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151922

hardware and expertise. They are unsuitable for mass screening or early screening of a populace. Our project addresses this issue by making early diagnosis possible through non-invasive methods including analysis of facial features and EEG through consumer-grade equipment and open-source programs, making cognitive health evaluation more accessible [7].

Dauwels et al. considered applying machine learning methods, i.e., Support Vector Machines, to EEG classification and detection of Alzheimer from spectral features. Their work demonstrated promising diagnosis but required large labeled training data, along with fixed model output. In our system, live prompt engineering using OpenAI is applied to dynamically interpret EEG data, without requiring curated training data and possibly having flexible context-dependent analysis [8].

The Montreal Cognitive Assessment, created by Nasreddine et al., has been a popular cognitive screening instrument. However, it is manual, subjective, and not digitally integrated. Our system screens automatically via AI-driven analysis of biometrical inputs, which is objective, reproducible, and scalable for longitudinal monitoring of cognition.

In this project, we have created a desktop application to screen Alzheimer’s disease by leveraging brainwave data and facial recognition. To detect whether one has Alzheimer’s disease, we track the patient’s head and eye movement. “Alzheimer’s patients tend to make eye movements that lead to the target and often make anticipatory saccades in the direction of target motion. Additionally, tracking often trails target motion, causing patients to make significantly more compensatory saccades” (Molitor et al.) and we utilize this trait to track their eye and head movement with MediaPipe from Google.

To predict when a person will develop Alzheimer’s disease, we track the patient’s brainwave data and analyze it with OpenAI algorithms. Consumer grade Muse2 headband is used to track down a patient's brainwave data for one minute. Once the detection is completed, the headband saves the data as a text file and send it to OpenAI, where AI Algorithms analyze the data [9].

In the first experiment, we tested the effect of different prompt styles on the consistency of AI interpretations of EEG readings. The goal was to determine which prompt formats best guided the AI in generating medically sound and coherent summaries without attempting to make a diagnosis. Ten prompt variations were tested on the same EEG data and the accuracy for each was measured. Prompts with disclaimers and showing a clear educational intent performed best. In the second experiment, we tested MediaPipe's ability to follow eye movement under less-than-optimal conditions, including when there was partial facial occlusion. The standard deviation in eye placement was calculated for ten subjects. One subject had a significantly higher deviation, likely caused by movement, facial angle, or occlusion. Both experiments highlighted the critical role of environmental and linguistic clarity in determining data interpretation and system performance. This showed that both prompt engineering and user conditions needed to be optimized to ensure the effective and accuracy of the system.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Accuracy

In developing the face recognition technology, we encountered limitations in accuracy, attributed to the large distance between the individual's face and the camera. To address this challenge, we propose to utilize the FaceMesh technology provided by Google's MediaPipe to effectively track and monitor face and eye movements with high accuracy relative to traditional approaches. A series of mathematical computations can be utilized to calculate the distances between various key facial landmarks, such as the center of the eye, the sides of each eye, the middle of the face, and the edges of the face. These methodologies facilitates precise detection and measurements of eye and facial movement.

2.2. Connectivity

The brainwave analysis function, which is the central component of our project, encountered significant challenges due to the connectivity of the headband. These connectivity issues hindered us from gathering and analyzing the data effectively. Further, stringent privacy protocol poses additional challenges in identifying sufficient data points for our analysis. To mitigate these challenges, we look into upgrading to a more advanced headband with a higher resolution capacity. This would enable us to gather more data points, thereby improving the overall quality of our analysis. Further, by leveraging prompt engineering practices, we can incorporate data comparisons from comparable sources. With these methodologies, we can compare brainwave activity against more accurate benchmarks, while remain compliant with the necessary privacy protocol.

2.3. Prompt engineering

Prompt engineering is the crucial and essential part of the process of data analysis refinement within the scope of the project. The prompt was crafted and refined to ensure that the OpenAI model is presented with questions that are both clear and highly focused on the topic of Alzheimer's detection. The carefully crafted questions make a significant improvements in the AI's ability to differentiate between the subtle brainwave patterns or facial expressions that are linked directly to the symptoms of Alzheimer's disease. In addition, we propose to execute several different versions of prompts to test the specific prompts that yield the most relevant and actionable output. The approach gives us high confidence that not only the model's output is correct, but it's also relevant in context, particularly in terms of making accurate diagnoses.

3. SOLUTION

The three major components are Face Recognition, Brainwave Analysis, and Prompt Engineering [10]. When we open the application, a landing screen appears with 2 buttons that say “Eyes Tracking Interface”, which does brainwave analysis, and “Brainwave Interface”, which does the brainwave analysis and uses prompt engineering to analyze the data. When the user clicks “Eyes Tracking Interface”, a button called “Use Webcam” appears to direct the test subject to look at the red dot when the video starts. We use face recognition to detect symptoms of Alzheimer’s disease. Alzheimer’s disease patients tend to move their head and the eye in the same direction when they are looking at an item, so we will track their head and eye movements. If they cannot keep their eyes in one spot, a message appears “Indicators of simultaneous movement have been detected, which align with Alzheimer’s symptoms.” Otherwise, it says “Assessment complete: No symptoms of Alzheimer’s detected.” We then predict when the test subject will get Alzheimer’s disease by brainwave analysis. When the user clicks “Brainwave Interface”, the screen will pop up with a button called “Start”. After clicking on it, it will connect to the headband, which takes up to 10 seconds. We track the brainwave data with the headband for one minute, and then store the data in a text file. The text field will be sent to OpenAI to be analyzed by a neurologist assistant using the gpt4o model, which returns a response that is displayed on the application for the user to see.

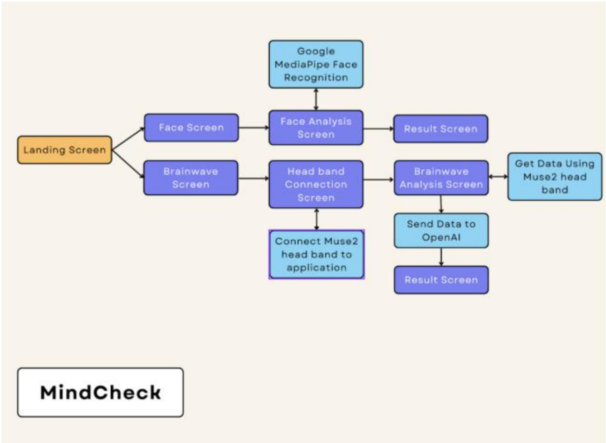


Figure 1. Overview of the solution

Face recognition is used to track eye and face movement to detect potential symptoms of Alzheimer's. We utilized this system using FaceMesh from MediaPipe, in addition to mathematical calculations that find the distance between the pupil and the eye edge and the nose center. This module relies on computer vision, which allows us to analyze facial landmarks to identify irregularities in the movement patterns. Within our program, this component functions by continuously capturing and processing facial data, allowing us to analyze deviations that may indicate cognitive decline.

Figure 2. Screenshot of face recognition

Figure 3. Screenshot of code 1

The start_tracking function leverages OpenCV and MediaPipe to perform real-time video capture and facial landmark tracking [11]. It begins by initializing video input from a specified path and retrieving the frame dimensions. Several variables are set up, including deques for storing iris and nose coordinates, a blink counter, and directional tracking flags. When the video starts, the interface prompts the user to focus on a red dot by updating a display label. The core loop runs for a fixed duration, continuously reading frames from the video stream. If a frame cannot be read, the loop terminates early. Each frame is optionally resized and converted to RGB format for compatibility with MediaPipe processing. The MediaPipe Face Mesh model is then applied to extract facial landmarks. A red circle is drawn on the screen at a specific target point to guide the user's gaze. This function is part of a broader system for facial feature recognition and gaze-based interaction, designed to improve real-time user engagement and provide immediate visual feedback during tracking.

One of the core components of our system is brainwave analysis, designed to capture and interpret EEG signals in real time. Its primary purpose is to monitor cognitive activity and extract meaningful patterns from brain data. To implement this, we used the Muse 2 headband, which records four EEG data channels simultaneously. These signals are visualized live on the user interface using Matplotlib, allowing both users and researchers to observe fluctuations in brain activity. After a one-minute recording session, the system saves the EEG data as a text file and sends it to OpenAI's Large Language Model (LLM) for interpretation. This step relies on the concept of LLM-based reasoning, where the model is prompted to analyze patterns, anomalies, or correlations within the data—sometimes in combination with contextual metadata such as task type or emotional cues.

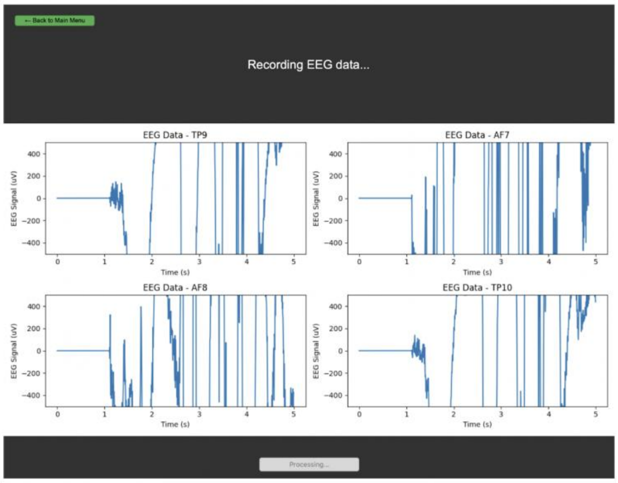
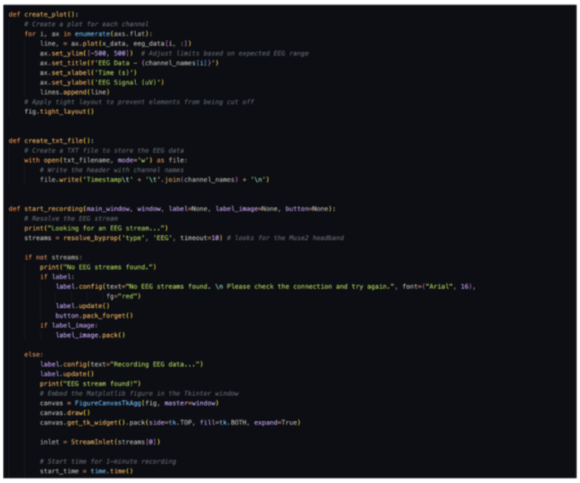


Figure 4. Screenshot of EEG data



```python
def create_plot():
    # Create a plot for each channel
    for i, ax in enumerate(axs.flat):
        line, = ax.plot(x_data, eeg_data[i, :])
        ax.set_ylim([-500, 500])  # Adjust limits based on expected EEG range
        ax.set_title(f'EEG Data - {channel_names[i]}')
        ax.set_xlabel('Time (s)')
        ax.set_ylabel('EEG Signal (uV)')
        lines.append(line)
    # Apply tight layout to prevent elements from being cut off
    fig.tight_layout()


def create_txt_file():
    # Create a TXT file to store the EEG data
    with open(txt_filename, mode='w') as file:
        # Write the header with channel names
        file.write('Timestamp\t' + '\t'.join(channel_names) + '\n')


def start_recording(main_window, window, label=None, label_image=None, button=None):
    # Resolve the EEG stream
    print("Looking for an EEG stream...")
    streams = resolve_byprop('type', 'EEG', timeout=10) # looks for the Muse2 headband

    if not streams:
        print("No EEG streams found.")
        if label:
            label.config(text="No EEG streams found. \n Please check the connection and try again.", font=("Arial", 16),
                         fg="red")
            label.update()
            button.pack_forget()
        if label_image:
            label_image.pack()

    else:
        label.config(text="Recording EEG data...")
        label.update()
        print("EEG stream found!")
        # Embed the Matplotlib figure in the Tkinter window
        canvas = FigureCanvasTkAgg(fig, master=window)
        canvas.draw()
        canvas.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH, expand=True)

        inlet = StreamInlet(streams[0])

        # Start time for 1-minute recording
        start_time = time.time()
```

Figure 5. Screenshot of code 2

The Python code in Figure 5 processes the EEG data through a GUI [12]. It consists of three main functions: `create_plot`, `create_txt_file`, and `start_recording`. `create_plot` plots each EEG channel by cycling through the data and automatically establishing reasonable y-axis limits for anticipated EEG signal amplitudes. It labels the x-axis as time and the y-axis as the corresponding EEG signal for a channel, ensuring that the layout is fixed so nothing is cut off. The `create_txt_file` function outputs a text file to store the EEG data, printing a header with channel names that are time stamped for easier categorization. The `start_recording` function starts the recording by resolving the EEG stream from the Muse2 headband. It looks for available streams, updates the GUI to notify the user of the connection status, and starts the recording if a stream is available, adding a Matplotlib figure into the Tkinter window for real-time visualization.

The Prompt Engineering section translates brainwave data from the Muse2 headband into structured insights. It processes EEG data through OpenAI's API and translates it into a medical-grade report. It relies on Natural Language Processing (NLP) to translate raw data into actionable text [13]. It acts as the final stage of Brainwave Analysis to give users clear comprehension in terms of cognitive wellness. It helps to replicate a doctor's diagnosis and make complex brainwave data easy to read.

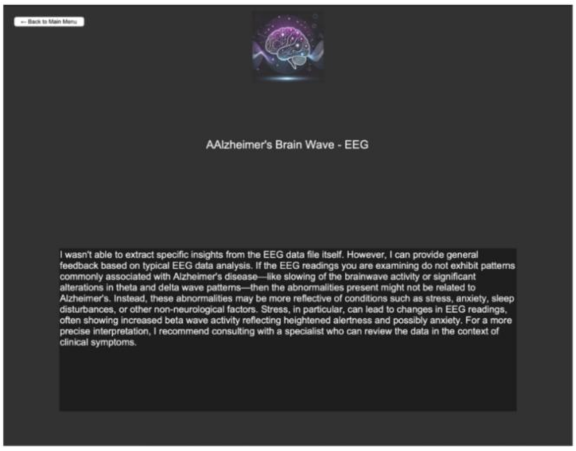


Figure 6. Screenshot of Brain Wave

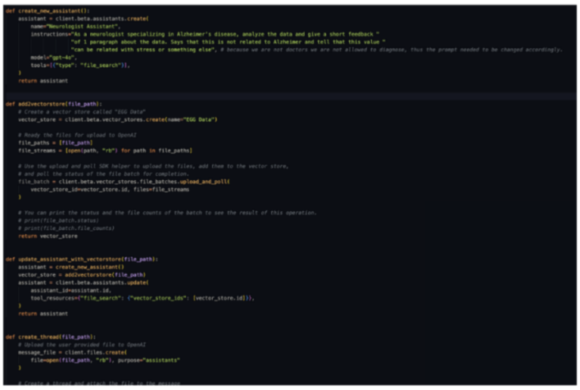


Figure 7. Screenshot of code 3

The code demonstrates how to build an AI assistant that searches for information on Alzheimer's disease through OpenAI's API. It begins with the `create_new_assistant()` function to set up an assistant by the name "Neurologist Assistant." This assistant has special conditions to read through data and provide feedback but is not allowed to diagnose any disease since it is against the law. It uses `add2vectorstore(file_path)` to establish a vector store by the name "EGG Data." It organizes files to be uploaded and facilitates easier uploading through the OpenAI API. It ensures that files are organized for easy access. It uses `update_assistant_with_vectorstore(file_path)` to link the assistant to the vector store for better capabilities. Finally, it uses `create_thread(file_path)` to enable users to easily upload files and communicate with the assistant directly. This is a simple way to make sure that the assistant can read through and analyze crucial medical data in a correct manner.

4. EXPERIMENT

4.1. Experiment 1

One possible blind spot in our program is the small number of EEG detectors available—just four—compared to specialized machines with more than 60. This means that timely accuracy is essential to guarantee the AI delivers useful insights without misinterpretation. A well-crafted prompt ensures maximum data reliability in spite of these limitations.

The objective of this experiment is to find out the impact of different prompts on the accuracy and reliability of the AI analysis of EEG data. The EEG dataset uploaded serves as a control so that all the prompts are tested on the same input. Ten different versions of the prompt were created, each changing the language but maintaining the core purpose, from levels of specificity to different framing styles. The scoring system uses Expected Accuracy (pre-programmed based on how well a prompt can get the AI to a non-diagnostic response) and Actual Accuracy (tested subsequently based on coherence and constraint satisfaction). The prompts are run against the EEG dataset, and the results are recorded and graded. The AI is tested for its ability to avoid medical diagnoses without providing related information. The final comparison of expected and actual accuracy determines the prompt structures guiding the AI to the desired analysis.

Prompt Tested (score out of 10, 1 being the least accurate)	Expected Acc	Actual Acc
As a neurologist, analyze the EEG data and provide a brief summary of any notable observations. Avoid making any medical diagnoses.	8	8
Analyze the EEG data and provide a short explanation. Emphasize that variations in the data could be due to stress or environmental factors rather than a specific disease.	9	9
Analyze the EEG data and compare it to typical healthy brain activity. Provide a general explanation while avoiding medical conclusions.	7	6
Explain the EEG data in simple terms that a non-expert could understand. Avoid any mention of disease and focus on general patterns.	6	7
Examine the EEG data and discuss possible non-medical factors that might influence the patterns. Clearly state that this analysis is not a diagnosis.	9	9
Analyze the EEG data and explain whether it appears within a normal range. If any irregularities are found, suggest common non-medical reasons they might occur.	8	7
Provide a scientific analysis of the EEG data without making any medical conclusions. Discuss possible influences such as stress, fatigue, or external stimuli.	9	8
Analyze the EEG data and describe any interesting features. Include a disclaimer that this is an AI-generated summary and not a medical opinion.	10	10
Review the EEG data and summarize any patterns observed. Suggest that a medical professional should be consulted for further insights, if necessary.	7	8
Analyze the EEG data while considering external influences such as sleep quality, stress, or recent activity. Provide a general explanation without implying medical conditions.	8	9

Figure 8. Table of experiment 1

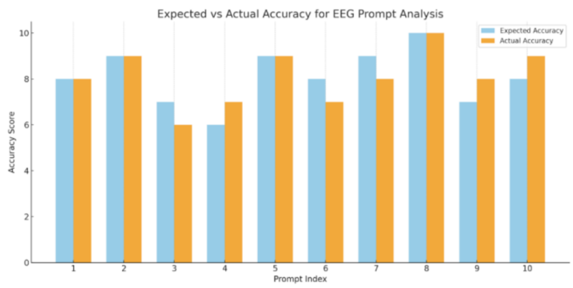


Figure 9. Expected vs Actual Accuracy for EEG Prompt Analysis

Averaging the ten prompts, both expected and actual mean was 8.1, and median expected and actual was 8. Minimum score was 6, for comparing EEG data to healthy normal brain function, and maximum was 10. Most striking was that the "explain in simple terms" prompt did better than expected (6 expected vs. 7 actual), and that the "compare to healthy activity" prompt did worse. These exceptions indicate that specificity and precise language—especially prompts with express disclaimers, sensible scope, and medical context—significantly affect performance. Prompts that explicitly delimit expectations, disclaim diagnostic intent, and claim educational or informational intent enable the model to generate more focused, accurate output. Vague or open prompts, on the other hand, tend to create interpretive uncertainty that discounts effectiveness. In general, the close correspondence between expected and observed median scores demonstrates good calibration, with the outliers showing how even subtle differences in prompt design can significantly affectthe results.

4.2. Experiment 2

One potential blind spot in the program lies in MediaPipe's limitations when detecting and tracking eye landmarks under partial occlusion or head tilts. This is a critical concern, as precise eye tracking is essential for applications such as Alzheimer's detection, attention monitoring, and gaze estimation. If the system fails to perform reliably when the user briefly looks to the side or momentarily obstructs their face—due to a hand gesture, eyewear, or natural movement—it undermines both the accuracy and credibility of the overall system. Addressing these edge cases is vital to ensure robustness and real-world applicability in diverse user scenarios.

The test shows a red mark in the middle of the screen and requests the user to focus on it for 10–15 seconds. MediaPipe is used to continuously find the inner corners of the eye and store their locations. It seeks to identify the capability of the user to maintain consistent attention in their eyes, something commonly influenced by patients with Alzheimer's. Eye movement is diagnosed using standard deviation of eye position over time. Stable attention is indicated by low deviation, and frequent alteration or high variance indicates inability to maintain focus. This tool provides a useful and simple method to determine onset of Alzheimer's by assessing vision focus without requiring invasive or sophisticated equipment.

Subject	Standard Deviation of Eye Position
1	0.032
2	0.048
3	0.020
4	0.075
5	0.059
6	0.043
7	0.105
8	0.039
9	0.027
10	0.088

Figure 10. Table of experiment 2

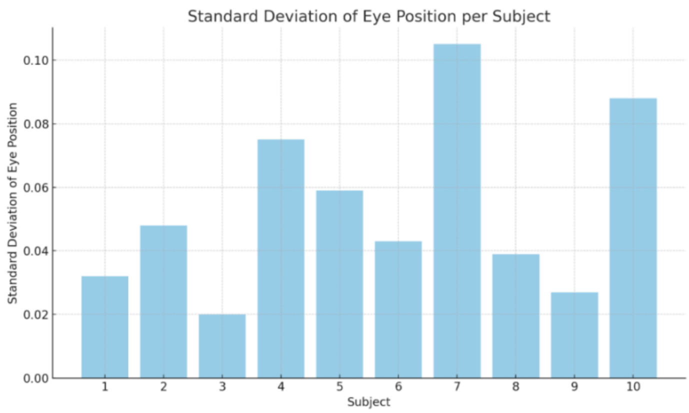


Figure 11. Standard Deviation of Eye Position per Subject

The mean standard deviation found in the analysis was 0.0536, median 0.0455, minimum 0.020 and maximum 0.105. Sample number 7 was an outlying data point having a much higher deviation of 0.105 than the others. This spike probably indicates that the user either repeatedly looked away, having occluded parts of their face (e.g., by hand or spectacles), or head-tilted, which affected MediaPipe's performance in tracking eye landmarks accurately. Large deviation between standard deviation values can be attributedto inadequate lighting or camera position, reflective materials such as glasses disrupting detection, temporary occlusion like scratching or blinking, or even user fidgeting potentially resulting from incipient cognitive loss. Among all the variables, the most significant ones responsible for inconsistent results are face visibility, user movement, and environmental conditions because they all directly affect MediaPipe's ability to deliver stable eye tracking.

5. RELATED WORK

Selkoe et al. discusses the pathology of Alzheimer's disease and describes progress in early detection with imaging and biomarkers such as PET scans and cerebrospinal fluid assays [1]. These steps aim to find Alzheimer's before the symptom onset by assessing amyloid-beta deposits or tau tangles. Although effective for clinical and research applications, these methods are expensive, invasive, and not generally available for early or screening purposes. They also require specialized equipment and trained personnel, preventing their use in the general population. Our project advances Alzheimer's detection by creating an inexpensive, non-invasive desktop application to monitor facial and EEG signals using consumer equipment and open-source libraries. By highlighting subtle cues of behavior and brainwaves and using real-time analysis, we can provide the screening to a wider community.

In a paper by Dauwels et al., the authors propose a machine learning technique to classify Alzheimer's patients based on EEG data [2]. The model employs spectral features and classifiers like Support Vector Machines (SVM) to identify abnormalities in EEG waveforms. While it successfully demonstrates diagnostic potential, the approach requires a large and curated EEG dataset for training, and the resulting output is static. Our methodology avoids the need for a predefined training dataset by integrating OpenAI's real-time prompt engineering and natural language processing to interpret new brainwave data on the fly. This dynamic approach not only avoids training restrictions but also allows for tailored analysis for varying users in various situations and hence is more scalable and flexible.

The Nasreddine et al. 2006 study introduces the Montreal Cognitive Assessment (MoCA), a cognitive screening tool for early diagnosis of Alzheimer's through a brief pen-and-paper test of memory, attention, and visuospatial skills [3]. While MoCA is highly convenient and widely applied in practice, it depends on human operation and subjective evaluation. It is also devoid of integration with electronic systems and does not benefit from biometric inputs. In contrast, our solution fully digitizes the diagnostic process and is centered on passive data acquisition through vision and EEG signals. Our application bridges the gap between behavioral screening and digital biometrics and makes Alzheimer's screening more accessible, objective, and conducive to long-term, repeated monitoring.

6. CONCLUSIONS

A major project constraint is the cost of medical-grade EEG headsets [14]. Consumer-level headsets, while inexpensive, are lower-resolution and fewer channels than the medical-grade equipment. The limitation degrades the accuracy of the analysis of the brainwave data. To address this issue, next phases of the project needs to be funded by grants or partnerships with research institutions in an attempt to acquire more sophisticated EEG equipment. Another major challenge is regarding data privacy and compliance with stringent medical data laws. It is difficult to obtain real EEG data from patients suffering from Alzheimer's disease at the moment because of legal and ethical concerns. Collaborations with medical institutions and practitioners could make accessing anonymized data with the right permission easier. Additionally, the current eye and head tracking system, although functional, can still be optimized. Enhancing the accuracy of the algorithm under different lighting conditions and face angles can give more standard and reliable results.

Overall, this project holds vast potential for mass-scale, real-time Alzheimer’s disease detection with consumer-grade tools. As it adds facial recognition, EEG processing, and advanced AI techniques, this application is an encouraging step towards intervention at an early stage to proactively manage cognitive health and improve overall quality of life [15].

REFERENCES

[1] Nestor, Peter J., Philip Scheltens, and John R. Hodges. "Advances in the early detection of Alzheimer's disease." Nature medicine 10.Suppl 7 (2004): S34-S41.
[2] Laske, Christoph, et al. "Innovative diagnostic tools for early detection of Alzheimer's disease." Alzheimer's & Dementia 11.5 (2015): 561-578.
[3] Mosconi, Lisa, et al. "Early detection of Alzheimer’s disease using neuroimaging." Experimental gerontology 42.1-2 (2007): 129-138.
[4] Blennow, Kaj, Mony J. de Leon, and Henrik Zetterberg. "Alzheimer's disease." The lancet 368.9533 (2006): 387-403.
[5] Lugaresi, Camillo, et al. "Mediapipe: A framework for building perception pipelines." arXiv preprint arXiv:1906.08172 (2019).
[6] Schachter, Armand S., and Kenneth L. Davis. "Alzheimer's disease." Dialogues in clinical neuroscience 2.2 (2000): 91-100.
[7] Soleymani, Mohammad, et al. "Analysis of EEG signals and facial expressions for continuous emotion detection." IEEE Transactions on Affective Computing 7.1 (2015): 17-28.
[8] Brockman, Greg, Ilya Sutskever, and Open AI Team. "Introducing OpenAI." Zugang: https://blog. openai. com/introducing-openai/(15.12. 2019) and the future Digital Single Market. Zugang: https://edps. europa. eu/sites/edp/files/publication/18-04-24_giovanni_buttarelli_keynote_speech_telecoms_forum_ en. pdf (15.12. 2019) (2015).
[9] Cheng, Lu, Kush R. Varshney, and Huan Liu. "Socially responsible ai algorithms: Issues, purposes, and challenges." Journal of Artificial Intelligence Research 71 (2021): 1137-1181.
[10] Tolba, A. S., A. H. El-Baz, and A. A. El-Harby. "Face recognition: A literature review." International Journal of Signal Processing 2.2 (2006): 88-103.
[11] Tomar, Yashika, Sanjana Devi, and Husanpreet Kaur. "Human Motion Tracker using OpenCv and Mediapipe." 2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). IEEE, 2023.
[12] Mukherjee, Ratri, Shemul Sutra Dhar, and Kusum Tara. "Prediction of disorder of brain using EEG signal processing in MATLAB GUI platform." 2017 2nd International Conference on Electrical & Electronic Engineering (ICEEE). IEEE, 2017.
[13] Chowdhary, KR1442, and K. R. Chowdhary. "Natural language processing." Fundamentals of artificial intelligence (2020): 603-649.
[14] Campbell, Andrew, et al. "NeuroPhone: brain-mobile phone interface using a wireless EEG headset." Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds. 2010.
[15] Kaur, Paramjit, et al. "Facial-recognition algorithms: A literature review." Medicine, Science and the Law 60.2 (2020): 131-139.

©2025 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.