Software Engineering Project

Automated Face and Eye Tracker for Autism Diagnosis

Final Report

Group #4

Robert Coven
Alexander Jandi
Veronika Kermoshchuk
Aidan Turner
Erik Yzeiraj

December 10, 2023

Individual Contributions

- Alex
  - Helped create the original eye contact, gaze, and emotion logic
  - Refactored the code into the classes for neatness, readability, and ease of integrating with the other groups
  - Worked on remaking the code again in order to return timestamps as it processes
  - Communicated with Group 2 to create the original proposal and aided with integration
  - For the report, I wrote the explanation of the project creation, a difficulty and its solution, added screenshots of the working program, added a citation
- Veronika Kermoshchuk
  - Changed the eye tracking library to take video instead of webcam input. Worked to create the logic for eye contact, gaze tracking, and emotion change and worked on making it work in real-time. Changed the output to save in the JSON file instead of printing the results. Made JSON file include the description for the emotion change in addition to the timestamp. Fixed bugs related to on tree select function and other bugs. Fixed the stretched video in GUI.
  - For the report, I did the abstract, project objectives, and professional awareness sections.
- Aidan
  - Rewrote the video processing and GUI classes to facilitate multithreading
  - Worked to separate the logic for eye contact, gaze, and emotion tracking into different classes
  - Changed the GUI class to accept JSON input
  - Edited the video processing thread to send JSON output to the GUI then refresh, allowing the sidebar to populate real-time as the video processes
  - Researched optimal combinations of video saturation/brightness to detect the pupil of the eye
  - General bug fixing
  - In terms of the report, I did the timeline and the graphs

- Erik
  - Helped to plan which behaviors were to be tracked.
  - Researched to figure out which models were best suited for building our application off of as well as figuring out the efficacy of these tools.
  - Created prototype GUI using PYSimpleGUI
  - Assisted with editing and debugging of GUI and Video_Processing, and occasionally other modules as errors were discovered.
  - For the report, I edited and added to the abstract, difficulties/solutions, results/discussion, how the software can be improved, and references.
- Robert
  - Worked on GUI design and functionality
  - Frame processing
  - Sidebar
  - Clickable timestamps for sidebar items
  - Live sidebar population, Multithreading
  - Video and frame processing display
  - Frame by frame display
  - Browse and load file functionality
  - For the report I did Difficulties/solutions, Conclusion, and Features list

Abstract

This project uses facial tracking tools to create a program that automates the recognition and time stamping of certain behaviors of potentially autistic children in a given video. Using these tools, we note the movement of the eyes as signifiers of eye contact and attention problems. We also track the movement of the face, specifically facial expressions as signifiers of problems with self-expression, erratic and repetitive behavior. Detected signifiers of relevant behaviors are noted and pinned on the software at the time of detection which reduces the amount of manual recognition that the user must do. Automating manual gaze coding allows the user to spend more of their time on difficult-to-detect behaviors. The user can launch the program and a GUI appears with a button containing an option to “Browse Files”. After a video is uploaded, the user can select a “Process File” button, which will begin the process. The detection tool produces a list of results and gives them to the user in the form of drop-down menus where each label, when clicked, reveals the list of the related timestamps found in the video. The drop-down includes specific behaviors as well as the time when the action took place in the video. The user can then click on one of these timestamps to take them directly to that point of the video in the video player. The behaviors that can be automatically detected with this program include only those that are purely physical and do not require context. Some examples of this include: “Staring”, “Emotion Change”, and “Eye Contact”. The program detects behaviors by recognizing where the eyes are looking and where certain points of the face remain and it does this through an algorithmic understanding which includes the use of precise logic and math.

Objectives

The main objective of this project is to use video processing, facial recognition, and eye-tracking libraries and tools to automate some of the clinician's tasks. These tasks include identifying and labeling both normal and abnormal behaviors in children suspected of having Autism Spectrum Disorder (ASD). This will reduce the time and cost related to this analysis.

To reach the main objective, some sub-objectives need to be considered. The first sub-objective is to research and determine the best-suited open-source facial recognition and eye-tracking libraries that would track eye contact, staring, as well as changes in facial expressions from the video.

As the libraries are determined, another sub-objective is to use an eye-tracking library to focus on the eye movement of the children in the video. Using this library track if the child is making eye contact with the camera and create a label with the timestamp once it has been identified. In addition, the same library should be used to determine if the child in the video is staring and label this behavior accordingly as it has been identified in the video and create a timestamp for each instance of staring.

Using the facial recognition library, detect a change in facial expression and label the action accordingly each time it is detected. The timestamp should be created which shows the time when the action happened in the video and a brief description showing what was the emotion and what it changed to.

Test the program to make sure it detects the target behaviors correctly and records the right timestamps.

Project Description

Detailed Explanation of the Creation Process

The first flag that was implemented was detecting when the child is making eye contact with the camera. The library that was used, *Gaze Tracking*, is designed for real-time use, which makes it fast and optimal for video analysis. The library is able to detect the direction that the child is looking, relative to the direction of the camera. This was utilized in order to record whether or not the child was looking at the camera for every frame of the video and saved into memory. On each frame as well, the previous frames are parsed through and if there are enough frames with eye contact for a long enough time, then the eye contact timestamp is created in the middle of the eye contact duration.

The second flag that was implemented was staring, which is when the child is looking in any direction for over x seconds without blinking. The same library, *Gaze Tracking*, was used for this flag in addition to the previous flag. Using the direction relative to the camera, the eye-direction coordinates can be retrieved and saved into memory for every frame. Then for every frame, the program checks the direction for every previous frame. The software then checks a small "window" around where the child is looking, and records if they are still within the bounds during the previous frames. If this is the case, then the staring timestamp is recorded in the middle of the interval of staring.

The third and final flag that was implemented was detecting when the child's facial changes. This flag utilizes the *DeepFace* library, which is able to analyze an image and return the most likely facial emotion that someone is making, as well as the confidence percent for each of the seven emotions, which are "Angry", "Disgust", "Fear", "Happy", "Sad", "Surprise", and "Neutral". On every frame, the program stores the confidence values into memory. Then once per frame, the software checks the previous frames and calculates the average confidence, which is also stored into memory. The software looks for a pattern in the averages as follows: some first emotion, not confident in any emotion, and lastly a second emotion. This pattern demonstrates that the face has changed emotions. The software then records the time that this has occurred as well as the first and second emotion and appends both of these into the timestamp.

Software Features

- Facial Tracking Tools: The program uses facial tracking technology to analyze videos.
- Recognition and Time Stamping of Behaviors: Automatically recognize and timestamp specific behaviors in potentially autistic children.
- Eye Movement Tracking: Identifies issues with eye contact and attention by tracking eye movements.
- Facial Expression Analysis: Tracks facial expressions to identify problems with erratic behavior.
- Automated Detection: Relevant behaviors are automatically detected and marked in the software at the time they occur, minimizing manual effort.
- Graphical User Interface (GUI): The program includes a user-friendly GUI with a "Browse Files" option for uploading videos.
- Process File Functionality: There's a "Process File" button to start the behavior detection process after a video is uploaded.
- Results Display: Detected behaviors are presented in drop-down menus, showing related timestamps from the video.
- Interactive Timestamps: User can click on time stamped events which will seek to the time in the video where the event occurred.
- Focus on Difficult to Detect Behaviors: Automates the process of gaze coding, allowing users to focus more on behaviors that are harder to detect.

Difficulties and Solutions

Problem: Old timestamps stayed in the sidebar even after loading a new video

Solution: Clearing the added_events set with added_events.clear()

Problem: The accuracy of the facial expressions being recognized was very far off.

Solution: We changed the face recognition model used by DeepFace from OpenCV to MTCNN

Problem: The timestamps only appeared after the entire video had been processed.

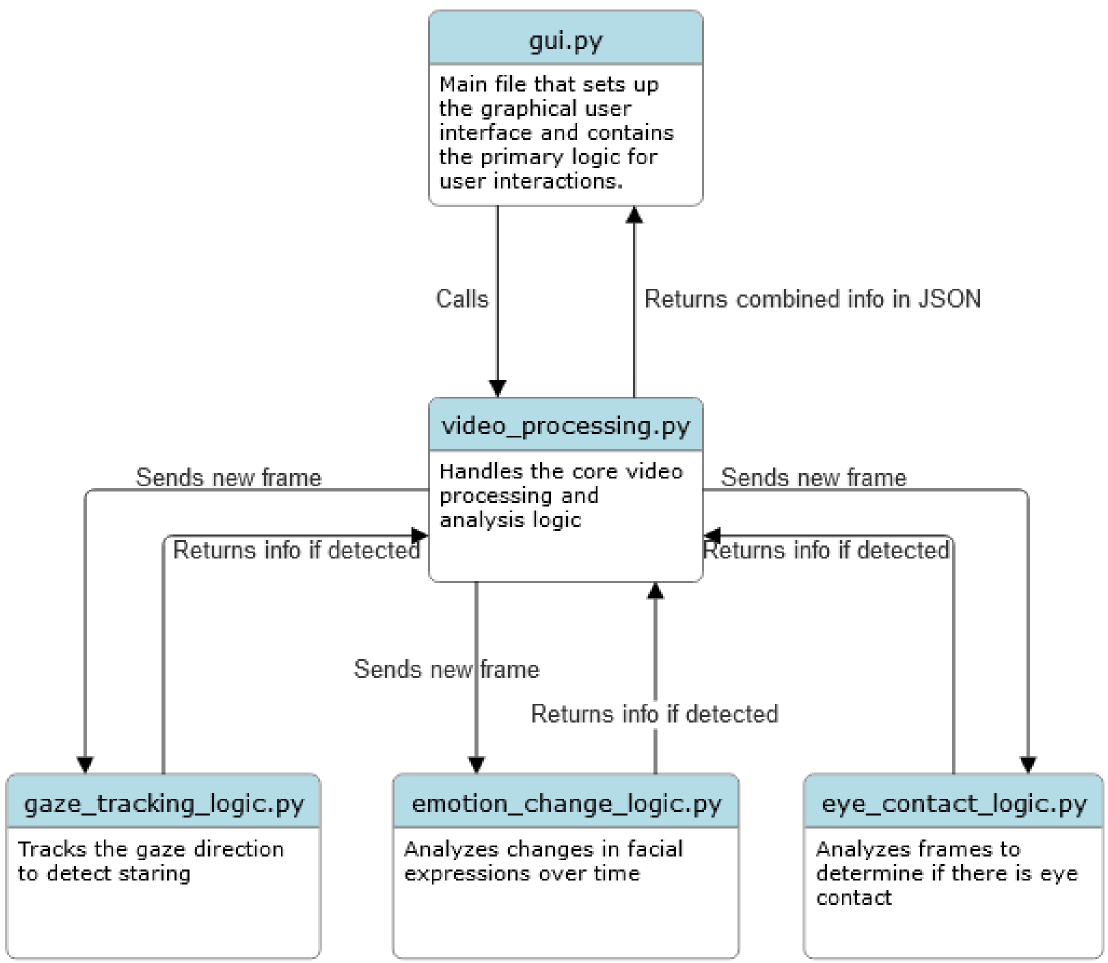
Solution: Refactored the code. The original looping through the frames at the end was adapted to each frame of the video as the program runs.

Problem: The GUI froze while the processing was running.

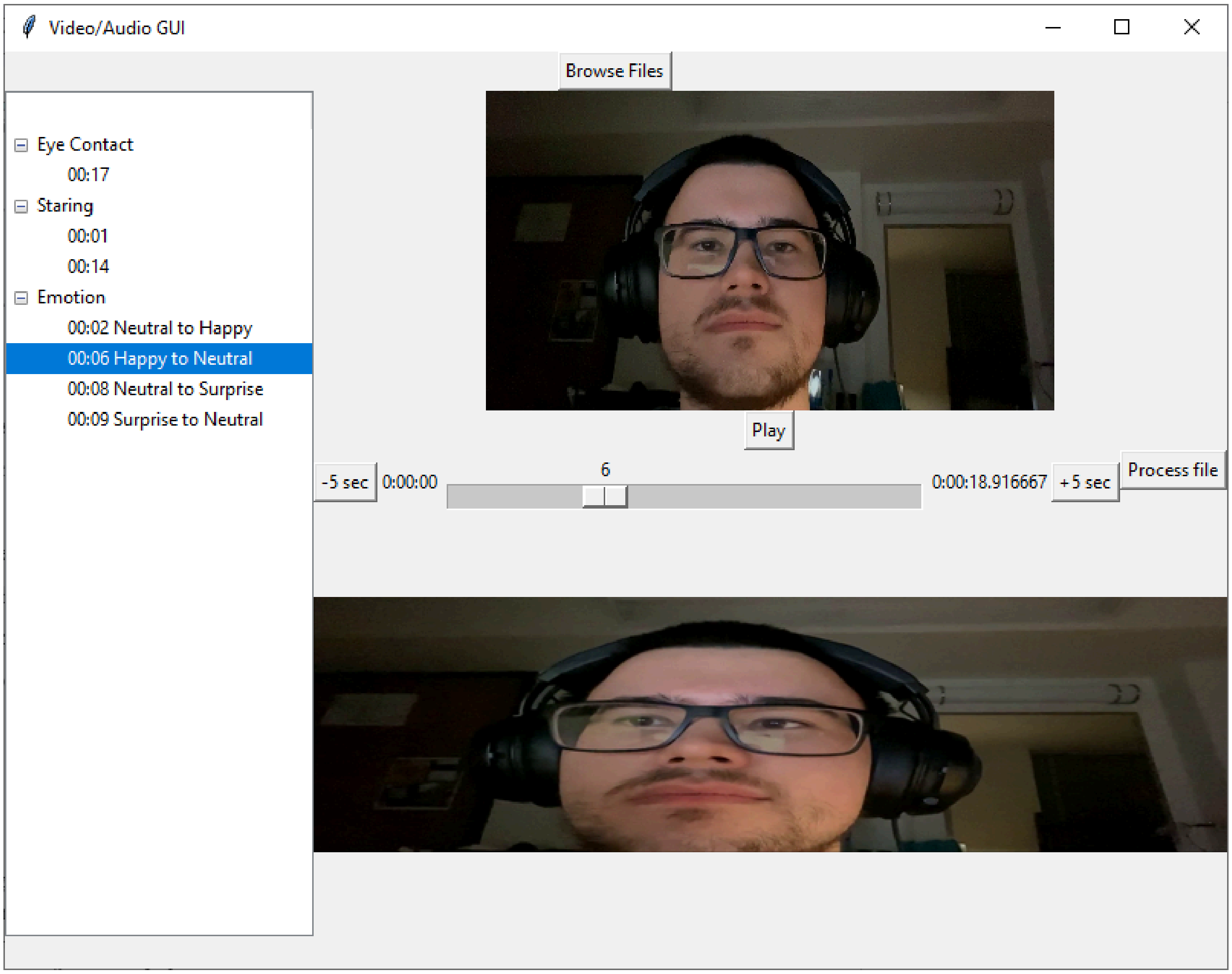
Solution: We changed the processing so it ran on a separate thread from the GUI.

Charts/Graphs on the Software Development Process

Project UML:



Results and Discussion



The program functions as expected and returns the results we were looking for. It accurately provides time stamp events following the logic written. However, if the conditions set in our detection logic are based on an inaccurate assumption, the results would be inaccurate as well. For example, let's say the conditions for our staring detection differ from how it is defined in the medical community. In that case, the results would be unreliable. The benefit of our project is that it can be altered quickly to accommodate new conditions. Another major issue with the results is the length of processing. It's still very slow in its current state. Despite this, it works automatically and can be operated by anyone. According to Courtney E. Venker et al. (2019) other major drawbacks to manual gaze coding is that specialists require extensive reliability training and are subjective. This means no additional time will be required from trained professionals.

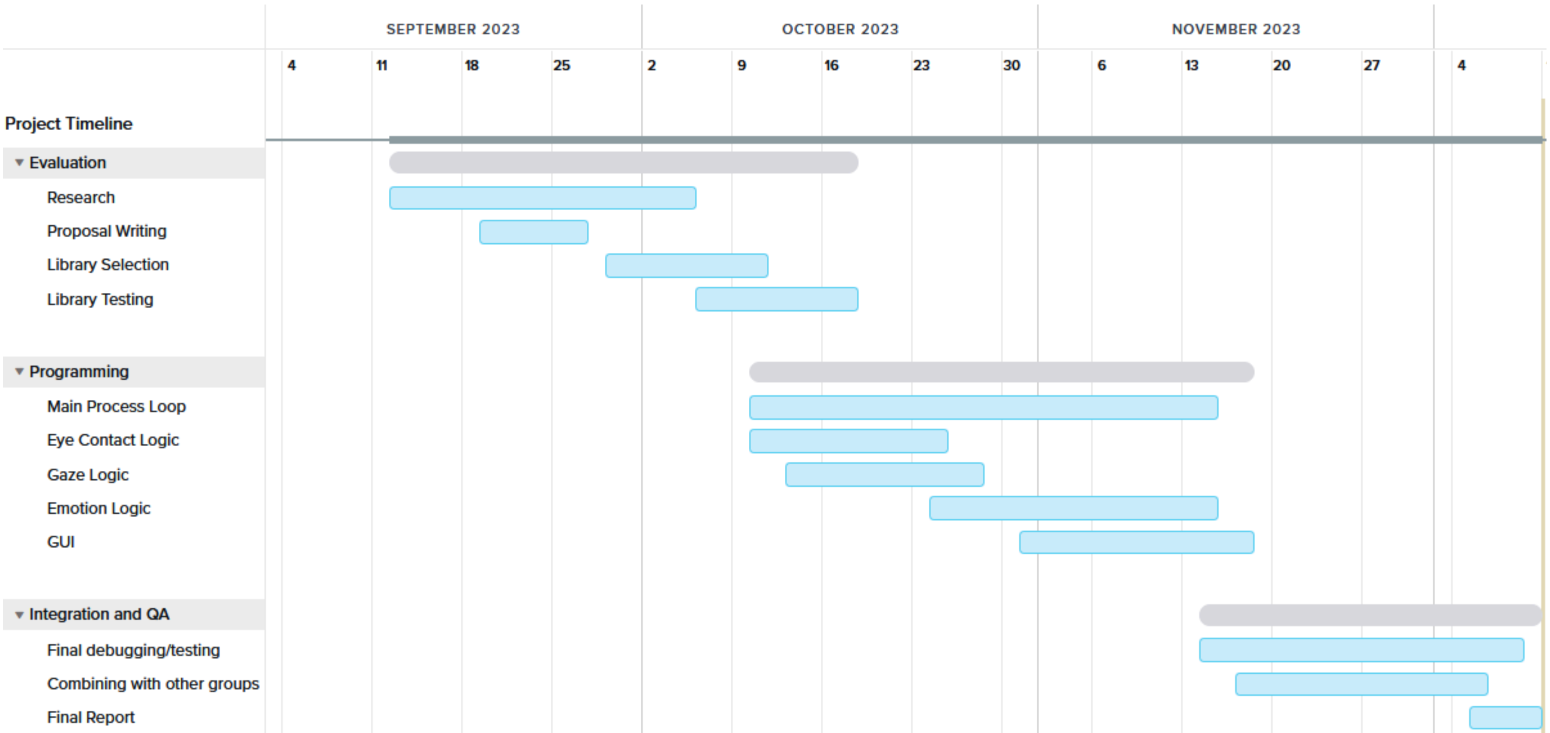
How the Software Could be Improved Upon

- Refine GUI design and functionality
    - There are plenty of features that could be added to cater to new needs of the users based on feedback.
    - The GUI would need to be programmed with HTML if it were to be made accessible through a website.
- Increase processing speed
    - There are other models that could be as accurate as DeepFace, but would require some remodeling. There was a study (Ahmed, et al., 2022) where a few deep learning algorithms were used and the result showed that the best performance was with the MobileNet model. The model was able to accomplish 95% accuracy for the validation data as well as 100% accuracy in the training data with regards to identifying Autism.
- State which direction the child in the video is staring
- Increase the accuracy for every behavior
    - Some behaviors should be disabled if the child is not capable of the behavior in order to act as a safeguard and reduce misclassifications (LeBlanc, 2020)
- New Calculation for facial expression frequency
    - According to a psychiatric research team from University of Bordeaux (2021), "A recent meta-analytical review reported that autistic individuals display spontaneous facial expressions less frequently and for a shorter duration than in non-autistic individuals. Furthermore, they are less accurate and lower in quality (19). Only 7 studies were found to focus specifically on spontaneous FEEs in ASD using new technology"
- Classifying location specific engagement
    - You could potentially use the eye tracking feature to calculate how focused a patient is on a particular spot in a video frame. Researchers from *Translational Psychiatry* mentioned in an article, "Computer vision has also been used to predict engagement and learning performance. For example, Ahmed and Goodwin[53] analysed facial expressions from video recordings obtained when kids interacted with a computer-assisted instruction programme. Their results showed that emotional and behavioural engagement can be quantified automatically using computer vision analysis." (De Belen, 2020)

Professional Awareness

Developing a program designed to assist physicians in diagnosing ASD through automated video analysis will significantly improve our skills as programmers. This project will require a mastery of machine learning algorithms for pattern recognition, expertise in video processing, and a deep understanding of the medical criteria for ASD. We will also gain valuable experience in integrating healthcare guidelines and ethical considerations into software design, enhancing our ability to create solutions that are not only technologically advanced but also socially responsible and clinically relevant. Finally, we will improve our ability to work in team environments and further develop vital skills such as communication that are required to further our professional careers.

Project Timeline



Conclusion and Future Work

The main objective of this assignment was to automate the diagnostic process of ASD. In doing so, we expect quicker turnaround time for the diagnostic process leading to fewer cases going undetected. To achieve this we used a combination of eye tracking, facial analysis, and video processing. The program we designed displays time stamped events such as eye contact, staring spells and emotion change. Events are displayed in the sidebar of the graphic user interface. The timestamps are clickable so that a clinician can instantly seek to the event in the video for review. The development process had many challenges, starting with repurposing a webcam based gaze tracking library to be able to handle recorded video instead of live camera footage. Developing the logic which handled checks for eye contact, staring, and emotion state change went through many iterations.

Ultimately, our hope is that our project will serve as a valuable tool for clinicians, assisting in the expedited and accurate diagnosis of ASD. By reducing the time and effort required for manual behavior analysis, clinicians can focus more on the support of their patients and hopefully catch more undiagnosed cases. Future iterations of this software could include more advanced features such as a wider range of video format support, more elegant GUI design, and faster processing time to further reduce diagnostic turnaround.

References

Ahmed, Ibrahim Abdulrab, et al. “Eye tracking-based diagnosis and early detection of autism spectrum disorder using machine learning and Deep Learning Techniques.” *Electronics*, vol. 11, no. 4, 2022, p. 530, https://doi.org/10.3390/electronics11040530.

Briot, Kellen, et al. “New Technologies as promising tools for assessing facial emotion expressions impairments in ASD: A systematic review.” *Frontiers in Psychiatry*, vol. 12, 5 May 2021, https://doi.org/10.3389/fpsyt.2021.634756.

De Belen, Ryan Anthony, et al. “Computer vision in autism spectrum disorder research: A systematic review of Published Studies from 2009 to 2019.” Translational Psychiatry, vol. 10, no. 1, 30 May 2020, https://doi.org/10.1038/s41398-020-01015-w.

Leblanc, Emilie, et al. "Feature replacement methods enable reliable home video analysis for machine learning detection of autism." Scientific reports 10.1 (2020): 21245.

Venker, Courtney E., et al. “Comparing automatic eye tracking and manual gaze coding methods in young children with autism spectrum disorder.” *Autism Research*, vol. 13, no. 2, 2019, pp. 271–283, https://doi.org/10.1002/aur.2225.