Beach ***E***nvironment for the ***A***nalyti***c***s of ***H***uman ***Gaze*** (BEACH-Gaze) User Manual

Release Version: 1.0

Release Date: 10.13.2024

***D**ata* Semantics & Human ***D**ata* Interaction Lab (D[2] Lab)

California State University Long Beach

Privacy and ethics statement: as an open-source software distributed under the GNU General Public License, BEACH-Gaze promotes transparency in its development and use. The source code is publicly available, allowing the broader scientific community to review, contribute to, and improve the software. BEACH-Gaze does not store any data uploaded by the user. The software processes gaze data solely for the purpose of providing descriptive and predictive gaze analytics. All data processed by the software remains on the user's local machine and is not transmitted or stored on any external servers. Users have complete control over their data, ensuring privacy and security throughout their use of the software. BEACH-Gaze supports ethical research and applications of eye tracking. We emphasize the importance of obtaining informed consent from all participants whose data is collected and analyzed using our software. Researchers and developers using BEACH-Gaze are expected to follow ethical guidelines and standards set by their respective institutions and governing bodies. The software should not be used for any purpose that could harm individuals or violate their privacy.

Table of Contents

1 Introduction

Beach **E**nvironment for the **A**nalyti**c**s of **H**uman **Gaze** (BEACH-Gaze) provides reuseable and extensible support to the broader scientific community interested in descriptive and predictive gaze analytics applied to a wide range of domains. BEACH-Gaze is compatible with Gazepoint GP3 and GP3 HD eye trackers and uses the Waikato Environment for Knowledge Analysis (WEKA) [1] to generate gaze-based classifications. BEACH-Gaze is open-source software issued under the GNU General Public License.

2 Setup

BEACH-Gaze has been tested on machines running Windows 10, Windows 11, and Mac OS 13.

You will need to set up a compatible Gazepoint eye tracker with a monitor up to 24". Follow the instructions provided by Gazepoint to set up your study environment such as calibration. This release supports raw gaze data produced by the GP3 and GP3 HD eye trackers eye trackers.

You will need to have Apache Maven [2] support and the Java 21 JDK installed on your machine. Maven will automatically install and update the required dependencies.

NB: note that the screen size is hard-coded to be 1920x1080 in the `Constants.java` file. If the resolution of the monitor that the experiment was conducted on differs from this size, it is necessary to update the `Constants.java` file to reflect the correct screen size. Failure to do so may result in incorrect calculations.

3 How to Use BEACH-Gaze

First clone the repository from GitHub here [3] using git or your preferred version control system. You can then execute the codebase using your instance of Apache Maven. Two of our preferred methods are through VSCode with the Maven for Java extension (see [4]), or through the command line (see [5]).

Upon launching BEACH-Gaze, you will be greeted with 3 tabs on the top; **Descriptive Analytics**, **Predictive Analytics**, and **Help**. **Descriptive Analytics** is the default tab shown (see Fig. 1).

3.1 Descriptive Analytics

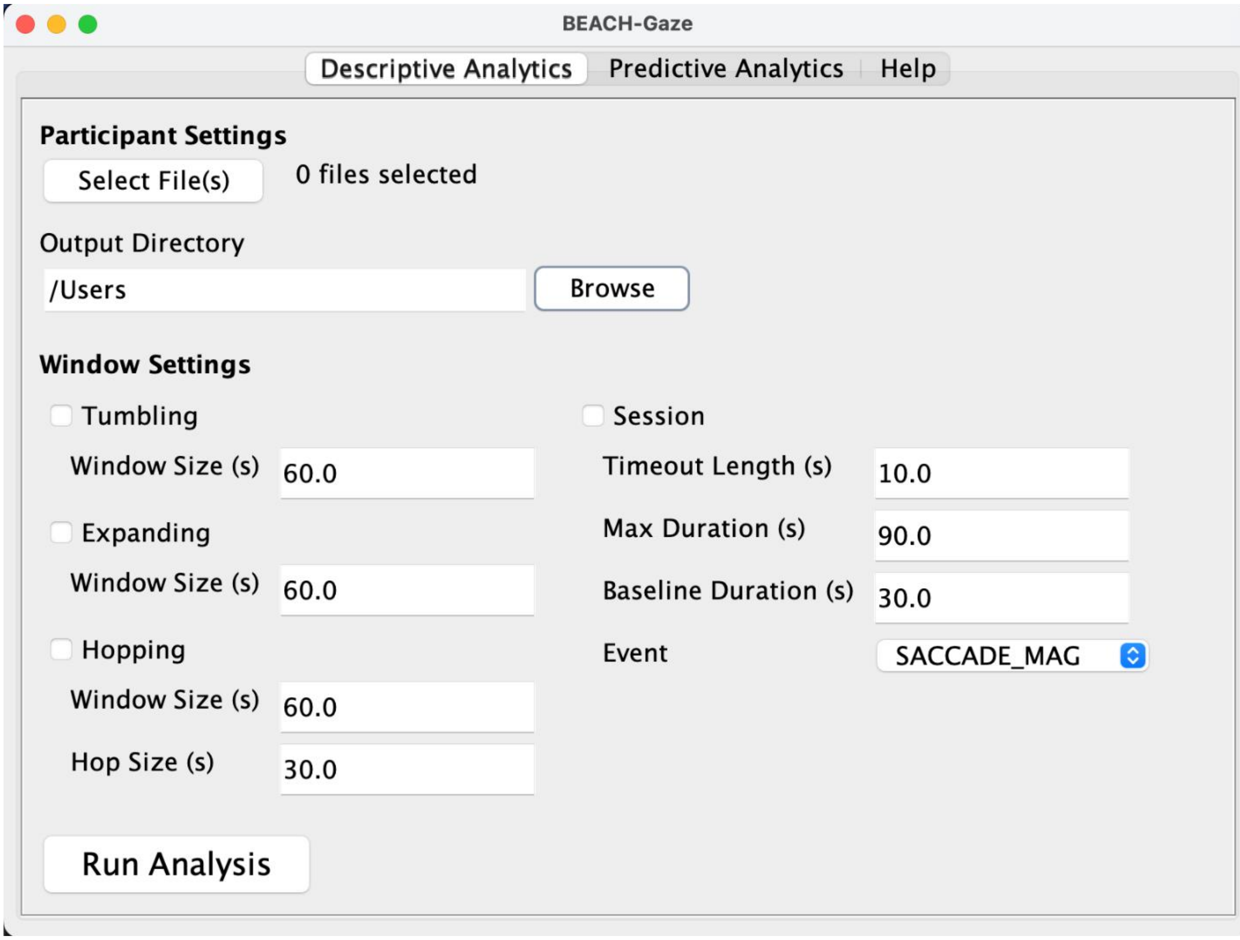

Fig. 1: Descriptive Analytics

The Descriptive Analytics tab is used to process raw gaze data from Gazepoint Analysis and produce a list of descriptive gaze measures (DGMs) for each participant. Multiple participant files can be selected to do a batch analysis of participants. To learn more about the participants files and the DGMs produced, see section 4.1.

Users can select a desired output directory for the resulting DGMs. By default, this directory will be in the same location where BEACH-Gaze is located.

Window settings allow users to perform window-based analyses of the DGMs [6], wherein the participant's gaze file can be analyzed over time by:

- taking a **scheduled digest view** of the gaze data using a **tumbling window** that is non-overlapping and fixed in size.
- taking the **most recent snapshot view** of the gaze data using a **hopping window** that is overlapping and fixed in size.

- taking an **event-based view** of the gaze data using a **session window** that is non-overlapping and non-fixed in size.
- taking a **cumulative view** of the gaze data using an **expanding window** that is overlapping and non-fixed in size.

To enable a window, simply select the checkbox appearing next to the window's name and enter the desired parameters corresponding to each window. All window parameters are defined in seconds. Once users have selected one or more files and filled out their desired fields, they can press the "Run Analysis" button to begin generating DGMs.

3.1.1 Tumbling Window

Gaze data is analyzed in a series of non-overlapping, fixed-size windows at scheduled contiguous time intervals, shown in Fig. 2. The analysis process makes initial predictions based on user gaze found in a given time zone (i.e., window 1), then moves on to the next bordering time zone to make subsequent predictions (i.e., window 2), and tumbles forward until reaching the end of an interaction. The window size, or time zone, can be defined at any interval as appropriate depending on the given scenario. For example, Fig. 1 shows a default value of 60s, whereby window 1 (containing gaze data between 0-60 seconds into an interaction), 2 (containing gaze data between 60-120 seconds into an interaction), 3 (containing gaze data between 120-180 seconds into an interaction), and so on, are 60 seconds long.

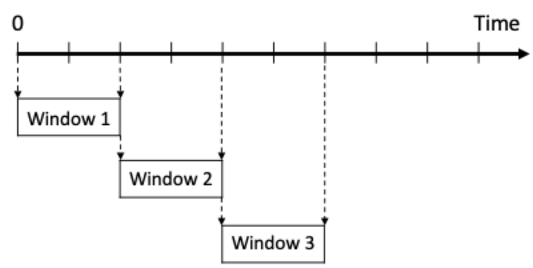

Fig. 2. Taking a scheduled digest view of the gaze data using a tumbling window that is non-overlapping and fixed in size.

3.1.2 Hopping Window

Taking a snapshot view of one's gaze, it is possible to capture the most recent state of user's visual attention by utilizing an overlapping hopping window, shown in Fig. 3, whereby scheduled overlapping windows are utilized at a given interval. A hopping window analyzes gaze in a defined window size, then moves forward to the next scheduled hop relative to the previous one. For example, Fig. 1 shows a default value of a 60s window size and a 30s hop size, meaning every 30 seconds, gaze over the last 60 seconds is analyzed, i.e., window 1 contains gaze data between 0-60 seconds, window 2 contains gaze data between 30-90 seconds, window 3 contains data between 60-120 seconds, and so on.

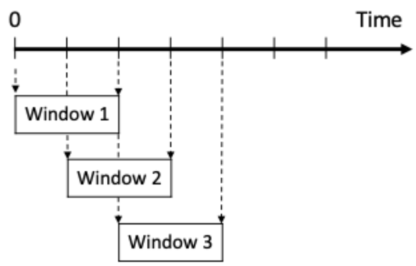

Fig. 3. Taking the most recent snapshot view of the gaze data using a hopping window that is overlapping and fixed in size.

3.1.3 Session (Event-Based) Window

During an interaction, users may experience defining moments that are impactful enough to affect their performance to a notable degree. These defining moments may translate to notable gaze behaviors resembling phases of significant events. Within this context, gaze events can be defined as anything unusual relative to what is already known of a user's gaze. In other words, a user's initial gaze values observed at the beginning of an interaction are considered as a baseline, whereby future values can be compared against and deemed unusual if found longer than the baseline. BEACH-Gaze currently supports the following seven definitions of a notable gaze event, including:

- SACCADE_MAG, also known as saccadic length, i.e., the magnitude of the saccade calculated as distance between each fixation (in pixels);
- SACCADE_DIR, also known as absolute angles, which is the angle between each fixation in degrees from horizontal (in degrees);
- LPMM, the diameter of the left eye pupil (in millimeters);
- RPMM, the diameter of the right eye pupil (in millimeters);
- LPMM + RPMM, which is the average value combining LPMM and RPMM (in millimeters);
- POGD, the duration of a fixation (in seconds); and
- BKPMIN, the number of blinks in the previous 60 second period of time (count).

A non-overlapping, non-fixed-size session window is used to achieve event-based gaze analytics, shown in Fig. 4. A session window begins when the first event is found; it then keeps searching for the next event within a specified time period. If nothing is found, it would close at a specified timeout (e.g., window 1); if another event is found, the session window would extend the search within another timeout period and repeat this process (e.g., window 3) until a specified maximum duration (e.g., window 2). For example, Fig. 1 shows a default value of 10s timeout, 90s maximum duration, 20s baseline duration, and SACCADE_MAG, meaning that using the saccadic length to define a notable gaze event, a baseline (i.e., the average saccadic length observed) is established for a user after 20 seconds, whereby subsequent saccadic lengths that are higher in value are deemed as "events" for that person. If another event is found, the session window grows (i.e., the 10-second timeout is renewed) until it meets the maximum duration set to 90 seconds. If no further events are found, the session window closes.

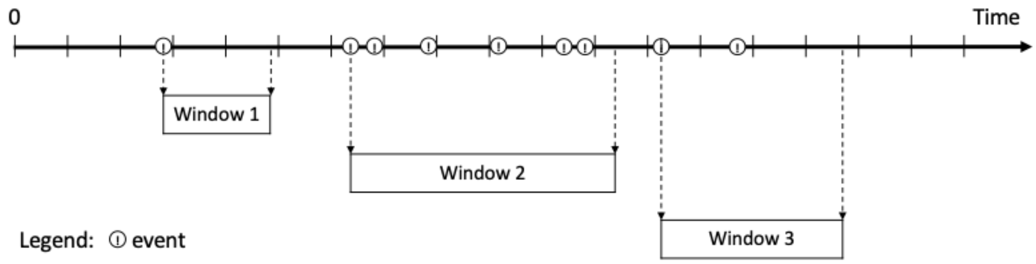

Fig. 4. Taking an event-based view of the gaze data using a session window that is non-overlapping and non-fixed in size.

3.1.4 Expanding Window

An expanding window takes a cumulative view to account for all that is known for a person's gaze, shown in Fig. 5. An initial set of gaze data collected from a user is analyzed (e.g., window 1), this can then be expanded to include new gaze data generated for that person at the next specified time interval (e.g., window 2 and 3). For example, Fig. 1 shows a default value of 60s, meaning that every 60 seconds, the window grows (to 120 seconds, then 180 seconds, and so on) to include new gaze data to what is already known about the person.

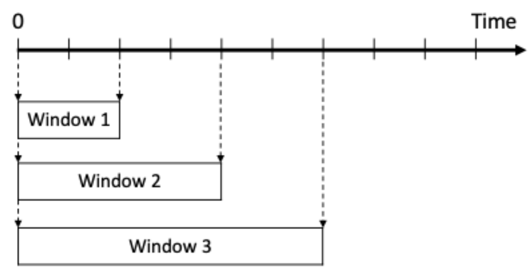

Fig. 5. Taking a cumulative view of the gaze data using an expanding window that is overlapping and non-fixed in size.

3.2 Predictive Analytics

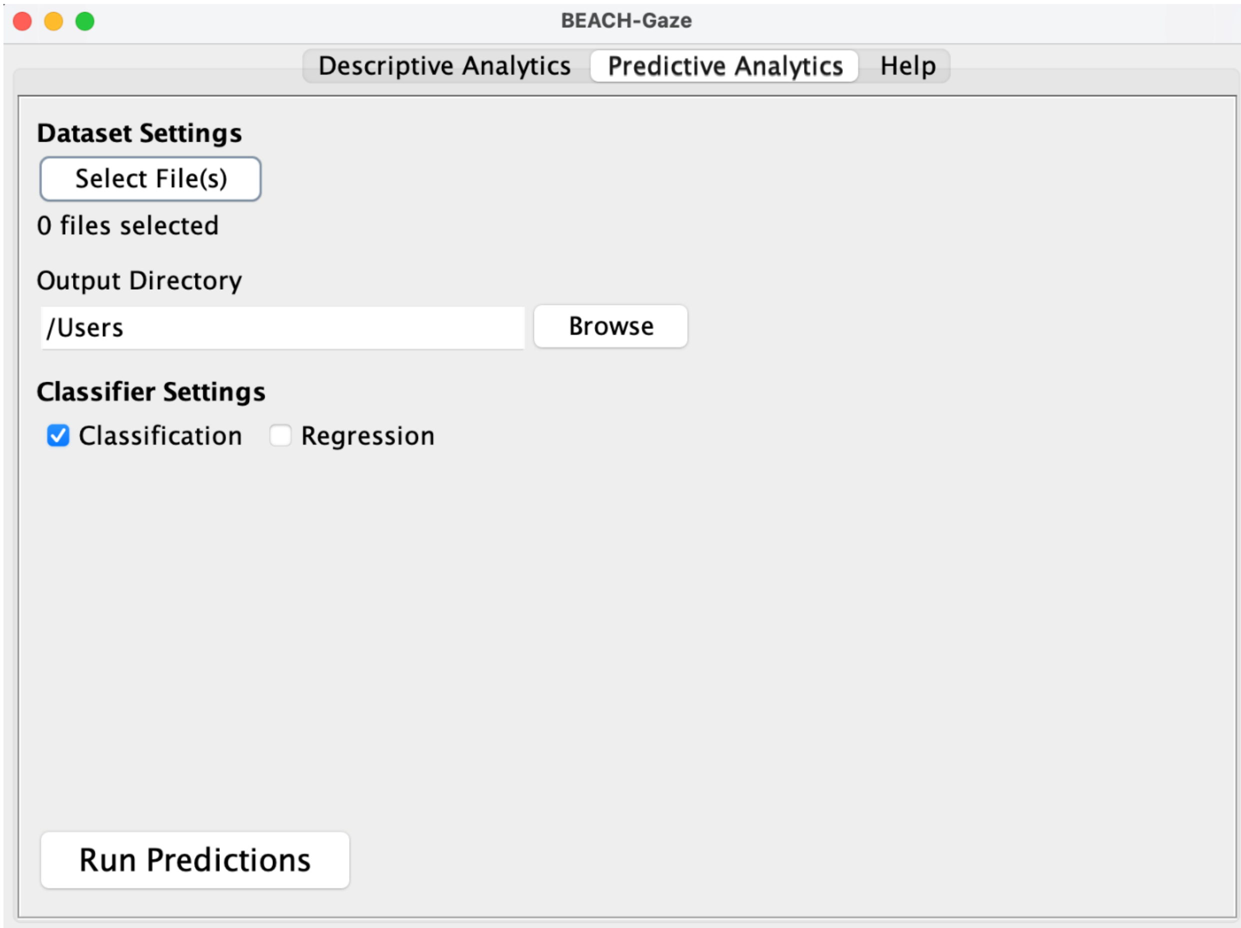

Fig. 6: Predictive Analytics

The Predictive Analytics tab is used to generate predictions on nominal and binary data using the DGMs produced by Descriptive Analytics. It makes use of the Waikato Environment for Knowledge Analysis (WEKA) API version 3.8.6. [7], which is a software collection of established machine learning algorithms. Users can select one or multiple datasets to be run against a collection of classifiers (either classification or regression classifiers). See Table 1 for the list of classifiers. To learn more about the dataset files and prediction files produced, see section 4.2.

Table 1: List of classifiers

Classification	Regression
ZeroR (baseline)	ZeroR (baseline)
BayesNet	GaussianProcesses
NaiveBayes	LinearRegression
NaiveBayesMultinomialText	MultilayerPerceptron
NaiveBayesUpdateable	SimpleLinearRegression

NaiveBayesMultinomial NaiveBayesMultinomialUpdateable Logistic MultilayerPerceptron SGD SGDText SimpleLogistic SMO VotedPerceptron IBk Kstar LWL AdaBoostM1 AttributeSelectedClassifier Bagging ClassificationViaRegression CVParameterSelection FilteredClassifier IterativeClassifierOptimizer LogitBoost MultiClassClassifier MultiClassClassifierUpdateable MultiScheme RandomCommitee RandomizableFilteredClassifier RandomSubSpace Stacking Vote WeightedInstancesHandlerWrapper InputMappedClassifier DecisionTable Jrip OneR PART DecisionStump HoeffdingTree J48 LMT RandomForest RandomTree REPTree	SMOreg Bagging CVParameterSelection RegressionByDiscretization MultiScheme RandomCommittee RandomizableFilteredClassifier RandomSubSpace Stacking Vote WeightedInstancesHandlerWrapper DecisionTable M5Rules M5P REPTree IBk Kstar LWL DecisionStump RandomForest AdditiveRegression AttributeSelectedClassifier ElasticNet IsotonicRegression LeastMedSq IterativeAbsoluteErrorRegression

Classifiers are run using WEKA's default parameters. In WEKA's documentation, the default settings for classifiers typically specify default parameters such as the number of iterations, the base model used, and other algorithm-specific options that optimize performance. By default, the test option for evaluating a model is set to perform 10-fold cross-validation, which divides the dataset into 10 parts, using 9 parts for training and 1 part for testing, cycling through all combinations to ensure a robust evaluation.

Users can also select a desired output directory where the resulting classifications and directories produced will be created. The default directory will be in the same location where BEACH-Gaze is located.

Once all required fields are populated, users can press the "Run Predictions" button to begin generating predictive analytics.

3.3 Help

The Help tab provides details related to the four types of window-based analytics, discussed in section 3.1.

4 Inputs and Outputs

The inputs required for BEACH-Gaze depend on whether the user aims to perform descriptive analytics (discussed in section 4.1) or predictive analytics (discussed in section 4.2).

4.1 Descriptive Analytics

4.1.1 File Inputs

Descriptive Analytics takes in the raw gaze-data files exported from Gazepoint Analysis as input. When you export from Gazepoint Analysis after a participant recording, you will receive two .CSV files named `(`*`participant`*`)_all_gaze.csv` and `(`*`participant`*`)_fixations.csv`. Note that `(`*`participant`*`)` is a placeholder for the participant's name defined in Gazepoint Analysis and that `_all_gaze.csv` and `_fixations.csv` are default naming conventions from Gazepoint. Descriptive Analytics will only take the `(`*`participant`*`)_all_gaze.csv` file (from now on referred to as `all_gaze.csv`) as an input, so each participant will correspond to a single .csv file.

4.1.2 File Outputs and Data Produced

Below is a list of files and directories that are produced after a successful analysis.

4.1.2.1 AOIs (directory)

The AOIs directory contains subset .csv files of the participant's `all_gaze.csv` file, partitioned by AOI. For example, if an AOI A is defined in the `all_gaze.csv` file, every row found to be within AOI A will be pushed into a subset `all_gaze.csv` file named `A_all_gaze.csv`.

4.1.2.2 baseline (directory)

The baseline directory contains a subset of the participant's `all_gaze.csv` file from the beginning to the length of the event baseline duration. By default, this value is 30 seconds. It also contains the descriptive gaze measures calculated during this period, see section 4.1.2.10 for more details on descriptive gaze measures.

4.1.2.3 event/expanding/hopping/tumbling (directory)

The window directories are optionally produced directories that must be enabled (see section 3.1 on how to enable them). Within each directory are subdirectories of each defined window and the DGMs for each, as well as an AOIs directory, an `AOI_DGMs.csv`, and `AOI_Transitions.csv`. There is also an `all_window_DGMs.csv` which is a compilation of each window's DGMs into one .csv file. See the corresponding sections below for more information on `AOI_DGMs.csv`, `AOI_Transitions.csv`, and `DGMs.csv`.

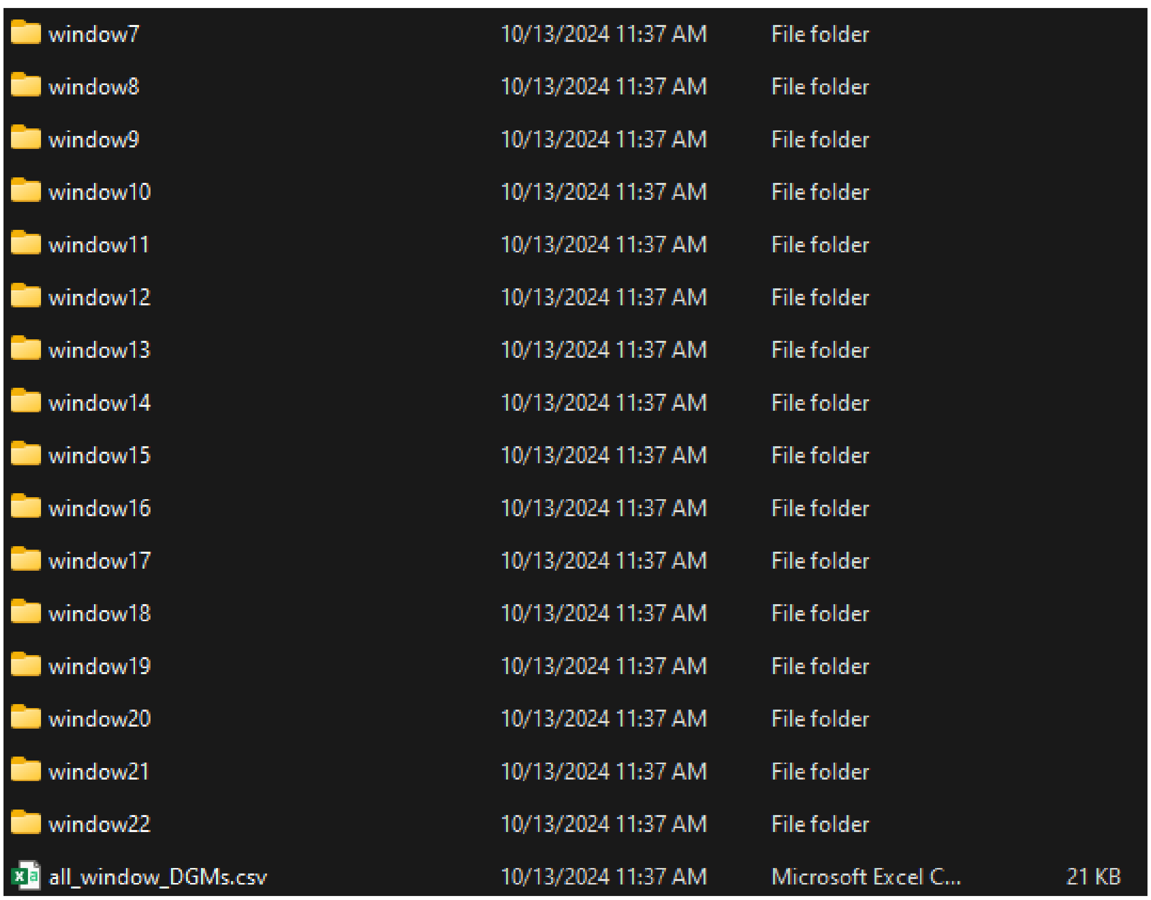

Fig 7. Inside of an event/expanding/hopping/tumbling directory

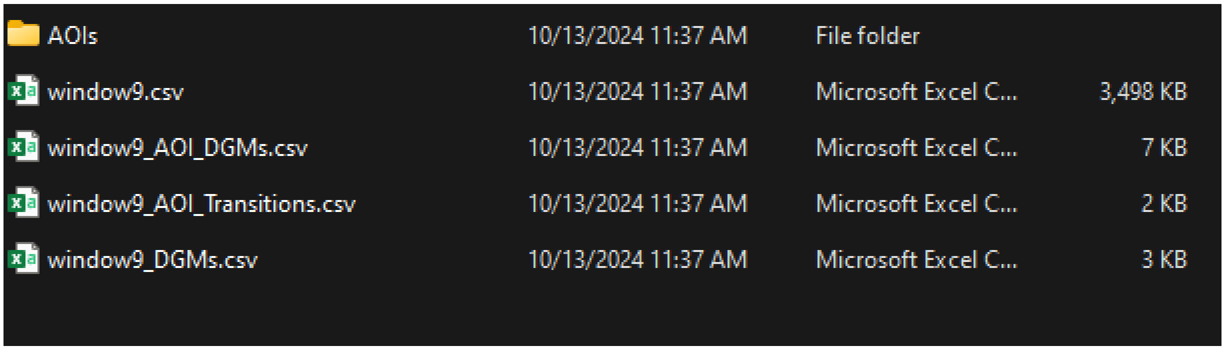

Fig. 8: Inside of a window# directory

4.1.2.4 aoiDescriptions.txt

The aoiDescriptions.txt is a text file used to map AOIs to a code letter. This is used for pattern and sequence analysis.

4.1.2.5 collapsedSequence.txt

The collapsedSequence.txt is a text file that contains the sequence of characters representing the order in which fixations have occurred and visited AOIs. AOIs are mapped to a code letter which is defined in aoiDescriptions.txt. The sequence is collapsed, meaning that any adjacent pairs of AOIs are removed. For example, the sequence “ACCDEABAAAABC” would become “ACDEABABC”.

4.1.2.6 expandedSequence.txt

The expandedSequence.txt is a text file that contains the sequence of characters representing the order in which fixations have occurred and visited AOIs. AOIs are mapped to a code letter which is defined in aoiDescriptions.txt. The sequence is expanded, meaning that adjacent pairs of AOIs are allowed. For example, the sequence “ACCDEABAAAABC” would be a valid sequence.

4.1.2.7 (*participant*)_AOI_DGMs.csv

The AOI_DGMs.csv file is a DGMs file that contains descriptive gaze measures produced per AOI rather than the whole screen (see 4.1.2.10 for information about DGMs file). Each row represents the descriptive gaze measures for a unique AOI. In addition to the standard DGMs, additional AOI-specific measures are produced. See the list below for details.

- aoi: The name of the AOI
- proportion_of_fixations_spent_in_aoi (%): The percentage of fixations that were spent in that defined AOI.
- proportion_of_fixations_durations_spent_in_aoi (%): The percentage of fixation duration that was spent in the defined AOI.

4.1.2.8 (*participant*)_AOI_Transitions.csv

The AOI_Transitions.csv file contains transitional features for each AOI pair, describing the transitions from one AOI to the next. See the list below for details on each measure produced:

- aoi_pair: The name of an AOI pair describing a transition that starts in the first AOI and ends in the second. Note that order matters.
- transition_count: The number of transitions that occurred for the given AOI pair.
- transition_proportion_including_self_transitions: The proportion of total transitions that occurred between AOI pairs, inclusive of self-transitions. A self-transition refers to a situation where the gaze moves away from an AOI and then returns to the same AOI without visiting any other AOIs in between
- transition_proportion_excluding_self_transitions: The proportion of total transitions that occurred between AOI pairs, exclusive of self-transitions.

4.1.2.9 (*participant*)_collapsedPatterns.csv

The collapsedPatterns.csv file contains descriptive measures produced from analyzing the patterns exhibited in a participant's collapsed sequence [8, 9]. See the list below for details.

- pattern_string: A string of characters representing a sequence of AOIs.
- frequency: The number of times a pattern appears in a participant's sequence.
- sequence_support: The number of sequences in a group where the pattern occurs as a proportion of the total number of sequences in the group
- average_pattern_frequency: Total number of occurrences of the pattern in all sequences of a group divided by the group's total number of sequences.
- proportional_pattern_frequency: The total number of occurrences of the pattern per group as a proportion of the total number of patterns in the group.

4.1.2.10 (*participant*)_DGMs.csv

The DGMs file contains a list of descriptive gaze measures that characterize a participant through their gaze behavior. See the list below for details on each DGM.

- total_number_of_fixations: The number of fixation points found in an interaction.
- sum_of_all_fixation_duration (s): The sum duration of the fixations.
- mean_fixation_duration (s): The mean duration of the fixations.
- median_fixation_duration (s): The median duration of the fixations.
- stdev_of_fixation_durations (s): The standard deviation of the duration of fixations.
- min_fixation_duration (s): The minimum duration of the fixations.
- max_fixation_duration (s): The maximum duration of the fixations.
- total_number_of_saccades: The number of saccades found in an interaction.
- sum_of_all_saccade_lengths (px): The sum of the distance between fixations.
- mean_saccade_length (px): The mean distance between fixations.
- median_saccade_length (px): The median distance between fixations.
- stdev_of_saccade_lengths (px): The standard deviation of the distances between fixations.
- min_saccade_length (px): The minimum distance between fixations.
- max_saccade_length (px): The maximum distance between fixations.
- sum_of_all_saccade_durations (px): The sum of all the saccade durations.

- mean_saccade_duration (s): The mean duration of the saccades.
- median_saccade_duration (s): The median duration of the saccades.
- stdev_of_saccade_durations (s): The standard deviation between saccade durations.
- min_saccade_duration (s): The minimum duration of the saccades.
- max_saccade_duration (s): The maximum duration of the saccades.
- sum_of_all_saccade_amplitudes(°): The sum of all the saccade durations.
- mean_saccade_amplitude (°): The mean duration of the saccades.
- median_saccade_amplitude (°): The median duration of the saccades.
- stdev_of_saccade_amplitude (°): The standard deviation between saccade durations.
- min_saccade_amplitude (°): The minimum duration of the saccades.
- max_saccade_amplitude (°): The maximum duration of the saccades.
- scanpath_duration (s): The duration of all fixations and saccades.
- fixation_to_saccade_ratio: The sum of fixation duration divided by the sum of saccadic duration.
- sum_peak_saccade_velocity (°/s): The sum of all peak saccade velocities. See section 4.2.3 for more information
- mean_peak_saccade_velocity (°/s): The average value of the peak saccade velocity. See section 4.2.3 for more information.
- median_peak_saccade_velocity (°/s): The median peak saccade velocitiy. See section 4.2.3 for more information
- std_peak_saccade_velocity (°/s): The standard deviation between saccade velocities. See section 4.2.3 for more information
- min_peak_saccade_velocity (°/s): The minimum peak saccade velocity. See section 4.2.3 for more information
- max_peak_saccade_velocity (°/s): The maximum peak saccade velocity. See section 4.2.3 for more information
- sum_mean_saccade_velocity (°/s): The sum of all mean saccade velocities. See section 4.2.3 for more information
- mean_mean_saccade_velocity (°/s): The average value of the mean saccade velocities See section 4.2.3 for more information.
- median_mean_saccade_velocity (°/s): The median mean saccade velocity. See section 4.2.3 for more information
- std_mean_saccade_velocity (°/s): The standard deviation between mean saccade velocities. See section 4.2.3 for more information
- min_mean_saccade_velocity (°/s): The minimum mean saccade velocity. See section 4.2.3 for more information
- max_mean_saccade_velocity (°/s): The maximum mean saccade velocity. See section 4.2.3 for more information
- sum_of_all_absolute_degrees (°): The sum of all absolute saccadic angles.
- mean_absolute_degree (°): The mean absolute saccadic angles.
- median_absolute_degree (°): The median absolute saccadic angle.
- stdev_of_absolute_degrees (°): The standard deviation between absolute saccadic angles.
- min_absolute_degree (°): The minimum absolute saccadic angle.
- max_absolute_degree (°): The maximum absolute saccadic angle.
- sum_of_all_relative_degrees (°): The sum of all relative saccadic angles.

- mean_relative_degree (°): The mean relative saccadic angle.
- median_relative_degree (°): The median relative saccadic angle.
- stdev_of_relative_degrees (°): The standard deviation between relative saccadic angles.
- min_relative_degree (°): The minimum relative saccadic angle.
- max_relative_degree (°): The maximum relative saccadic angle.
- convex_hull_area (px^2): The area of the smallest convex set that encloses all fixation points
- stationary_entropy: The concentration/dispersion in a participant's gaze. See section 4.2.3 for more information.
- transition_entropy: The extent to which a participant has switched between the different AOIs of a stimulus. See section 4.2.3 for more information.
- average_blink_rate_per_minute (blink/m): The rate at which a participant blinks every minute
- average_pupil_size_of_left_eye (mm): The average size of the left pupil during the interaction.
- average_pupil_size_of_right_eye (mm): The average size of the right pupil during the interaction.
- average_pupil_size_of_both_eyes (mm): The average size of both pupils during the interaction.
- total_number_of_l_mouse_clicks: The number of left mouse clicks found in an interaction.
- total_number_of_valid_recordings: The number of valid data rows.

4.1.2.11 (*participant*)_expandedPatterns.csv

The expandedPatterns.csv file contains descriptive measures produced from analyzing the patterns exhibited in a participant's expanded sequence [7, 8]. See the list below for details.

- pattern_string: A string of characters representing a sequence of AOIs.
- frequency: The number of times a pattern appears in a participant's sequence.
- sequence_support: The number of sequences in a group where the pattern occurs as a proportion of the total number of sequences in the group
- average_pattern_frequency: Total number of occurrences of the pattern in all sequences of a group divided by the group's total number of sequences.
- proportional_pattern_frequency: The total number of occurrences of the pattern per group as a proportion of the total number of patterns in the group.

4.1.2.12 (*participant*)_fixations.csv

The fixations.csv file is a subset of the (*participant*)_all_gaze.csv file that was taken in as input from the user. Every row in this file represents a unique fixation. This is equivalent to the (*participant*)_fixations.csv file produced by Gazepoint Analysis.

4.1.2.13 (*participant*)_valid_all_gaze.csv

The valid_all_gaze.csv file is a subset of the (*participant*) all_gaze.csv that was taken in as an input from the user. To mitigate effects on data granularity caused by noisy data, we run the participant data through a cleaning process before performing calculations. Every row in this file is considered a valid row post data cleaning. For more information on the cleaning process, see the code documentation.

4.1.2.14 (*participant*)_valid_fixations.csv

The valid_all_gaze.csv file is a subset of the (*participant*)_fixations.csv file. Similar to the valid_all_gaze.csv file mentioned in section 4.1.2.13, every row in this file is considered a valid row post data cleaning.

4.1.2.15 collapsedPatterns.csv

If multiple participant files are analyzed, a collapsedPatterns.csv file will be produced. This is similar to the participant-specific file described in section 4.1.2.9, but is instead analyzed over the group of participants.

4.1.2.16 combinedDGMs.csv

If multiple participant files are analyzed, a combinedDGMs.csv file will be produced. This file will compile the DGMs from each participant and output it as a singular file, where each row represents the DGMs of a participant.

4.1.2.17 expandedPatterns.csv

If multiple participant files are analyzed, an expandedPatterns.csv file will be produced. This is similar to the participant-specific file described in section 4.1.2.11, but is instead analyzed over the group of participants.

4.2 Predictive Analytics

4.2.1 File Inputs

Predictive Analytics takes in datasets in the .csv file format as an input. The last column in the dataset must be the class index, or the target attribute that is being predicted. Every other cell in the dataset that does not belong in the header row or the last column must be nominal/numerical. The cells in the last row column must all be either nominal or binary, depending on whether classification or regression is being utilized.

4.2.2 File Outputs and Data Produced

4.2.2.1 (*dataset)*_irl.csv

This file is the instances result listener, where (*dataset*) is the name of the input dataset file. This file is an intermediary file used by WEKA to produce its calculations. We leave it to provide more data to users.

4.2.2.2 predictions.csv

predictions.csv is a results matrix where the first column is the name of the dataset, and the first row is the name of the classifier. The corresponding cells are the classifier's prediction score for that dataset. For classification, we compare percent correct and for regression we compare root relative squared error.

4.2.3 Additional Details on Descriptive Gaze Measures

Some descriptive gaze measures require more explanation in their calculations than is provided in section 4.2.2. We provide additional details here for clarity.

4.2.3.1 Saccade Velocity

Saccade velocity is the speed achieved during a saccade [10, 11, 12, 13]. It can be calculated using the following equations:

$$v(t) = \frac{\vec{X}(t) - \vec{X}(t-\mathrm{h})}{\mathrm{h}}$$

$$\vec{X}(t) - \vec{X}(t-\mathrm{h}) \;=\; tan^{-1}\left(\frac{\sqrt{(x_t - x_{t-\mathrm{h}})^2 + (y_t - y_{t-\mathrm{h}})^2}}{k}\right)$$

where t is the time, h is the difference in time between two points, $v(t)$ is peak saccade velocity at time t, $\vec{X}(t) - \vec{X}(t-h)$ is saccadic amplitude between two points, x_t and y_t are the 2D coordinates of the participant's gaze at time t, and k is the participant's distance from the monitor. Note that saccade amplitude is the angular displacement in the participant's eye measured between two fixation points.

Peak saccade velocity is the fastest speed achieved during a saccade. Mean saccade velocity is the average speed during a saccade, calculated by summing up all velocities during a saccade and dividing by the total number of velocity measurements (which is one less than the total number of points, since n points produce n-1 velocities between consecutive points), excluding any measurements that exceed the velocity threshold. If no valid velocities remain, NaN is returned.

4.2.3.2 Stationary Entropy

Stationary entropy is a unitless measure that describes gaze dispersion and provides information about the distribution of gaze across AOIs [13]. It can be calculated using the following equation:

$$H_s = -\sum_{i=1}^{n} p_i log_{10}(p_i)$$

where n is the number of AOIs and p_i is the probability of viewing the i^{th} AOI. p_i can be obtained by dividing the number of fixations in an AOI by the total number of fixations for all AOIs.

The value for stationary entropy ranges from 0 - log(n), where log(n) indicates that gaze was distributed equally among all AOIs, i.e. all AOIs have an equal amount of fixations and viewing probability, and 0 indicates that gaze was concentrated entirely on a single AOI, i.e. all fixations are within a single AOI and it has a viewing probability of 1 whereas others have a probability of 0.

4.2.3.3 Transition Entropy

Transition entropy is a unitless measure that describes visual scanning patterns and indicates the extent of which a participant has switched between AOIs. [14] It can be calculated using the following equation:

$$H_t = -\sum_{i=1}^{n} p_i \sum_{j=1}^{n} p_{ij}\, log_{10}(p_{ij})$$

where n is the number of AOIs, p_i is the probability of viewing the i^{th} AOI, and p_{ij} is the probability of viewing the j^{th} AOI given that you are currently viewing the i^{th} AOI. p_i can be obtained by dividing the number of fixations in an AOI by the total number of fixations for all AOIs, and p_{ij} can be calculated by counting the amount of transitions from the i^{th} AOI to the j^{th} AOI and dividing by the total amount of AOI transitions.

Transition entropy ranges from 0 – log(n), where log(n) indicates a random scanning pattern wherein the probability of viewing a random AOI given another is all equal, and 0 indicates little to no scanning pattern, where transitions only occur in one AOI, i.e. $i = j$.

5 Limitations

The descriptive analytics produced by BEACH-Gaze relies on the existence of several data columns to perform its calculations. For an exhaustive list, see the code documentation. To ensure that the BEACH-Gaze functions correctly, select all the possible columns when exporting data from the Gazepoint Analysis. To do this, navigate to the “Analyze Data” tab and then click the “Export” button. Click on the “Select CSV Data” button and then make sure all columns are checked. See the figures below for a visual guide.

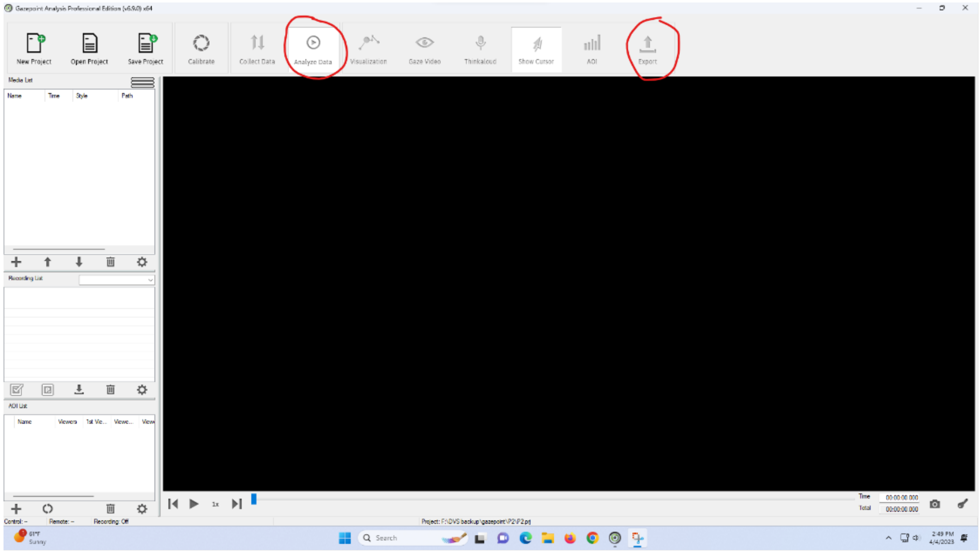

Fig. 9: Gazepoint Analysis where the “Analyze Data” tab and “Export” button are highlighted

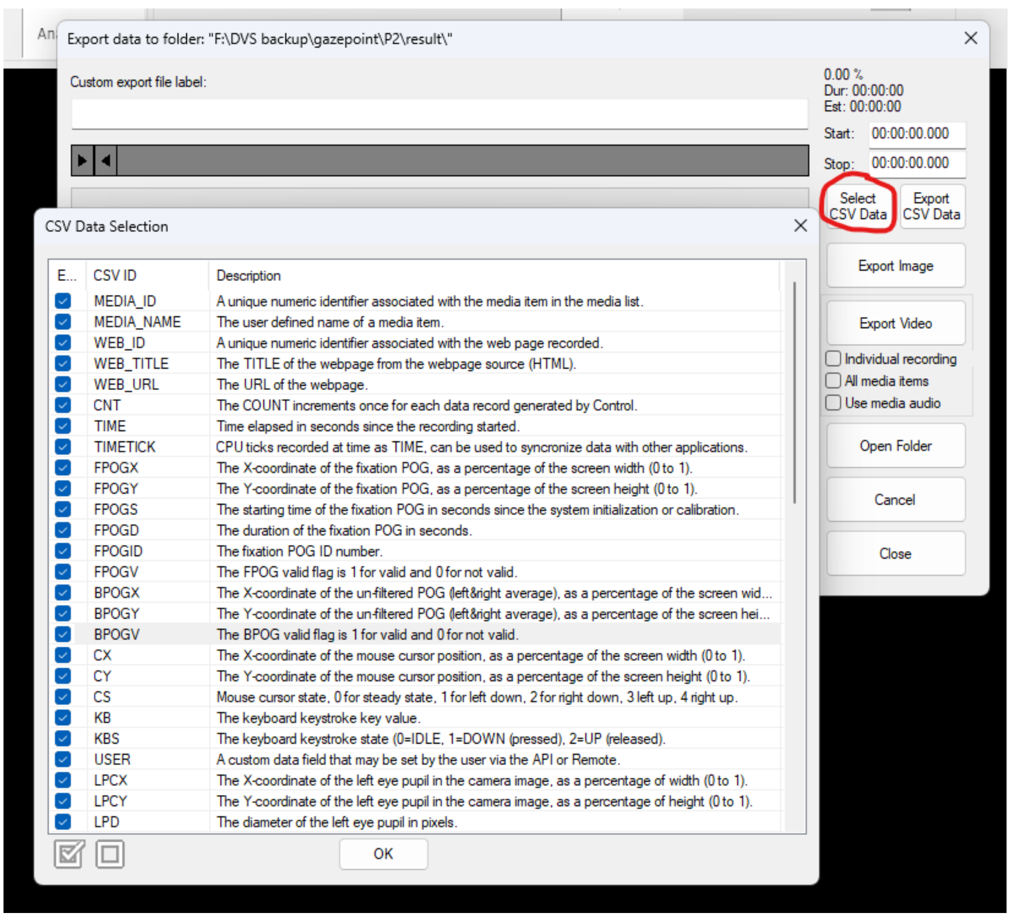

Fig. 10: The window popup produced when the “Export” button is pressed with the “Select CSV Data Button” highlighted

Note that the program will create a folder for the participant regardless of the success of the program execution. In the event of program failure, folders may be empty or incomplete.

6 References

[1] Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016. ISBN:978-0-12-804291-5

[2] B. Porter, J. van Zyl, and O. Lamy, “Welcome to Apache Maven,” Maven, https://maven.apache.org/ (accessed Oct. 13, 2024).

[3] TheD2Lab, “BEACH-Gaze,” GitHub, https://github.com/TheD2Lab/BEACH-Gaze (accessed Oct. 13, 2024).

[4] E. Redmond, “Maven in 5 minutes,” Maven, https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html (accessed Oct. 13, 2024).

[5] Microsoft, “Maven and Gradle support for java in visual studio code,” RSS, https://code.visualstudio.com/docs/java/java-build (accessed Oct. 13, 2024).

[6] Bo Fu, Predictive Gaze Analytics: A Comparative Case Study of the Foretelling Signs of User Performance During Interaction with Visualizations of Ontology Class Hierarchies, Multimodal Technologies and Interaction, 8(10):90, 2024 https://doi.org/10.3390/mti8100090

[7] University of Waikato, “WEKA,” Weka Wiki, https://waikato.github.io/weka-wiki/downloading_weka/ (accessed Oct. 13, 2024).

[8] B. Steichen, M. M. A. Wu, D. Toker, C. Conati, and G. Carenini, “Te,Te,Hi,Hi: Eye Gaze Sequence Analysis for Informing User-Adaptive Information Visualizations,” in *User Modeling, Adaptation, and Personalization*, Cham: Springer International Publishing, 2014, pp. 183–194. Accessed: Oct. 22, 2023. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-08786-3_16

[9] J. M. West, A. R. Haake, E. P. Rozanski, and K. S. Karn, “eyePatterns,” in *Proceedings of the 2006 symposium on Eye tracking research & applications - ETRA ’06*, 2006. Accessed: Oct. 22, 2023. [Online]. Available: http://dx.doi.org/10.1145/1117309.1117360

[10] A. Gibaldi and S. P. Sabatini, “The saccade main sequence revised: A fast and repeatable tool for oculomotor analysis,” *Behavior Research Methods*, vol. 53, no. 1, pp. 167–187, Jul. 2020. doi:10.3758/s13428-020-01388-2

[11] A. T. Bahill and J. D. McDonald, “Frequency limitations and optimal step size for the two-point central difference derivative algorithm with applications to human eye movement data,” *IEEE Transactions on Biomedical Engineering*, vol. BME-30, no. 3, pp. 191–194, Mar. 1983. doi:10.1109/tbme.1983.325108

[12] H. C. Cuve, J. Stojanov, M. X. Roberts-Gaal, C. Catmur, and G. Bird, *Validation of gazepoint low-cost eye-tracking and psychophysiology bundle*, Mar. 2021. doi:10.31234/osf.io/7nz9y

[13] A. T. Bahill, M. R. Clark, and L. Stark, "Saccadic velocity characteristics: Intrinsic variability and fatigue," *Biological Cybernetics*, vol. 19, no. 3, pp. 201-211, 1975.

[14] M. Doellken, J. Zapata, N. Thomas, S. Matthiesen, "Implementing innovative gaze analytic methods in design for manufacturing: a study on eye movements in exploiting design guidelines" in *Science Direct*, Elsevier Ltd., 2021. [Online].
Available: https://doi.org/10.1016/j.procir.2021.05.097