Klassifizierung von Webcam-basierten Eyetracking-Daten

ERGEBNISSE MEINER BA-ARBEIT

STARTING POINT

Zemblys Paper

We conclude that machine-learning techniques lead to superior detection compared to current state-of-the-art event detection algorithms [...].

Behav Res
DOI 10.3758/s13428-017-0860-3

CrossMark

Using machine learning to detect events in eye-tracking data

Raimondas Zemblys[1,2] · Diederick C. Niehorster[3,4] · Oleg Komogortsev[5] ·
Kenneth Holmqvist[2,6]

Abstract Event detection is a challenging stage in eye movement data analysis. A major drawback of current event detection methods is that parameters have to be adjusted based on eye movement data quality. Here we show that a fully automated classification of raw gaze samples as belonging to fixations, saccades, or other oculomotor events can be achieved using a machine-learning approach. Any already manually or algorithmically detected events can be used to train a classifier to produce similar classification of other data without the need for a user to set parameters. In this study, we explore the application of random forest machine-learning technique for the detection of fixations, saccades, and post-saccadic oscillations (PSOs). In an effort to show practical utility of the proposed method to the applications that employ eye movement classification algorithms, we provide an example where the method is employed in an eye movement-driven biometric application. We conclude that machine-learning techniques lead to superior detection compared to current state-of-the-art event detection algorithms and can reach the performance of manual coding.

Keywords Eye movements · Event detection · Machine learning · Fixations · Saccades

✉ Raimondas Zemblys
r.zemblys@tf.su.lt

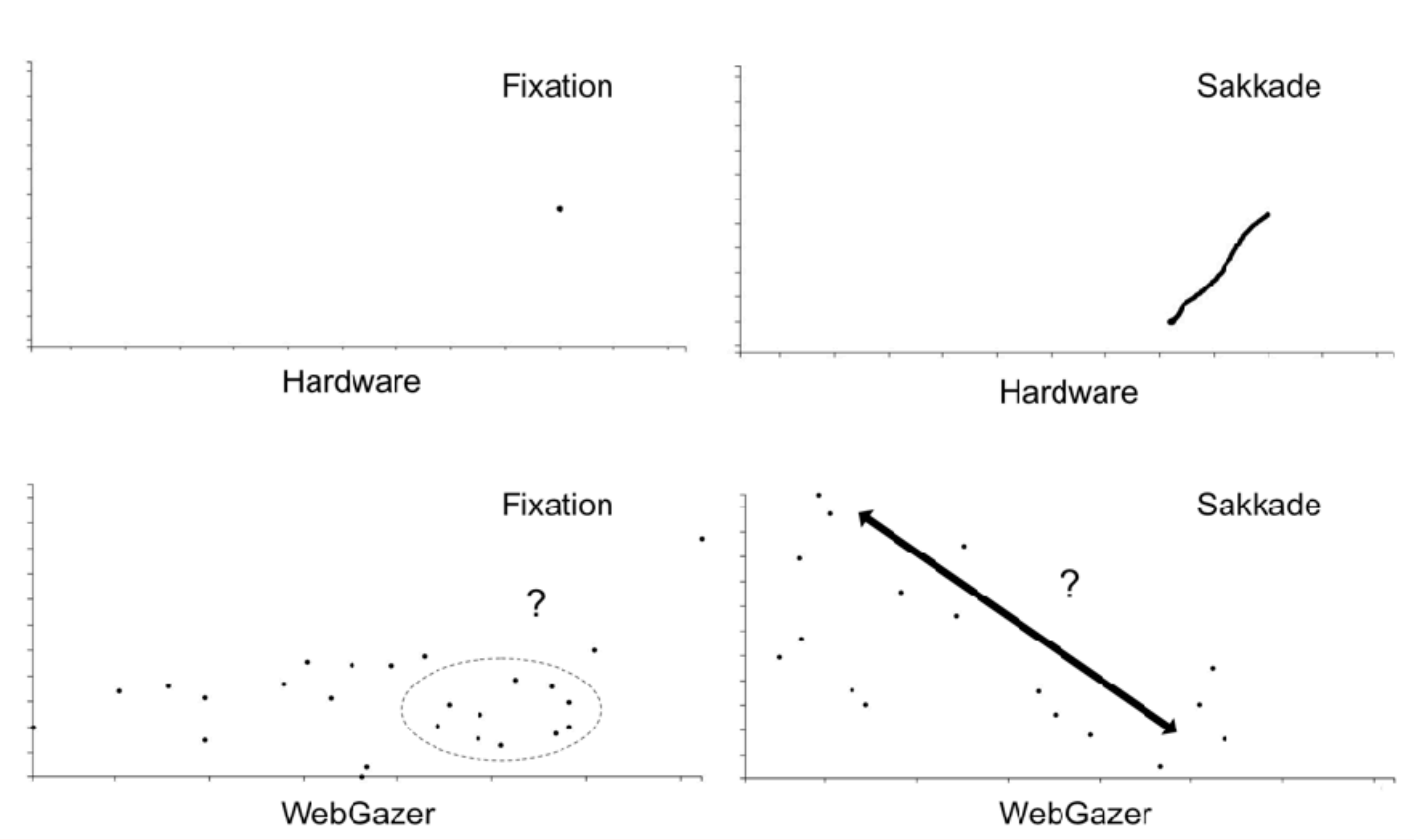
[1] Department of Engineering, Siauliai University, Siauliai, Lithuania

Introduction

In eye movement research, the goal of *event detection* is to robustly extract events, such as fixations and saccades, from the stream of raw data samples from an eye tracker, based on a set of basic rules and criteria which are appropriate for the recorded signal. Until recently, researchers who ventured to record eye movements were required to conduct time-consuming manual event detection. For instance, Hartridge and Thomson (1948) devised a method to analyze eye movements at a rate of 10000 s (almost 3 h) of analysis time for 1 s of recorded data, and as Monty (1975) remarked: "It is not uncommon to spend days processing data that took only minutes to collect" (p. 331–332).

Computers have fundamentally changed how eye movement data are analyzed. Today, event detection is almost exclusively done by applying a detection algorithm to the raw gaze data. For a long time, two broad classes

FIRST DATA


Fixation
Hardware
Sakkade
Hardware
Fixation
?
WebGazer
Sakkade
?
WebGazer

ADVANTAGES OF MACHINE-LEARNING

Zemblys Paper

Most of these algorithms work well within the assumptions they make of the data. Examples of common assumptions are that the input must be **high-quality data**, or data recorded at **high sampling frequencies** [...]. When the sampling frequency is too low, or too high, or the precision of the data is poor, or there is data loss, **many of these algorithms fail** (Holmqvist et al. 2012, 2016).

Behav Res
DOI 10.3758/s13428-017-0860-3

CrossMark

Using machine learning to detect events in eye-tracking data

Raimondas Zemblys[1,2] · Diederick C. Niehorster[3,4] · Oleg Komogortsev[5] · Kenneth Holmqvist[2,6]

Abstract Event detection is a challenging stage in eye movement data analysis. A major drawback of current event detection methods is that parameters have to be adjusted based on eye movement data quality. Here we show that a fully automated classification of raw gaze samples as belonging to fixations, saccades, or other oculomotor events can be achieved using a machine-learning approach. Any already manually or algorithmically detected events can be used to train a classifier to produce similar classification of other data without the need for a user to set parameters. In this study, we explore the application of random forest machine-learning technique for the detection of fixations, saccades, and post-saccadic oscillations (PSOs). In an effort to show practical utility of the proposed method to the applications that employ eye movement classification algorithms, we provide an example where the method is employed in an eye movement-driven biometric application. We conclude that machine-learning techniques lead to superior detection compared to current state-of-the-art event detection algorithms and can reach the performance of manual coding.

✉ Raimondas Zemblys
r.zemblys@tf.su.lt

[1] Department of Engineering, Siauliai University, Siauliai, Lithuania

Keywords Eye movements · Event detection · Machine learning · Fixations · Saccades

Introduction

In eye movement research, the goal of *event detection* is to robustly extract events, such as fixations and saccades, from the stream of raw data samples from an eye tracker, based on a set of basic rules and criteria which are appropriate for the recorded signal. Until recently, researchers who ventured to record eye movements were required to conduct time-consuming manual event detection. For instance, Hartridge and Thomson (1948) devised a method to analyze eye movements at a rate of 10000 s (almost 3 h) of analysis time for 1 s of recorded data, and as Monty (1975) remarked: "It is not uncommon to spend days processing data that took only minutes to collect" (p. 331–332).

Computers have fundamentally changed how eye movement data are analyzed. Today, event detection is almost exclusively done by applying a detection algorithm to the raw gaze data. For a long time, two broad classes

RANDOM FOREST

WebGazer data

	timestamp	x	y
401	127833	420.594802	760.075459
402	127890	423.654510	756.349346
403	127946	395.596778	701.915289

RANDOM FOREST

WebGazer data | Features

	timestamp	x	y	hz	distance	velocity	acceleration
401	127833	420.594802	760.075459	19.230769	42.527993	0.817846	0.015728
402	127890	423.654510	756.349346	17.543860	4.821382	0.084586	0.001484
403	127946	395.596778	701.915289	17.857143	61.239717	1.093566	0.019528

14 Features

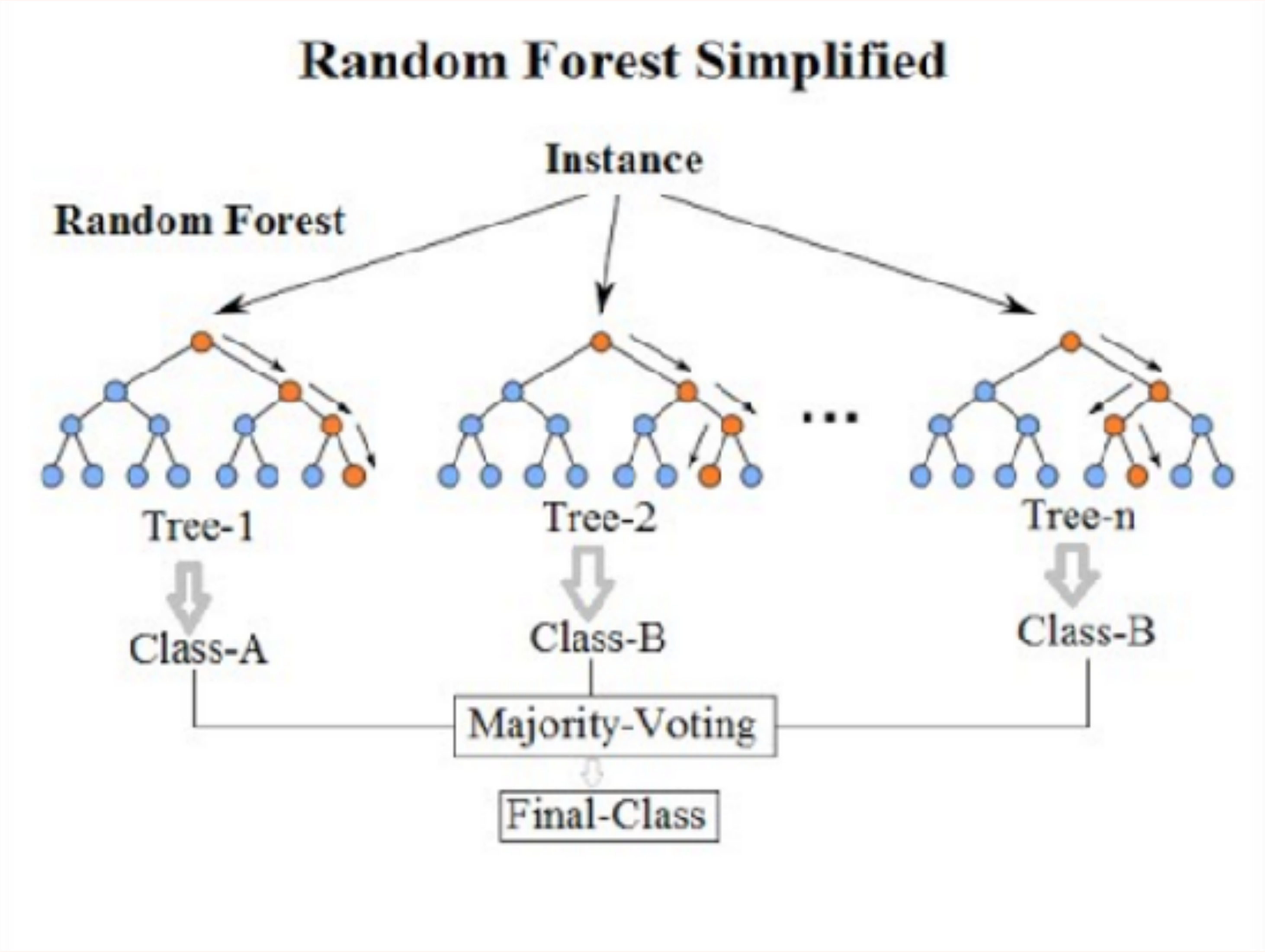
Feature	Description
fs	*sampling frequency* (Hz). As some features may provide different information at different sampling rates (e.g., SMI BeGaze uses velocity for data sampled at 200 Hz and more and dispersion at lower frequencies), providing the classifier with information about sampling frequency may allow it to make better decision trees
rms	*root mean square* (°) of the sample-to-sample displacement in a 100-ms window centered on a sample. The most used measure to describe eye-tracker noise (Holmqvist et al., 2011)
std	*standard deviation* (°) of the recorded gaze position in a 100-ms window centered on a sample. Another common noise measure (Holmqvist et al., 2011)
bcea	*bivariate contour ellipse area* ($°^2$). Measures the area in which the recorded gaze position in a 100-ms window is for P% of the time (Blignaut and Beelders, 2012). $P = 68$
disp	*dispersion* (°). The most common measure in dispersion-based algorithms (Salvucci & Goldberg, 2000). Calculated as $(x_{max} - x_{min}) + (y_{max} - y_{min})$ over a 100-ms window
vel, acc	*velocity* (°/s) and *acceleration* (°/s^2), calculated using a Savitzky–Golay filter with polynomial order 2 and a window size of 12 ms—half the duration of shortest saccade, as suggested by Nyström and Holmqvist (2010)
med-diff	*distance* (°) between the median gaze in a 100-ms window before the sample, and an equally sized window after the sample. Proposed by Olsson (2007)
mean-diff	*distance* (°) between the mean gaze in a 100-ms window before the sample, and an equally sized window after the sample. Proposed by Olsson (2007) and used in the default fixation detection algorithm in Tobii Studio
Rayleightest	a feature used by Larsson et al. (2015) that indicates whether the sample-to-sample directions in a 22-ms window are uniformly distributed
i2mc	introduced by Hessels et al. (2016) to find saccades in very noisy data. We used the final weights provided by the two-means clustering procedure as generated by the original implementation of the algorithm. A window size of 200 ms, centered on the sample was used
rms-diff, std-diff, bcea-diff	features inspired by Olsson (2007), but instead of differences in position, we take the difference between noise measures calculated for 100-ms windows preceding and succeeding the sample

A minimum of three samples are used in case there are not enough samples in the defined window, as may happen for lower frequency data

Zemblys Paper

WebGazer data | Features

	timestamp	x	y	hz	distance	velocity	acceleration
401	127833	420.594802	760.075459	19.230769	42.527993	0.817846	0.015728
402	127890	423.654510	756.349346	17.543860	4.821382	0.084586	0.001484
403	127946	395.596778	701.915289	17.857143	61.239717	1.093566	0.019528



https://upload.wikimedia.org/wikipedia/commons/7/76/Random_forest_diagram_complete.png

PROBLEM

For all Machine Learning Systems we need

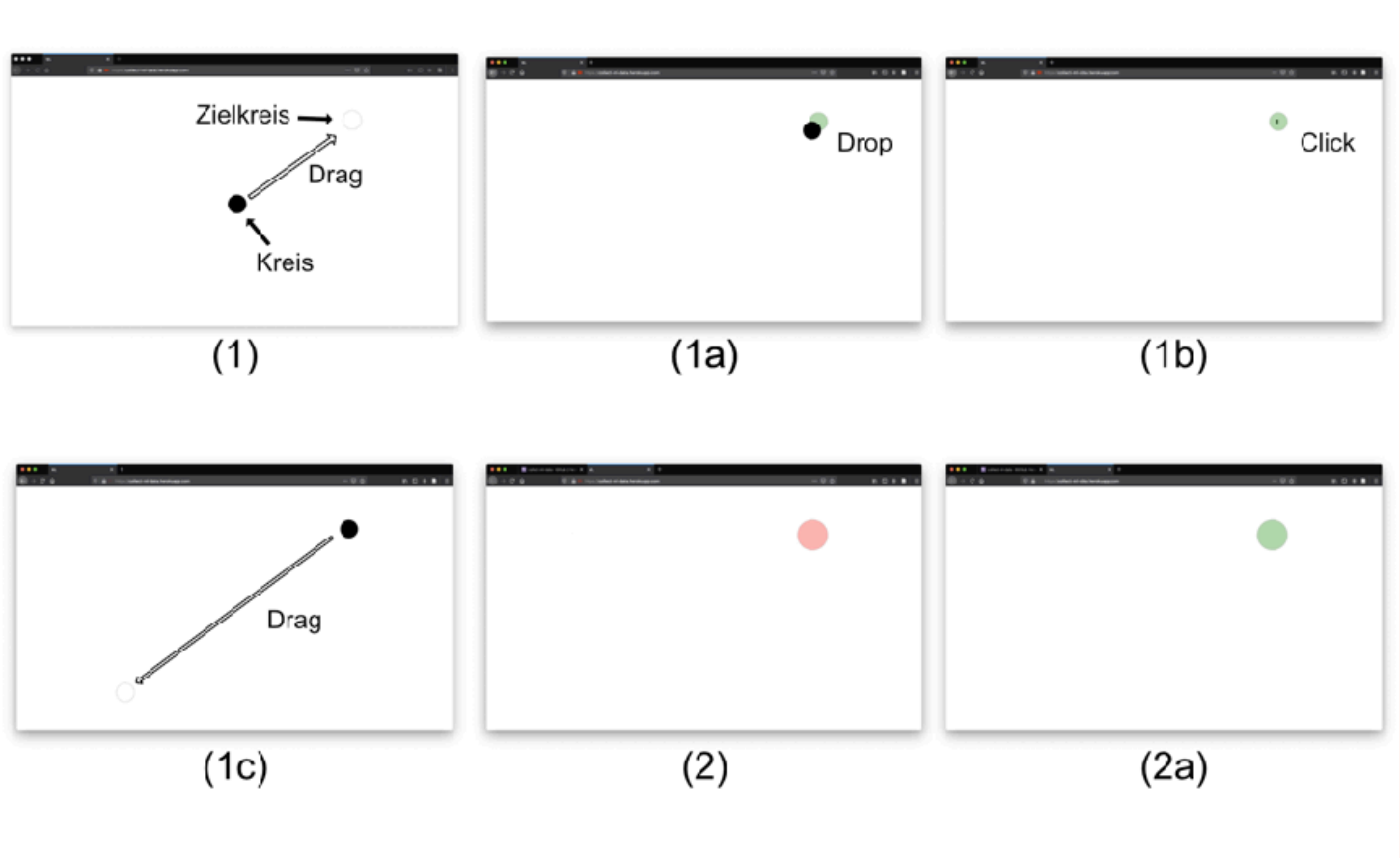
classified

data for training.

timestamp	x	y	label
127833	420.595	760.075	???
127890	423.655	756.349	???
127946	395.597	701.915	???

We don't have.

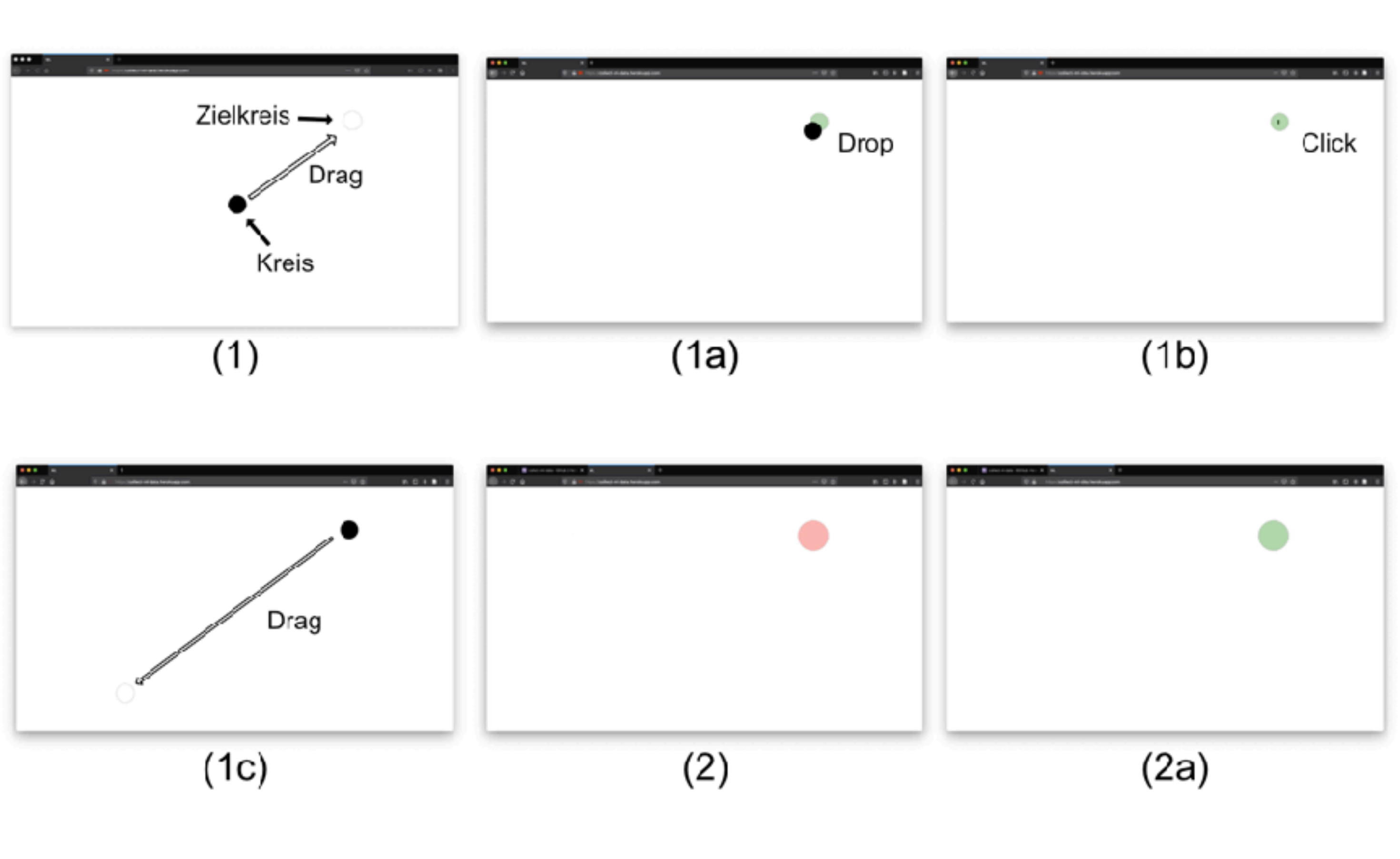
FIRST APPROACH



(1) (1a) (1b)
(1c) (2) (2a)

FIRST APPROACH

- **Heuristic**
  - **Drag & Drop –> *Saccade***
  - **Click –> *Fixation***



DATA PER SESSION

Beispielobjekt aus der Datenbank


```
_id: ObjectId("5f4e0828a6ba17062570a2bd")
timestamp: 1598949413945
name: "Daniel"
glasses: "true"
browser: "Netscape"
windowInnerWidth: 1440
windowInnerHeight: 803
marginHeight: 80.30000000000001
marginWidth: 288
> calibrationTargets: Array
> clickTargets: Array
> gazeTargets: Array
> calibrationData: Array
> clickData: Array
> gazeData: Array
```

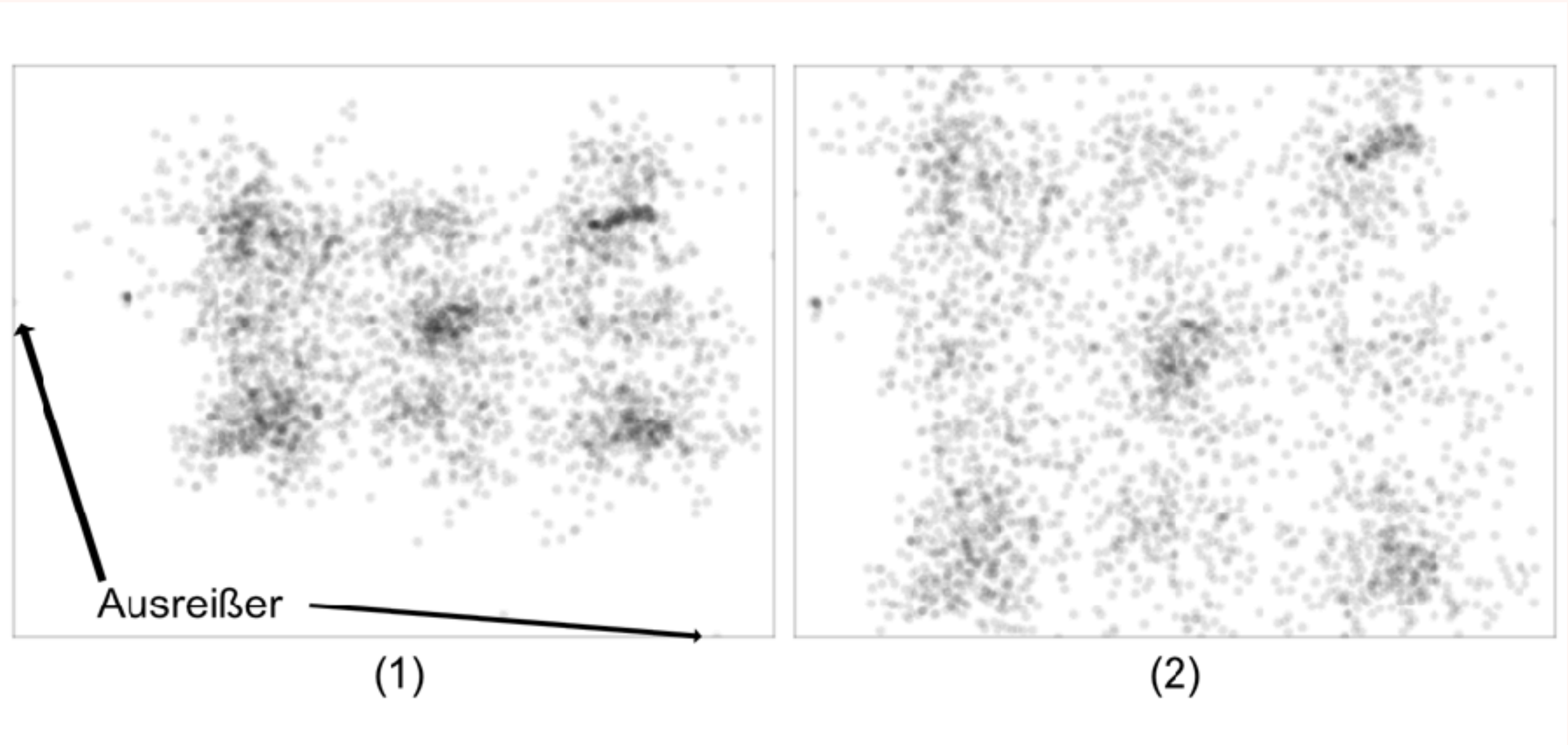

Array von Koordinaten der Zielkreise

x : Float
y : Float

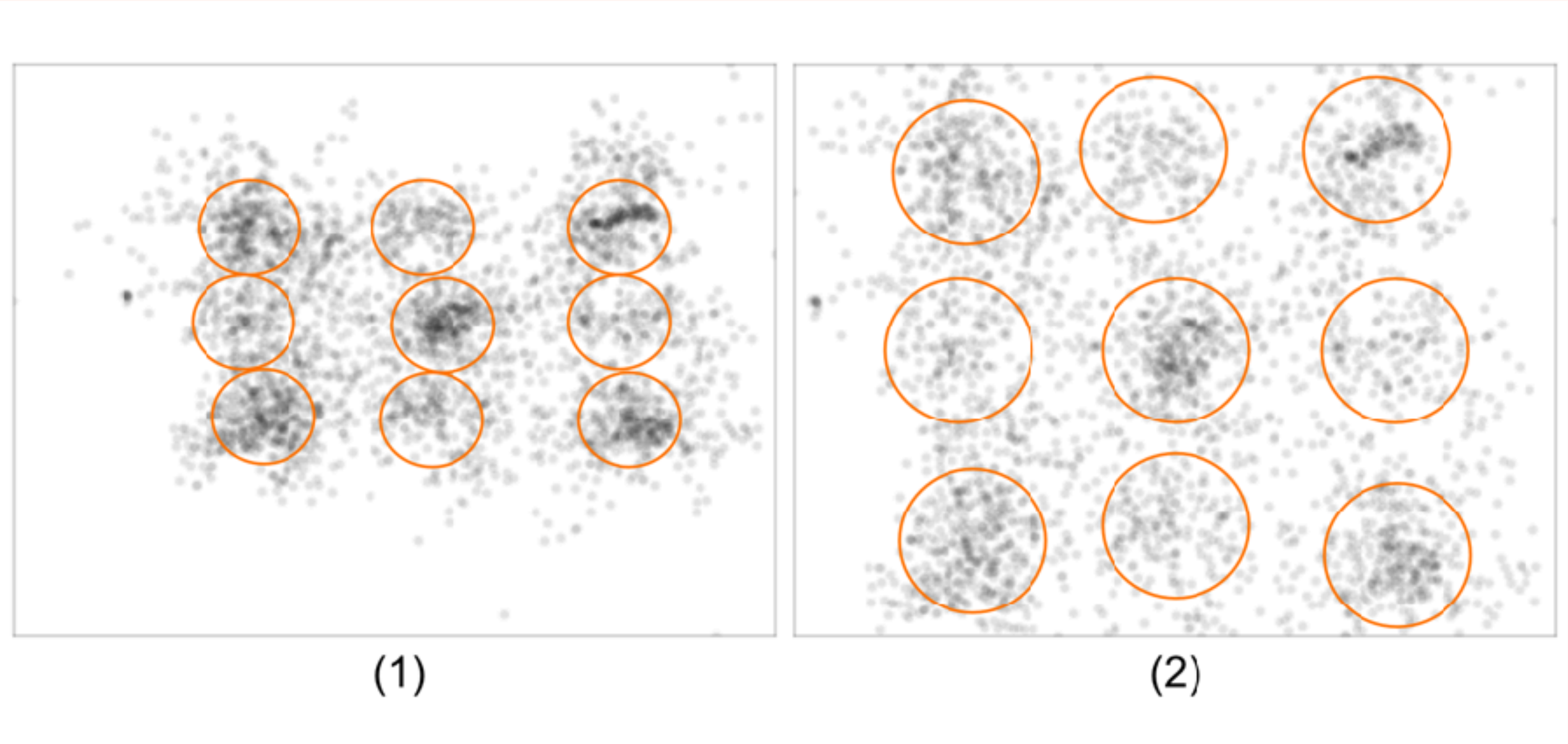
Array von WebGazer-Daten mit Label

timestamp : Float
x : Float
y : Float
label : String ("fixation" | "saccade")

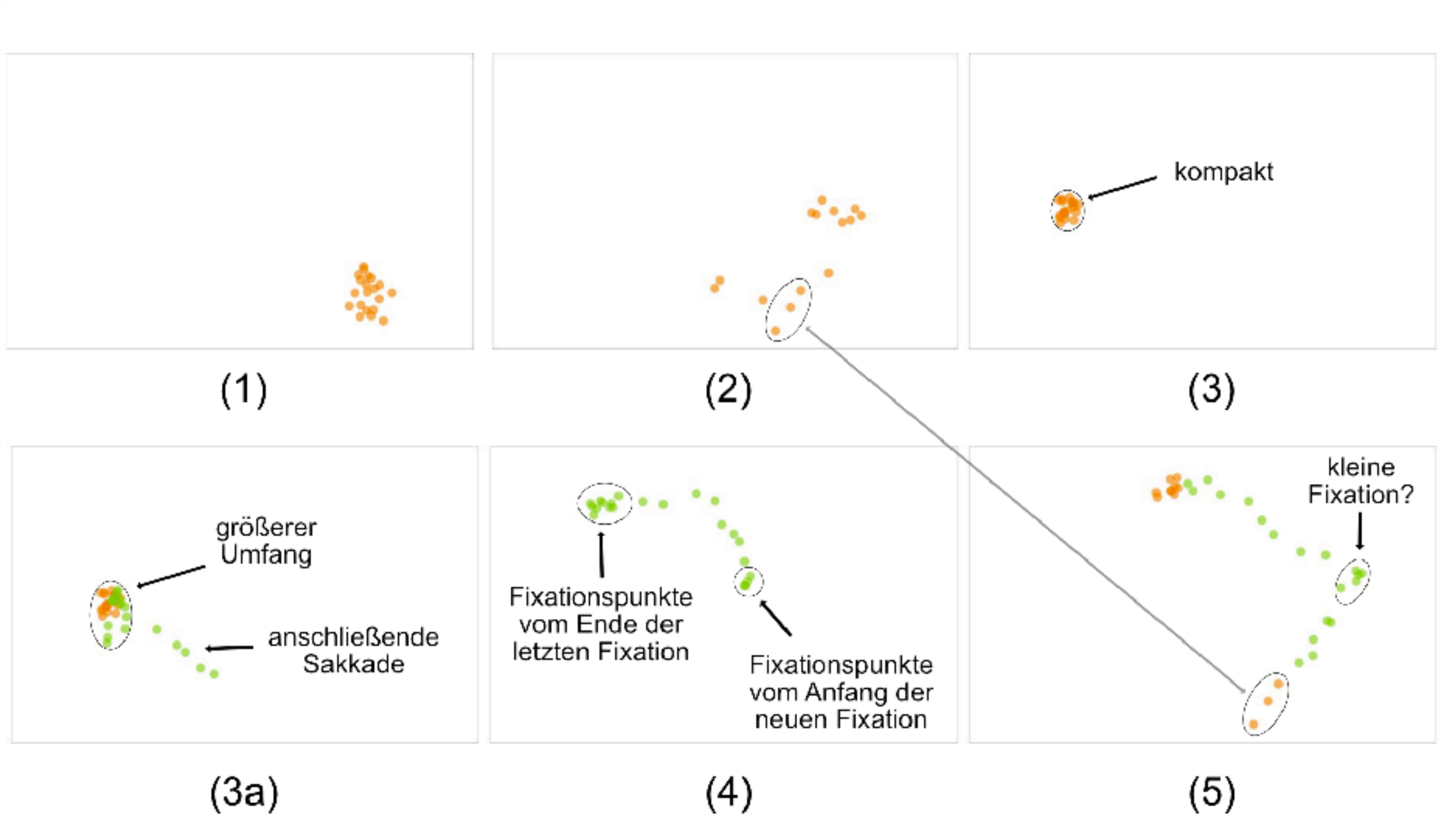
DATA FROM 2 SESSIONS



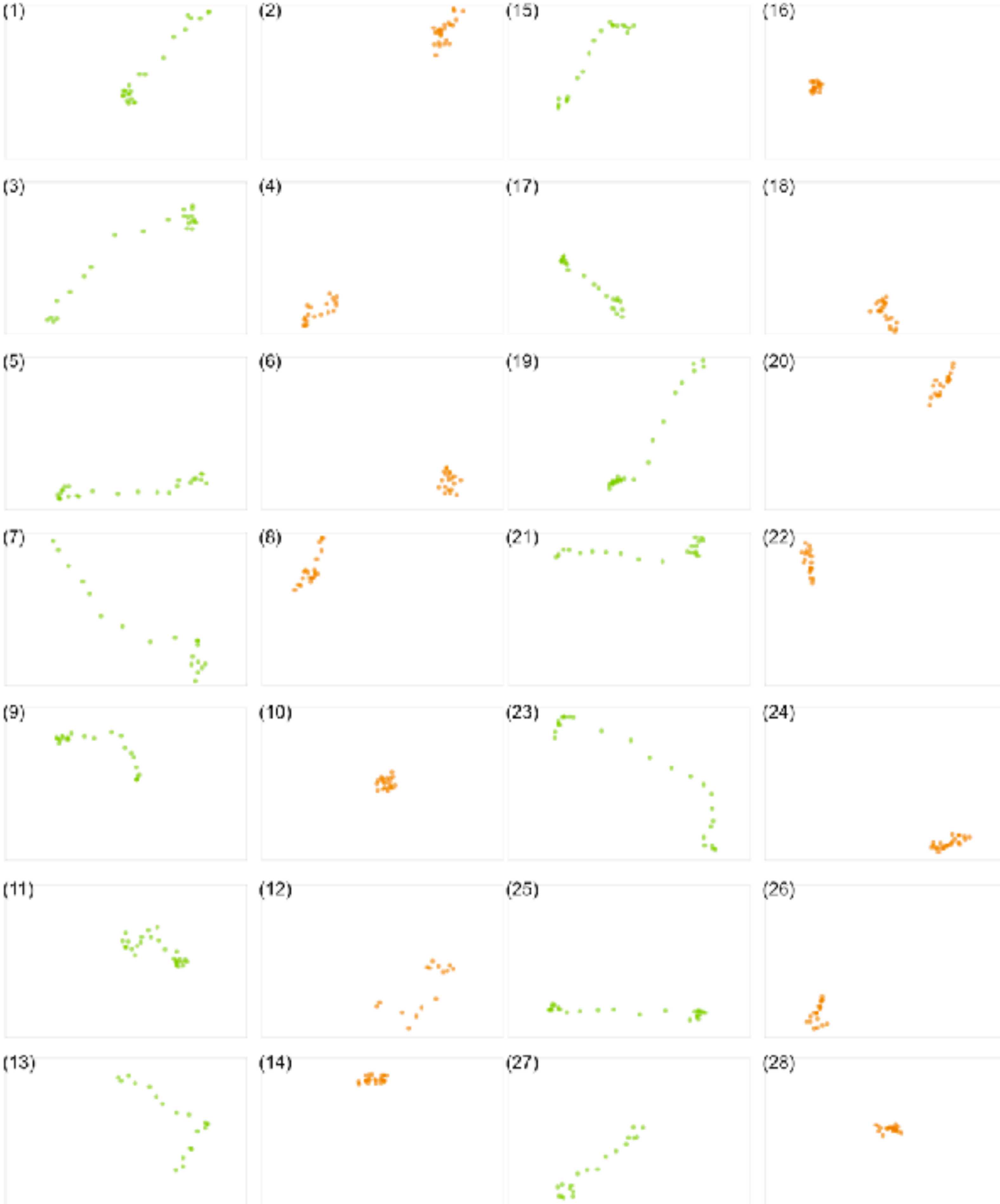
DATA FROM 2 SESSIONS

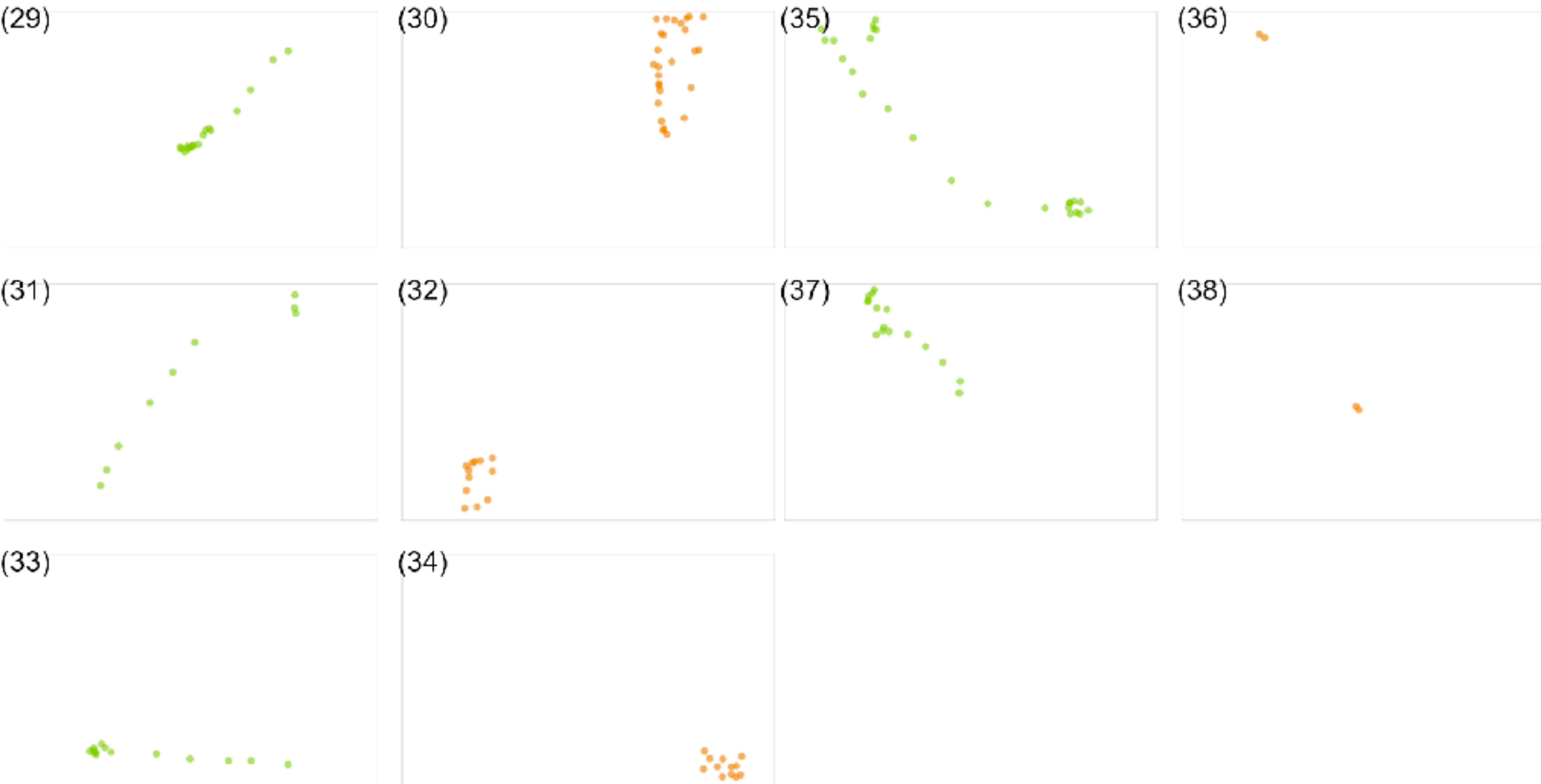


MORE DETAIL



kompakt
(1)
(2)
(3)
größerer Umfang
anschließende Sakkade
Fixationspunkte vom Ende der letzten Fixation
Fixationspunkte vom Anfang der neuen Fixation
kleine Fixation?
(3a)
(4)
(5)


(1)
(2)
(15)
(16)
(3)
(4)
(17)
(18)
(5)
(6)
(19)
(20)
(7)
(8)
(21)
(22)
(9)
(10)
(23)
(24)
(11)
(12)
(25)
(26)
(13)
(14)
(27)
(28)


(29)
(30)
(35)
(36)
(31)
(32)
(37)
(38)
(33)
(34)

RESULT FIRST APPROACH

- Heuristic works
- WebGazer can produce *Fixations* and *Saccades*
- Simple algorithms can detect those events
  - finding the correct thresholds is difficult
  - up to 85 % could be classified correctly
- Data can be created easily
  - can be done by everyone
  - can be repeated infinitely by the same person
- Heuristic can be tweaked by hand labeling

RESULT FIRST APPROACH

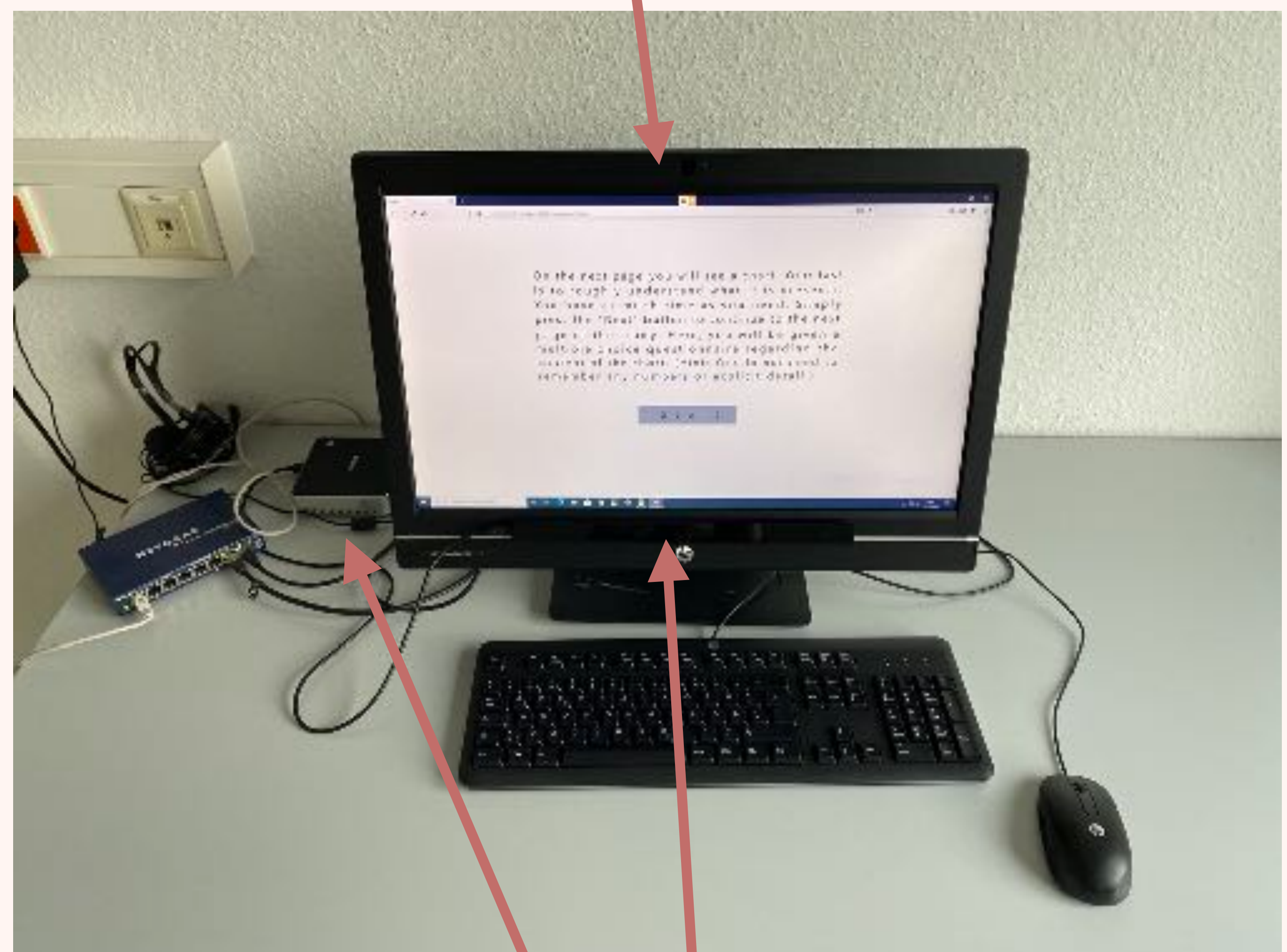
Can we use this **classified** dataset
to train a Machine Learning Model?

No.

- Heuristic is good, but needs correction by hand
  - hand labeling is time consuming
  - and needs an expert
- Data is not realistic enough

SECOND APPROACH

Webcam



Tobii Pro System

MAIN PAGE

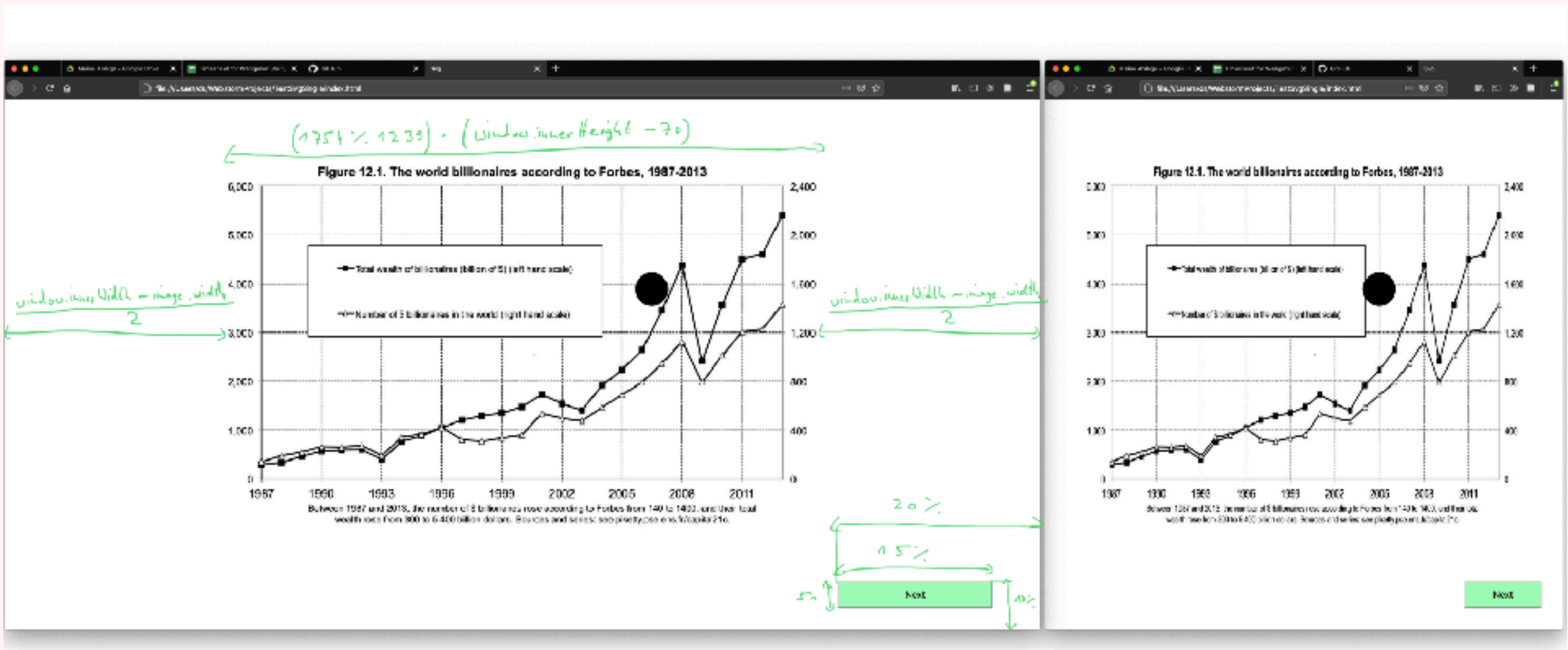
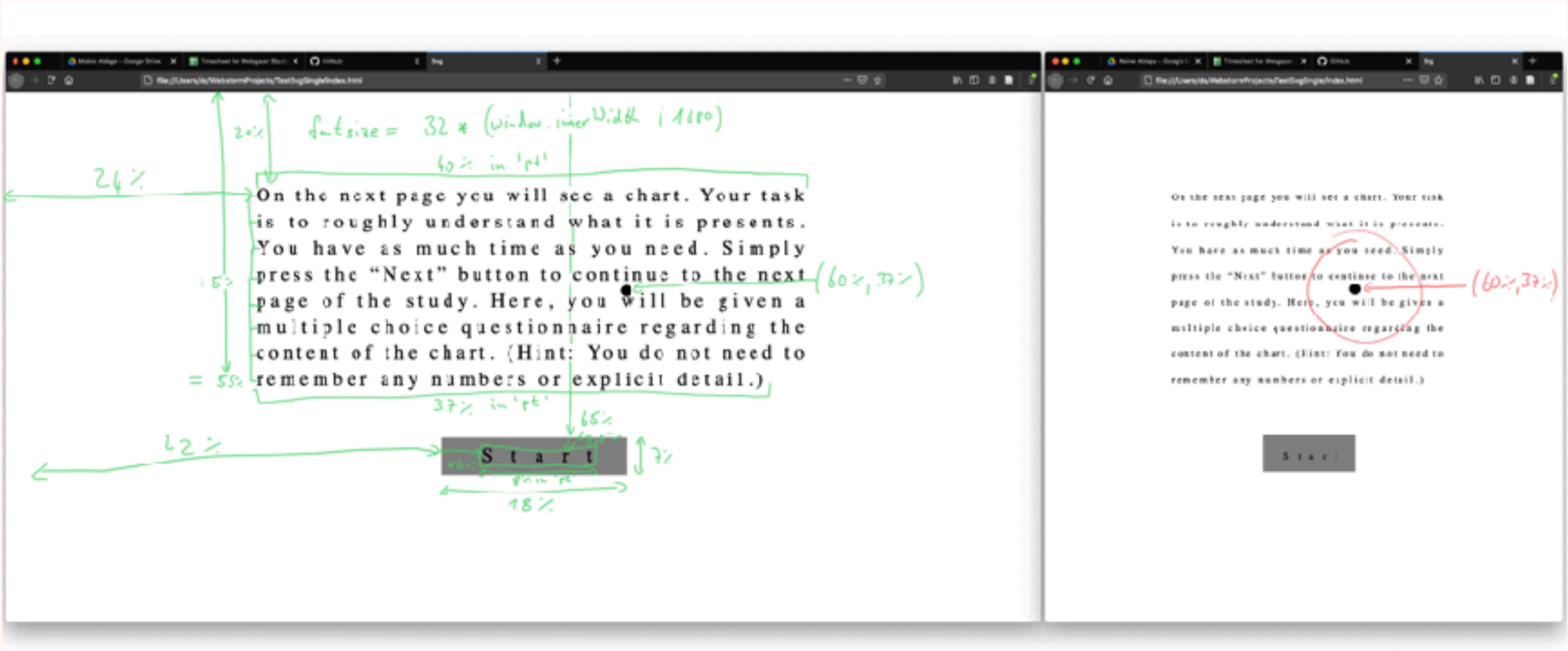
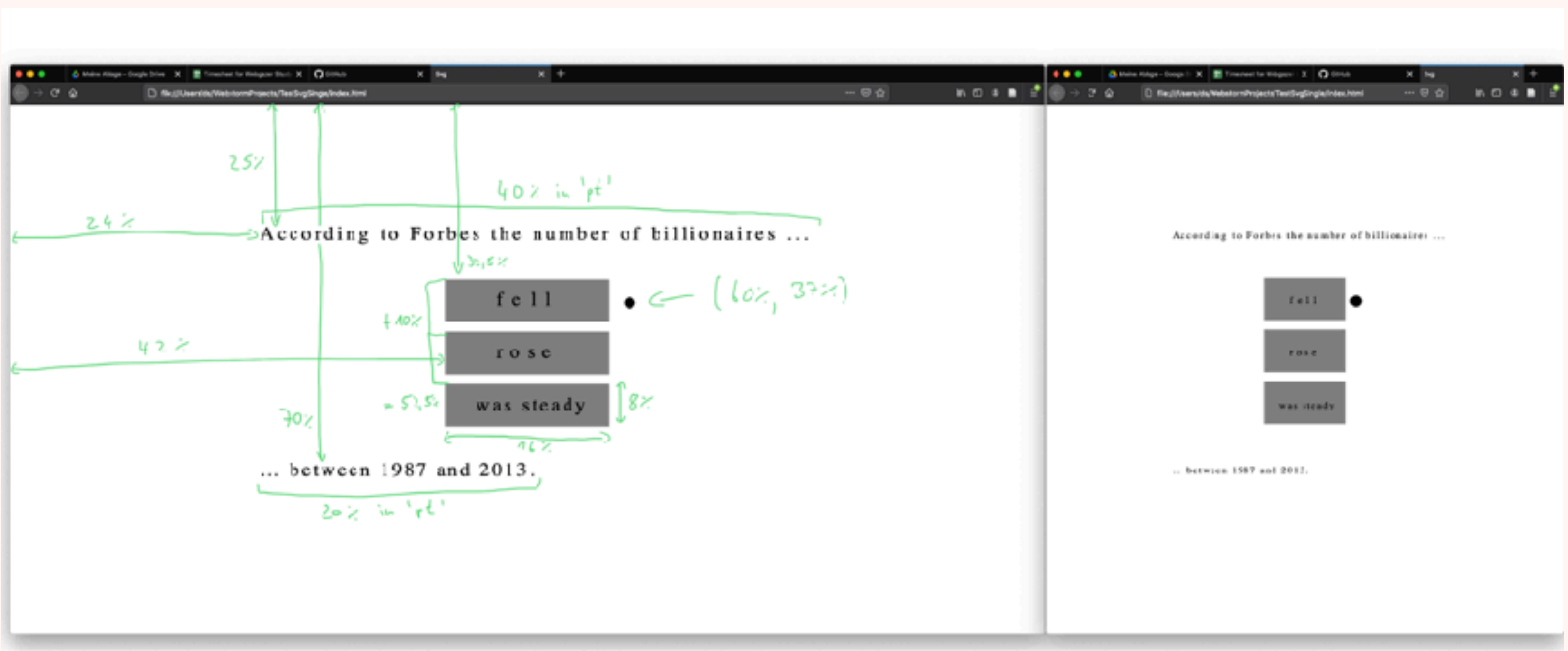
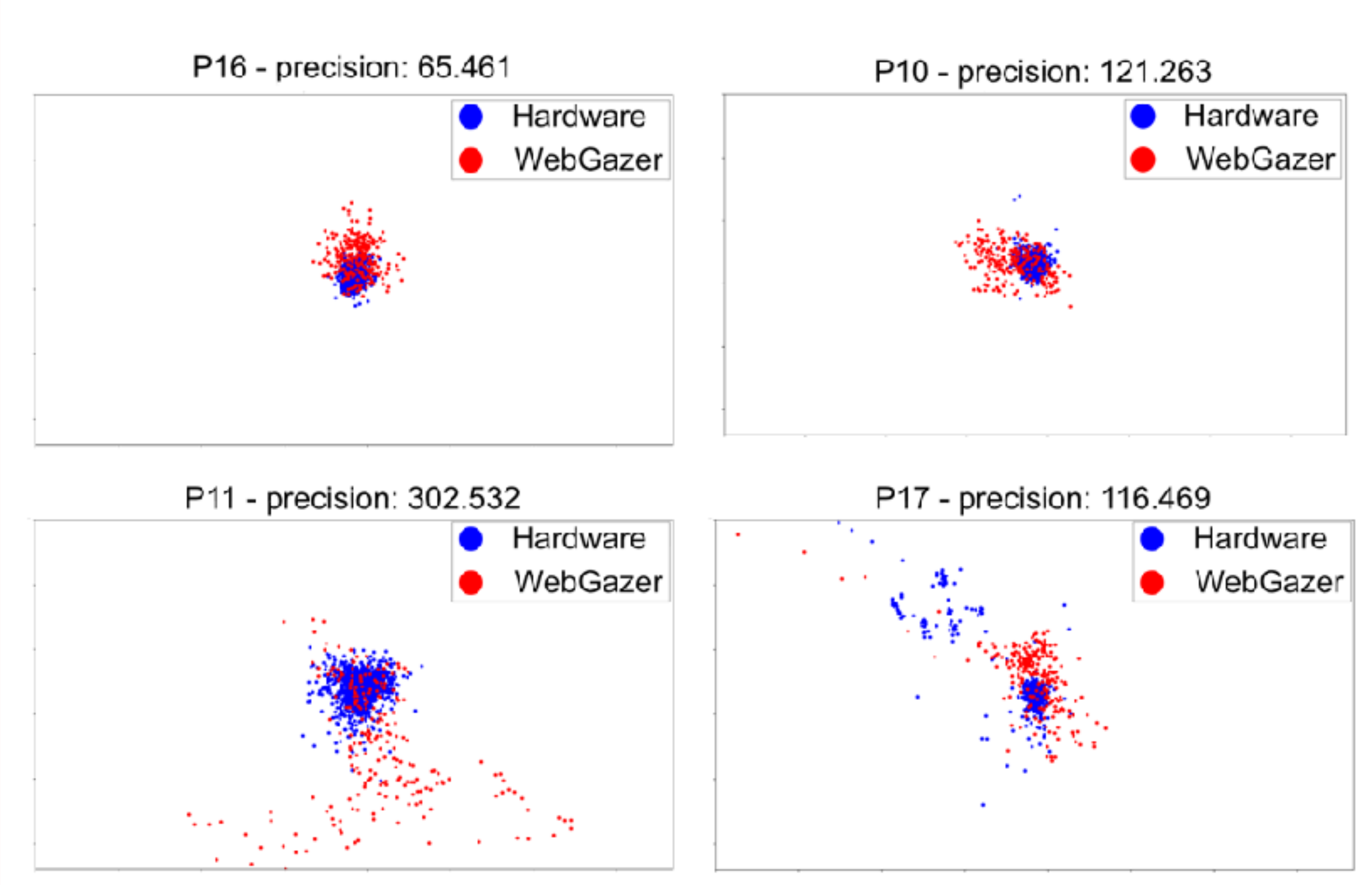


Figure 12.1. The world billionaires according to Forbes, 1987-2013
2,400
20%
15%
Next
Figure 12.1. The world billionaires according to Forbes, 1987-2013
Next


fontsize = 32 * (window.innerWidth / 1680)
60% in 'pt'
24%
20%
On the next page you will see a chart. Your task is to roughly understand what it is presents. You have as much time as you need. Simply press the “Next” button to continue to the next page of the study. Here, you will be given a multiple choice questionnaire regarding the content of the chart. (Hint: You do not need to remember any numbers or explicit detail.)
(60%, 37%)
= 55%
37% in 'pt'
65%
42%
S t a r t
7%
18%

25%
40% in 'pt'
24%
According to Forbes the number of billionaires ...
fell
rose
was steady
(60%, 37%)
47%
70%
8%
16%
... between 1987 and 2013.
20% in 'pt'

CALIBRATION

PRECISION


P16 - precision: 65.461
Hardware
WebGazer
P10 - precision: 121.263
Hardware
WebGazer
P11 - precision: 302.532
Hardware
WebGazer
P17 - precision: 116.469
Hardware
WebGazer

DATA

	timestamp	x	y
0	125382	805.724813	598.797895
1	125442	830.150531	575.433364
2	125527	853.116613	571.833358
3	125578	840.831675	574.131049
4	125628	841.620069	571.197856
...	...	...	...
1098	194304	992.337366	583.112148
1099	194358	939.116101	544.840858
1100	194409	925.749212	530.538707
1101	194490	928.691400	529.659512
1102	194542	1021.360609	556.513418

duration = 194542 - 125382
= 69160

WebGazer-Daten

	timestamp	GazePointX (ADCSpx)	GazePointY (ADCSpx)	GazeEventType
17150	1.603962e+12	932.0	494.0	Fixation
17151	1.603962e+12	931.0	513.0	Fixation
17152	1.603962e+12	931.0	527.0	Fixation
17153	1.603962e+12	970.0	551.0	Fixation
17154	1.603962e+12	984.0	553.0	Fixation
...	...	...	...	...
25449	1.603962e+12	889.0	490.0	Fixation
25450	1.603962e+12	890.0	488.0	Saccade
25451	1.603962e+12	903.0	457.0	Unclassified
25452	1.603962e+12	898.0	492.0	Saccade
25453	1.603962e+12	845.0	557.0	Saccade

> timestampForSync < timestampForSync + duration

Tobii-Pro-Daten

DATA COMBINATION

	timestamp	timeDiff	x	y
0	125382	0	805.724813	598.797895
1	125442	60	830.150531	575.433364
2	125527	145	853.116613	571.833358
3	125578	196	840.831675	574.131049
4	125628	246	841.620069	571.197856
...	...	...	...	...
1098	194304	68922	992.337366	583.112148
1099	194356	68975	939.116101	544.840856
1100	194409	69027	925.749212	530.538707
1101	194490	69108	928.691400	529.659512
1102	194542	69160	1021.360609	556.513416

	timestamp	timeDiff	GP_X	GP_Y	GazeEventType
17150	1.603962e+12	0.0	932.0	494.0	Fixation
17151	1.603962e+12	8.0	931.0	513.0	Fixation
17152	1.603962e+12	17.0	931.0	527.0	Fixation
17153	1.603962e+12	25.0	970.0	551.0	Fixation
17154	1.603962e+12	33.0	984.0	553.0	Fixation
...	...	...	...	...	...
25449	1.603962e+12	69126.0	889.0	490.0	Fixation
25450	1.603962e+12	69134.0	890.0	488.0	Saccade
25451	1.603962e+12	69142.0	903.0	457.0	Unclassified
25452	1.603962e+12	69151.0	898.0	492.0	Saccade
25453	1.603962e+12	69159.0	845.0	557.0	Saccade

	x	y	timeDiff	GazePointX (ADCSpx)	GazePointY (ADCSpx)	GazeEventType
0	805.725	598.798	0.0	932	494	Fixation
1			8.0	931	513	Fixation
2			17.0	931	527	Fixation
3			25.0	970	551	Fixation
4			33.0	984	553	Fixation
5			42.0	948	531	Fixation
6			50.0	985	549	Fixation
7			58.0	971	557	Fixation
8304	830.151	575.433	60.0			
8			67.0	973	537	Fixation
9			75.0	930	557	Fixation
10			83.0	976	562	Fixation
11			92.0	948	567	Fixation
12			100.0	934	567	Fixation
13			108.0	954	558	Fixation
14			117.0	955	558	Fixation
15			125.0	967	546	Fixation
16			133.0			Fixation
17			142.0	956	550	Fixation
8305	853.117	571.833	145.0			
18			150.0	964	552	Fixation

(before)

(after)

60 - 30 = 30

60 + 42.5 = 102.5

DATA COMBINATION

	x	y	timeDiff	GazePointX (ADCSpx)	GazePointY (ADCSpx)	GazeEventType
0	805.725	598.798	0.0	932	494	Fixation
1			8.0	931	513	Fixation
2			17.0	931	527	Fixation
3			25.0	970	551	Fixation
4			33.0	984	553	Fixation
5			42.0	948	531	Fixation
6			50.0	985	549	Fixation
7			58.0	971	557	Fixation
8304	830.151	575.433	60.0			
8			67.0	973	537	Fixation
9			75.0	930	557	Fixation
10			83.0	976	562	Fixation
11			92.0	948	567	Fixation
12			100.0	934	567	Fixation
13			108.0	954	558	Fixation
14			117.0	955	558	Fixation
15			125.0	967	546	Fixation
16			133.0			Fixation
17			142.0	956	550	Fixation
8305	853.117	571.833	145.0			
18			150.0	964	552	Fixation

(before) 60 - 30 = 30

(after) 60 + 42.5 = 102.5

DATA RESULT

We now have:

- **data from 16 sessions**
- **17955 classified data points**
  - **Fixation: 13458**
  - **Saccade: 3126**
  - **Unclassified: 1371**

	timestamp	x	y	hardXMean	hardYMean	labelMax
0	125382	805.724813	598.797895	941.000000	521.250000	Fixation
1	125442	830.150531	575.433364	961.000000	553.333333	Fixation
2	125527	853.116613	571.833358	963.571429	553.714286	Fixation
3	125578	840.831675	574.131049	961.333333	558.500000	Fixation
4	125628	841.620069	571.197856	884.857143	409.857143	Saccade
5	125699	842.518537	575.987936	808.875000	253.500000	Fixation
6	125760	874.386016	589.937787	735.428571	243.000000	Saccade
7	125810	904.827436	570.694648	564.571429	208.142857	Fixation
8	125881	927.766794	559.070929	561.571429	207.714286	Fixation
9	125932	889.724340	488.029719	564.142857	217.000000	Fixation
10	125992	866.871291	433.887626	628.571429	221.714286	Saccade
11	126046	816.148375	365.926099	706.000000	221.000000	Fixation
12	126125	657.384092	300.498369	745.375000	221.125000	Saccade
13	126174	606.804005	263.939244	860.571429	242.000000	Fixation
14	126243	595.516829	220.890731	866.000000	252.400000	Unclassified
15	126310	572.425903	178.137413	NaN	NaN	Unclassified

RESULT SECOND APPROACH

Can we use this **classified** dataset
to train a Machine Learning Model?

No.

- Method of data collection is good
- Data is realistic
- Synchronizing of data was successful
- BUT: Classification seems to be wrong

PROBLEMATIC CLASSIFICATION