<Calculating the Accuracy of Eye Tracker Devices and the Impact of Fixation Algorithms on this Accuracy>

Prepared by
Furkan Duman

Middle East Technical University Northern Cyprus Campus

Computer Engineering Department

01/01/2024

Contents

List of Figures

1 Purpose

This study aims to explore the process of calculating the accuracy of eye-tracking devices. Through my research, I found that typically, the accuracy is determined after calibration tests by analyzing the correct gaze ratio obtained from focusing on specific areas. In my previous research, I couldn't find a dedicated system for testing eye-tracker accuracy, so I decided to develop my own solution to address this gap.

The other primary motivation for starting the project was the insufficient number of studies on the accuracy of fixation algorithms. Expressing the accuracy of fixation algorithms mathematically is believed to further facilitate the calculation of "system accuracy" for the EASIEST project. Therefore, this study takes a step towards a thorough understanding of the process of determining the accuracy of eye-tracking devices and how fixation algorithms impact this accuracy.

2 Factors in Data Quality

I am aware that designing the lighting system well is crucial to achieve accurate results during eye tracking. During the tests, I contributed to obtaining more precise and accurate data by directing all lighting systems towards the face of the person conducting the test. Additionally, to ensure a more stable experience, the test participant aimed to minimize head movements during the tests.

Kerem Keptiğ played a significant role in integrating the eye tracker device into my system. I thank him very much for integrating the current most seamless version of Webgezer into our system, and we tried to achieve the best efficiency with this configuration.

3 Methods

In this **flask** project, first I created an HTML for the test screen. To calculate the accuracy of eye tracking devices, I created a total of 48 cells and determined the individual coordinates for each cell. During the test, the participant looks at cells from the 1st to the 48th every 5 seconds. After completing the test, the raw dataset provides X, Y, and time values of eye data. Using these values, the determination of which eye data corresponds to which cell is calculated. Subsequently, for each 5-second time interval, the analysis of which cell the eye data belongs to is conducted, and based on this analysis, the accuracy rate is calculated.

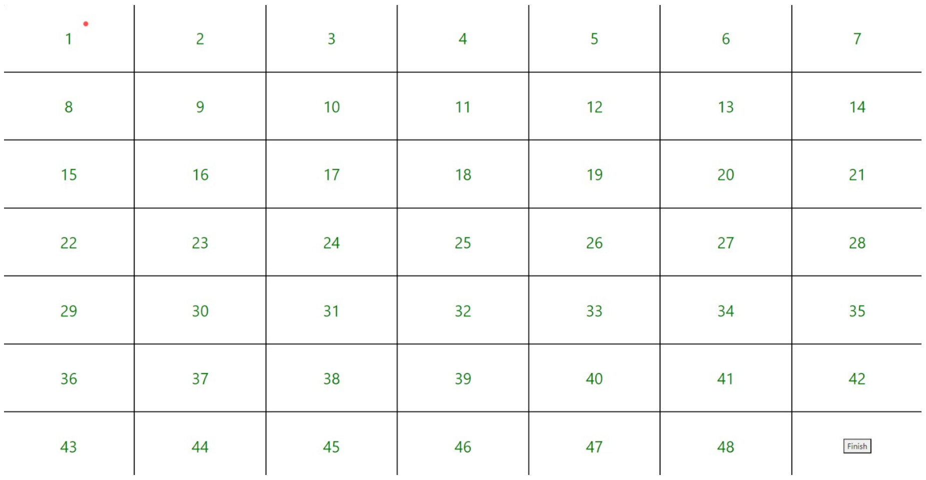
When calculating the accuracy rate, it is crucial to consider the time intervals. For example, if the participant looks at the 1st cell and its neighboring cells (8th and 9th cells) between 0 and 5 seconds, this gaze is classified as a correct point. If the gaze goes elsewhere, those points are classified as incorrect points. The accuracy rate of the 1st cell between 0 and 5 seconds is then calculated by dividing the number of correct points by the total eye data. **This process is repeated for each time interval.**

Subsequently, the average accuracy value of the system is calculated by taking the arithmetic mean of the accuracy values calculated for all 48 time intervals.

Later on, I obtained fixation points from the raw data collected during the test using fixation algorithms. I observed how fixation algorithms had an impact on the accuracy of the system by analyzing these fixation points.

To make it easier to understand how often cells are looked at in specific time intervals, I made pie charts for each time period using the **matplotlib** library. These charts show visually how much attention cells get, helping us see how well the system works at different times.

3.1 Test Screen



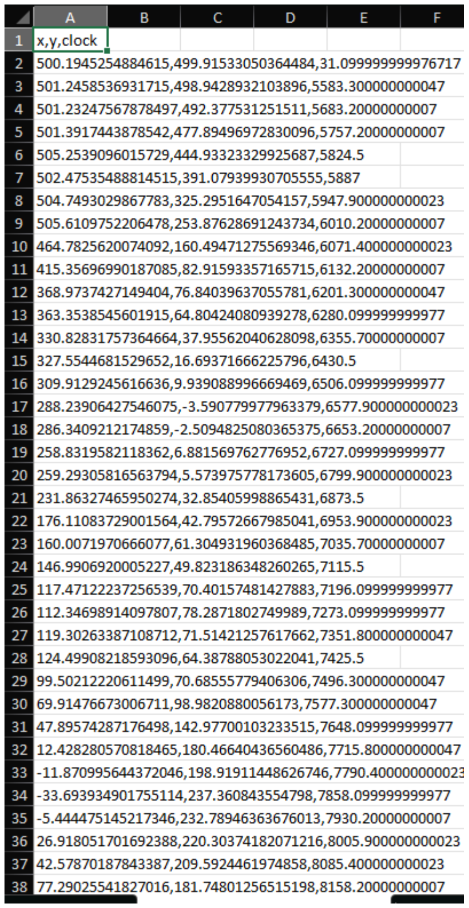
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	Finish

Figure 1: Test Screen

In this test display, there are a total of 48 cells.n. WebGazer usually starts recording eye data from the center of the screen, so when the eye data first arrives at the 1st cell, the system time is considered 0. Subsequently, every 5 seconds, the gaze shifts to the next cell. The entire test duration is calculated by multiplying 48 cells by 5 seconds, resulting in a total of 240 seconds.

At the end of the test, the raw data is transmitted to the system in the form of X, Y, and time metrics.

3.2 Eye Tracker Data Analysis and Find the Cells

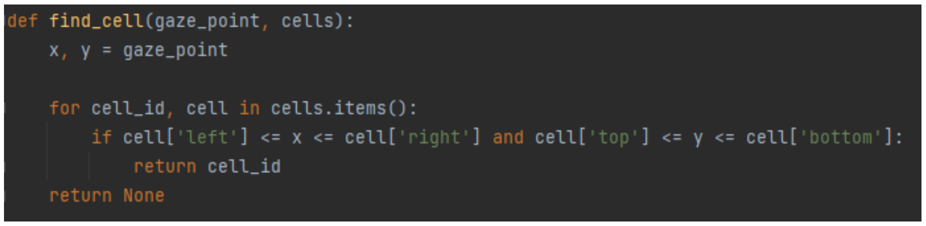


	A	B	C	D	E	F
1	x,y,clock					
2	500.1945254884615,499.91533050364484,31.099999999976717					
3	501.2458536931715,498.9428932103896,5583.300000000047					
4	501.23247567878497,492.377531251511,5683.20000000007					
5	501.3917443878542,477.89496972830096,5757.20000000007					
6	505.2539096015729,444.93323329925687,5824.5					
7	502.47535488814515,391.07939930705555,5887					
8	504.7493029867783,325.2951647054157,5947.900000000023					
9	505.6109752206478,253.87628691243734,6010.20000000007					
10	464.7825620074092,160.49471275569346,6071.400000000023					
11	415.35696990187085,82.91593357165715,6132.20000000007					
12	368.9737427149404,76.84039637055781,6201.300000000047					
13	363.3538545601915,64.80424080939278,6280.099999999977					
14	330.82831757364664,37.95562040628098,6355.70000000007					
15	327.5544681529652,16.69371666225796,6430.5					
16	309.9129245616636,9.939088996669469,6506.099999999977					
17	288.23906427546075,-3.590779977963379,6577.900000000023					
18	286.3409212174859,-2.5094825080365375,6653.20000000007					
19	258.8319582118362,6.881569762776952,6727.099999999977					
20	259.29305816563794,5.573975778173605,6799.900000000023					
21	231.86327465950274,32.85405998865431,6873.5					
22	176.11083729001564,42.79572667985041,6953.900000000023					
23	160.0071970666077,61.304931960368485,7035.70000000007					
24	146.9906920005227,49.823186348260265,7115.5					
25	117.47122237256539,70.40157481427883,7196.099999999977					
26	112.34698914097807,78.2871802749989,7273.099999999977					
27	119.30263387108712,71.51421257617662,7351.800000000047					
28	124.49908218593096,64.38788053022041,7425.5					
29	99.50212220611499,70.68555779406306,7496.300000000047					
30	69.91476673006711,98.9820880056173,7577.300000000047					
31	47.89574287176498,142.97700103233515,7648.099999999977					
32	12.428280570818465,180.46640436560486,7715.800000000047					
33	-11.870995644372046,198.91911448626746,7790.400000000023					
34	-33.693934901755114,237.360843554798,7858.099999999977					
35	-5.444475145217346,232.78946363676013,7930.20000000007					
36	26.918051701692388,220.30374182071216,8005.900000000023					
37	42.57870187843387,209.5924461974858,8085.400000000023					
38	77.29025541827016,181.74801256515198,8158.20000000007					

Figure 2: Extract the raw data

Raw data can contain thousands of rows for 240 seconds of data. I take all of these gaze points as x, y and clock, just like in the EASIEST project.

Then, I find which cell each point belongs to, according to the cell coordinates I calculated before. I use the following function for this.



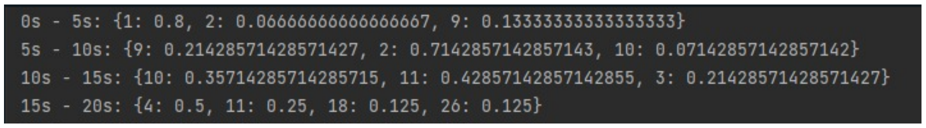
```
def find_cell(gaze_point, cells):
    x, y = gaze_point

    for cell_id, cell in cells.items():
        if cell['left'] <= x <= cell['right'] and cell['top'] <= y <= cell['bottom']:
            return cell_id
    return None
```

Figure 3: Function that finds which gaze points belong to which cell

3.3 Eye Data Focusing Rates

After analyzing the X, Y, and time values associated with eye movements, the system calculates the distribution of eye data across each time interval, determining the percentage of data points allocated to each cell.



```
0s - 5s: {1: 0.8, 2: 0.06666666666666667, 9: 0.13333333333333333}
5s - 10s: {9: 0.21428571428571427, 2: 0.7142857142857143, 10: 0.07142857142857142}
10s - 15s: {10: 0.35714285714285715, 11: 0.42857142857142855, 3: 0.21428571428571427}
15s - 20s: {4: 0.5, 11: 0.25, 18: 0.125, 26: 0.125}
```

Figure 4: Shows which cell the eye data belongs to at different time intervals

This image illustrates which cells the eye data corresponds to during specific time intervals. Typically, these data are calculated for up to 240 seconds, but I'm including this image briefly for explanation.

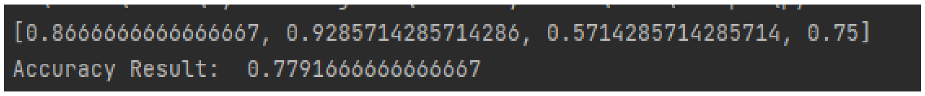
3.4 The method for obtaining test results

Then, the data is analyzed to determine which cells should be looked at during specific time intervals, and their respective rates are calculated. Subsequently, the arithmetic mean of these rates is obtained, resulting in the average accuracy calculation.

$$\text{Average Correct Area Viewing Ratio} = \frac{\text{Correct Area Viewing Ratio}_1 + \ldots + \text{Correct Area Viewing Ratio}_{48}}{48}$$

As I said before, correct look rates are calculated at each time interval, and since there are 48 cells, the average is taken by dividing by 48.

For a simpler explanation, I only showed the data up to the 20th second in the figure above. I will now calculate the accuracy rate for this data.



```
[0.8666666666666667, 0.9285714285714286, 0.5714285714285714, 0.75]
Accuracy Result:  0.7791666666666667
```

Figure 5: Accuracy Result

For instance, the value 0.8666666666666667 in this graph is calculated by summing the accuracy ratios of the 1st cell and its neighboring cells (neighbors for the 1st cell are the 8th and 2nd cells) between the 0th and 5th seconds. As an example, the last element in the list with a value of 0.75 is calculated as the total value of the 4th cell and its neighboring cells (ratios from the 4th cell are 0.5, and from the 11th cell, 0.25).

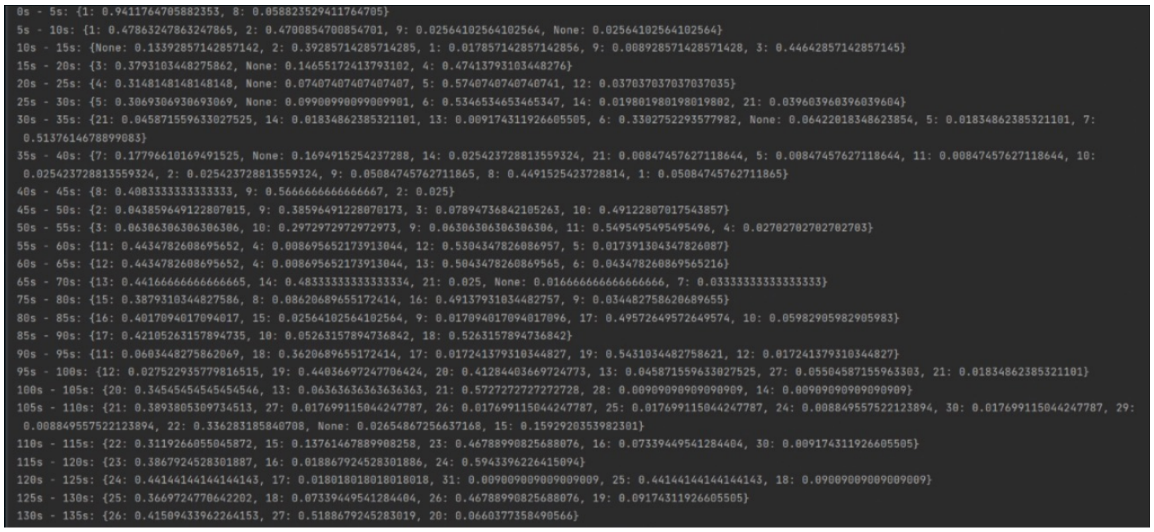
And the average accuracy was calculated by taking the average of these values.

I want to say again, I used only 4 time intervals for easier understanding, but the system has a total of 48 time intervals.

4 First Test

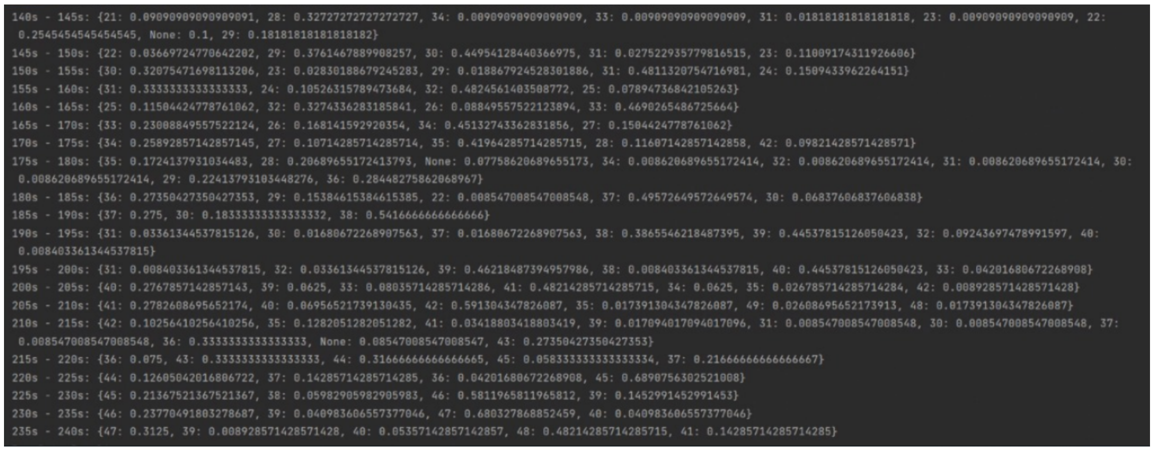
Kerem Keptiğ volunteered for the first system test, and I appreciate his participation. Thank you!

After setting up the necessary environment, the test started. A**t the beginning of the test, the calibration test of WebGazer yielded a result of 92%. In our test, the calculated value for accuracy was 87.788%.**



```
0s - 5s: {1: 0.9411764705882353, 8: 0.058823529411764705}
5s - 10s: {1: 0.47863247863247865, 2: 0.4700854700854701, 9: 0.02564102564102564, None: 0.02564102564102564}
10s - 15s: {None: 0.13392857142857142, 2: 0.39285714285714285, 1: 0.017857142857142856, 9: 0.008928571428571428, 3: 0.44642857142857145}
15s - 20s: {3: 0.3793103448275862, None: 0.14655172413793102, 4: 0.47413793103448276}
20s - 25s: {4: 0.3148148148148148, None: 0.07407407407407407, 5: 0.5740740740740741, 12: 0.037037037037037035}
25s - 30s: {5: 0.3069306930693069, None: 0.09900990099009901, 6: 0.5346534653465347, 14: 0.019801980198019802, 21: 0.039603960396039604}
30s - 35s: {21: 0.045871559633027525, 14: 0.01834862385321101, 13: 0.009174311926605505, 6: 0.3302752293577982, None: 0.06422018348623854, 5: 0.01834862385321101, 7:
 0.5137614678899083}
35s - 40s: {7: 0.17796610169491525, None: 0.1694915254237288, 14: 0.025423728813559324, 21: 0.00847457627118644, 5: 0.00847457627118644, 11: 0.00847457627118644, 10:
 0.025423728813559324, 2: 0.025423728813559324, 9: 0.05084745762711865, 8: 0.4491525423728814, 1: 0.05084745762711865}
40s - 45s: {8: 0.4083333333333333, 9: 0.5666666666666667, 2: 0.025}
45s - 50s: {2: 0.043859649122807015, 9: 0.38596491228070173, 3: 0.07894736842105263, 10: 0.49122807017543857}
50s - 55s: {3: 0.06306306306306306, 10: 0.2972972972972973, 9: 0.06306306306306306, 11: 0.5495495495495496, 4: 0.02702702702702703}
55s - 60s: {11: 0.4434782608695652, 4: 0.008695652173913044, 12: 0.5304347826086957, 5: 0.017391304347826087}
60s - 65s: {12: 0.4434782608695652, 4: 0.008695652173913044, 13: 0.5043478260869565, 6: 0.043478260869565216}
65s - 70s: {13: 0.44166666666666665, 14: 0.48333333333333334, 21: 0.025, None: 0.016666666666666666, 7: 0.03333333333333333}
75s - 80s: {15: 0.3879310344827586, 8: 0.08620689655172414, 16: 0.49137931034482757, 9: 0.034482758620689655}
80s - 85s: {16: 0.4017094017094017, 15: 0.02564102564102564, 9: 0.017094017094017096, 17: 0.49572649572649574, 10: 0.05982905982905983}
85s - 90s: {17: 0.42105263157894735, 10: 0.05263157894736842, 18: 0.5263157894736842}
90s - 95s: {11: 0.0603448275862069, 18: 0.3620689655172414, 17: 0.017241379310344827, 19: 0.5431034482758621, 12: 0.017241379310344827}
95s - 100s: {12: 0.027522935779816515, 19: 0.44036697247706424, 20: 0.41284403669724773, 13: 0.045871559633027525, 27: 0.05504587155963303, 21: 0.01834862385321101}
100s - 105s: {20: 0.34545454545454546, 13: 0.06363636363636363, 21: 0.5727272727272728, 28: 0.00909090909090909, 14: 0.00909090909090909}
105s - 110s: {21: 0.3893805309734513, 27: 0.017699115044247787, 26: 0.017699115044247787, 25: 0.017699115044247787, 24: 0.008849557522123894, 30: 0.017699115044247787, 29:
 0.008849557522123894, 22: 0.336283185840708, None: 0.02654867256637168, 15: 0.1592920353982301}
110s - 115s: {22: 0.3119266055045872, 15: 0.13761467889908258, 23: 0.46788990825688076, 16: 0.07339449541284404, 30: 0.009174311926605505}
115s - 120s: {23: 0.3867924528301887, 16: 0.018867924528301886, 24: 0.5943396226415094}
120s - 125s: {24: 0.44144144144144143, 17: 0.018018018018018018, 31: 0.009009009009009009, 25: 0.44144144144144143, 18: 0.09009009009009009}
125s - 130s: {25: 0.3669724770642202, 18: 0.07339449541284404, 26: 0.46788990825688076, 19: 0.09174311926605505}
130s - 135s: {26: 0.41509433962264153, 27: 0.5188679245283019, 20: 0.0660377358490566}
```

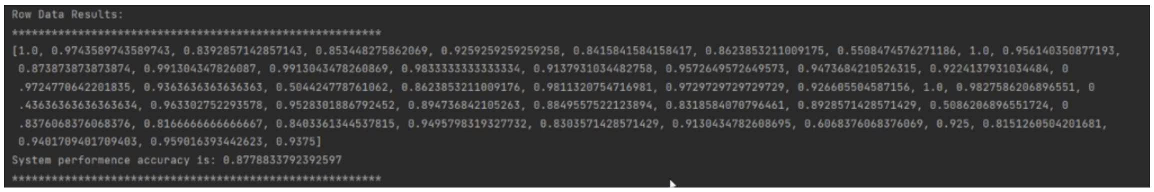
Figure 6: Eye gaze rates for each time period-PART1



```
140s - 145s: {21: 0.09090909090909091, 28: 0.32727272727272727, 34: 0.00909090909090909, 33: 0.00909090909090909, 31: 0.01818181818181818, 23: 0.00909090909090909, 22:
 0.2545454545454545, None: 0.1, 29: 0.18181818181818182}
145s - 150s: {22: 0.03669724770642202, 29: 0.3761467889908257, 30: 0.44954128440366975, 31: 0.027522935779816515, 23: 0.11009174311926606}
150s - 155s: {30: 0.32075471698113206, 23: 0.02830188679245283, 29: 0.018867924528301886, 31: 0.4811320754716981, 24: 0.1509433962264151}
155s - 160s: {31: 0.3333333333333333, 24: 0.10526315789473684, 32: 0.4824561403508772, 25: 0.07894736842105263}
160s - 165s: {25: 0.11504424778761062, 32: 0.3274336283185841, 26: 0.08849557522123894, 33: 0.4690265486725664}
165s - 170s: {33: 0.23008849557522124, 26: 0.168141592920354, 34: 0.4513274336283186, 27: 0.1504424778761062}
170s - 175s: {34: 0.25892857142857145, 27: 0.10714285714285714, 35: 0.41964285714285715, 28: 0.11607142857142858, 42: 0.09821428571428571}
175s - 180s: {35: 0.1724137931034483, 28: 0.20689655172413793, None: 0.07758620689655173, 34: 0.008620689655172414, 32: 0.008620689655172414, 31: 0.008620689655172414, 30:
 0.008620689655172414, 29: 0.22413793103448276, 36: 0.28448275862068967}
180s - 185s: {36: 0.27350427350427353, 29: 0.15384615384615385, 22: 0.008547008547008548, 37: 0.49572649572649574, 30: 0.06837606837606838}
185s - 190s: {37: 0.275, 30: 0.18333333333333332, 38: 0.5416666666666666}
190s - 195s: {31: 0.03361344537815126, 30: 0.01680672268907563, 37: 0.01680672268907563, 38: 0.3865546218487395, 39: 0.44537815126050423, 32: 0.09243697478991597, 40:
 0.008403361344537815}
195s - 200s: {31: 0.008403361344537815, 32: 0.03361344537815126, 39: 0.46218487394957986, 38: 0.008403361344537815, 40: 0.44537815126050423, 33: 0.04201680672268908}
200s - 205s: {40: 0.2767857142857143, 39: 0.0625, 33: 0.08035714285714286, 41: 0.48214285714285715, 34: 0.0625, 35: 0.026785714285714284, 42: 0.008928571428571428}
205s - 210s: {41: 0.2782608695652174, 40: 0.06956521739130435, 42: 0.591304347826087, 35: 0.017391304347826087, 49: 0.02608695652173913, 48: 0.017391304347826087}
210s - 215s: {42: 0.10256410256410256, 35: 0.1282051282051282, 41: 0.03418803418803419, 39: 0.017094017094017096, 31: 0.008547008547008548, 30: 0.008547008547008548, 37:
 0.008547008547008548, 36: 0.3333333333333333, None: 0.08547008547008547, 43: 0.27350427350427353}
215s - 220s: {36: 0.075, 43: 0.3333333333333333, 44: 0.31666666666666665, 45: 0.058333333333333334, 37: 0.21666666666666667}
220s - 225s: {44: 0.12605042016806722, 37: 0.14285714285714285, 36: 0.04201680672268908, 45: 0.6890756302521008}
225s - 230s: {45: 0.21367521367521367, 38: 0.05982905982905983, 46: 0.5811965811965812, 39: 0.1452991452991453}
230s - 235s: {46: 0.23770491803278687, 39: 0.040983606557377046, 47: 0.680327868852459, 40: 0.040983606557377046}
235s - 240s: {47: 0.3125, 39: 0.008928571428571428, 40: 0.05357142857142857, 48: 0.48214285714285715, 41: 0.14285714285714285}
```

Figure 7: Eye gaze rates for each time period-PART2

It was found that the eye data looked at which cells and at what rate in which seconds. Now, using this data, it will be calculated to what extent the eye data is looking at the right area.



```
Row Data Results:
********************************************************
[1.0, 0.9743589743589743, 0.8392857142857143, 0.853448275862069, 0.9259259259259258, 0.8415841584158417, 0.8623853211009175, 0.5508474576271186, 1.0, 0.956140350877193,
 0.873873873873874, 0.991304347826087, 0.9913043478260869, 0.9833333333333334, 0.9137931034482758, 0.9572649572649573, 0.9473684210526315, 0.9224137931034484, 0
 .9724770642201835, 0.9363636363636363, 0.504424778761062, 0.8623853211009176, 0.9811320754716981, 0.9729729729729729, 0.926605504587156, 1.0, 0.9827586206896551, 0
 .4363636363636363634, 0.963302752293578, 0.9528301886792452, 0.894736842105263, 0.8849557522123894, 0.8318584070796461, 0.8928571428571429, 0.5086206896551724, 0
 .8376068376068376, 0.8166666666666667, 0.8403361344537815, 0.9495798319327732, 0.8303571428571429, 0.9130434782608695, 0.6068376068376069, 0.925, 0.8151260504201681,
 0.9401709401709403, 0.959016393442623, 0.9375]
System performence accuracy is: 0.8778833792392597
********************************************************
```

Figure 8: Calculate the accuracy ratio

After calculating the correct gaze ratio for each time interval, the average accuracy was computed by taking the mean of these values. **The average accuracy was determined to be 87.79%.**

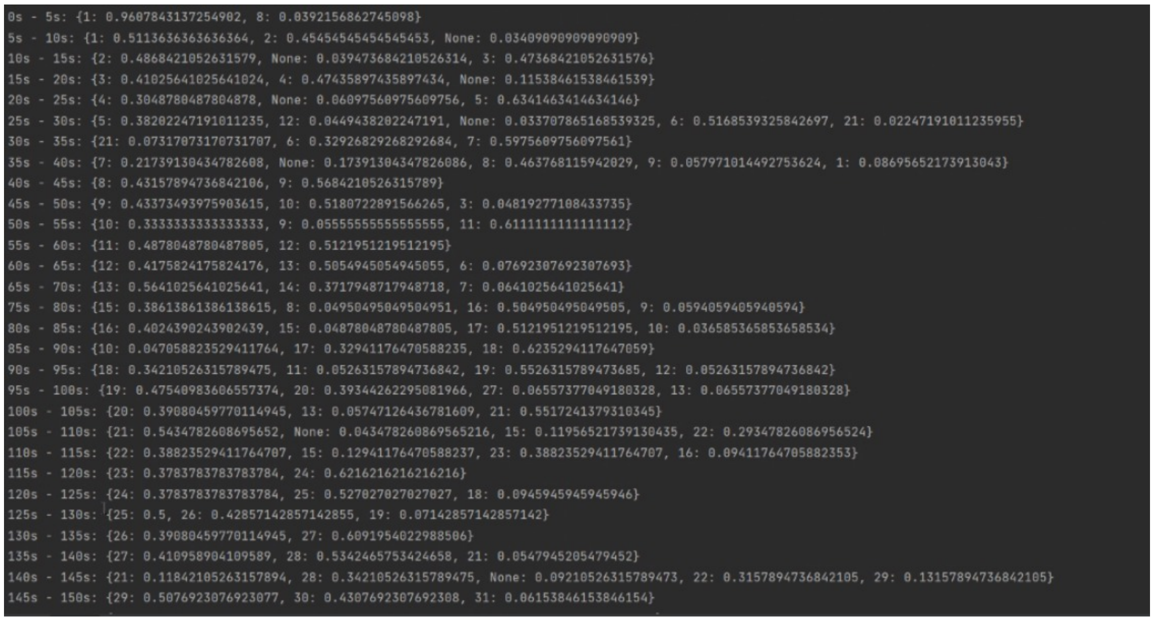
5 The Impact of Fixation Algorithms on Accuracy

We know that fixation algorithms improve accuracy. However, I couldn't find a study on the specific rates at which they enhance accuracy. Therefore, I aimed to conduct this study.

During my tests, I employed the I-DT algorithm, one of the most well-known dispersion-based fixation algorithms. The aim was to evaluate its impact on the accuracy of the eye-tracking system.

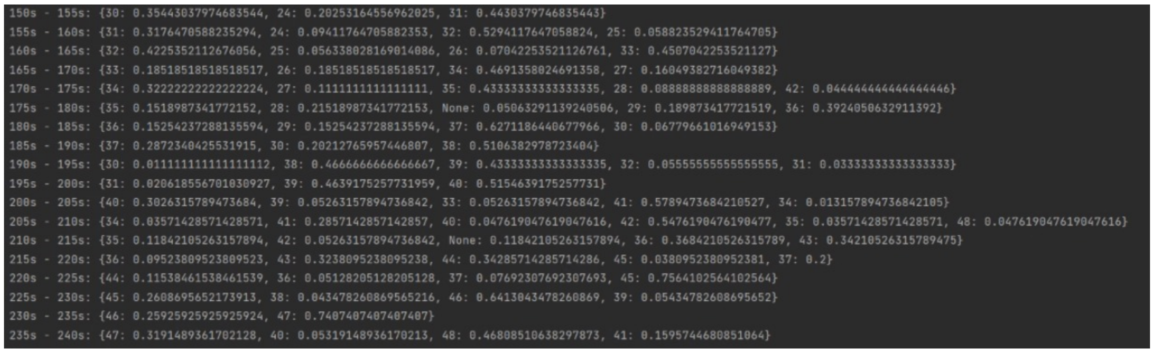
For the I-DT algorithm, I utilized the raw eye-tracking data from Kerem Keptiğ, which initially had an accuracy of 87.73%. Then, I calculated the corresponding cells for the resulting fixation points. By analyzing the time intervals, I determined which cells each fixation point belonged to. I calculated its corresponding cell and, based on the results, **the I-DT algorithm improved the accuracy of the data from 87.73% to 90.25%."**

5.1 Test With Fixation Algorithm



```
0s - 5s: {1: 0.9607843137254902, 8: 0.0392156862745098}
5s - 10s: {1: 0.5113636363636364, 2: 0.4545454545454453, None: 0.03409090909090909}
10s - 15s: {2: 0.4868421052631579, None: 0.039473684210526314, 3: 0.47368421052631576}
15s - 20s: {3: 0.41025641025641024, 4: 0.47435897435897434, None: 0.11538461538461539}
20s - 25s: {4: 0.3048780487804878, None: 0.06097560975609756, 5: 0.634146341463414 6}
25s - 30s: {5: 0.38202247191011235, 12: 0.0449438202247191, None: 0.033707865168539325, 6: 0.5168539325842697, 21: 0.02247191011235955}
30s - 35s: {21: 0.07317073170731707, 6: 0.32926829268292684, 7: 0.5975609756097561}
35s - 40s: {7: 0.21739130434782608, None: 0.17391304347826086, 8: 0.463768115942029, 9: 0.057971014492753624, 1: 0.08695652173913043}
40s - 45s: {8: 0.43157894736842106, 9: 0.5684210526315789}
45s - 50s: {9: 0.43373493975903615, 10: 0.5180722891566265, 3: 0.04819277108433735}
50s - 55s: {10: 0.3333333333333333, 9: 0.05555555555555555, 11: 0.6111111111111112}
55s - 60s: {11: 0.4878048780487805, 12: 0.5121951219512195}
60s - 65s: {12: 0.4175824175824176, 13: 0.5054945054945055, 6: 0.07692307692307693}
65s - 70s: {13: 0.5641025641025641, 14: 0.3717948717948718, 7: 0.0641025641025641}
75s - 80s: {15: 0.38613861386138615, 8: 0.04950495049504951, 16: 0.504950495049505, 9: 0.0594059405940594}
80s - 85s: {16: 0.4024390243902439, 15: 0.04878048780487805, 17: 0.5121951219512195, 10: 0.036585365853658534}
85s - 90s: {10: 0.047058823529411764, 17: 0.32941176470588235, 18: 0.6235294117647059}
90s - 95s: {18: 0.34210526315789475, 11: 0.05263157894736842, 19: 0.5526315789473685, 12: 0.05263157894736842}
95s - 100s: {19: 0.47540983606557374, 20: 0.39344262295081966, 27: 0.06557377049180328, 13: 0.06557377049180328}
100s - 105s: {20: 0.39080459770114945, 13: 0.05747126436781609, 21: 0.5517241379310345}
105s - 110s: {21: 0.5434782608695652, None: 0.043478260869565216, 15: 0.11956521739130435, 22: 0.29347826086956524}
110s - 115s: {22: 0.38823529411764707, 15: 0.12941176470588237, 23: 0.38823529411764707, 16: 0.09411764705882353}
115s - 120s: {23: 0.3783783783783784, 24: 0.6216216216216216}
120s - 125s: {24: 0.3783783783783784, 25: 0.527027027027027, 18: 0.0945945945945946}
125s - 130s: {25: 0.5, 26: 0.42857142857142855, 19: 0.07142857142857142}
130s - 135s: {26: 0.39080459770114945, 27: 0.6091954022988506}
135s - 140s: {27: 0.410958904109589, 28: 0.5342465753424658, 21: 0.0547945205479452}
140s - 145s: {21: 0.11842105263157894, 28: 0.34210526315789475, None: 0.09210526315789473, 22: 0.3157894736842105, 29: 0.13157894736842105}
145s - 150s: {29: 0.5076923076923077, 30: 0.4307692307692308, 31: 0.06153846153846154}
```

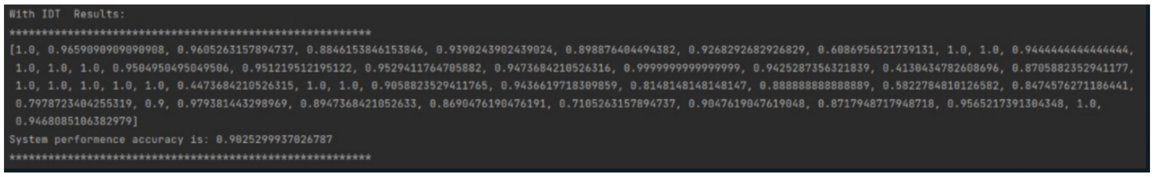
Figure 9: Eye gaze rates with I-DT for each time period-PART1



```
150s - 155s: {30: 0.35443037974683544, 24: 0.20253164556962025, 31: 0.4430379746835443}
155s - 160s: {31: 0.3176470588235294, 24: 0.09411764705882353, 32: 0.5294117647058824, 25: 0.058823529411764705}
160s - 165s: {32: 0.4225352112676056, 25: 0.056338028169014086, 26: 0.07042253521126761, 33: 0.4507042253521127}
165s - 170s: {33: 0.18518518518518517, 26: 0.18518518518518517, 34: 0.4691358024691358, 27: 0.16049382716049382}
170s - 175s: {34: 0.32222222222222224, 27: 0.1111111111111111, 35: 0.43333333333333335, 28: 0.08888888888888889, 42: 0.044444444444444446}
175s - 180s: {35: 0.1518987341772152, 28: 0.21518987341772153, None: 0.05063291139240506, 29: 0.189873417721519, 36: 0.3924050632911392}
180s - 185s: {36: 0.15254237288135594, 29: 0.15254237288135594, 37: 0.6271186440677966, 30: 0.06779661016949153}
185s - 190s: {37: 0.2872340425531915, 30: 0.20212765957446807, 38: 0.5106382978723404}
190s - 195s: {30: 0.011111111111111112, 38: 0.4666666666666667, 39: 0.43333333333333335, 32: 0.05555555555555555, 31: 0.03333333333333333}
195s - 200s: {31: 0.020618556701030927, 39: 0.4639175257731959, 40: 0.5154639175257731}
200s - 205s: {40: 0.3026315789473684, 39: 0.05263157894736842, 33: 0.05263157894736842, 41: 0.5789473684210527, 34: 0.013157894736842105}
205s - 210s: {34: 0.03571428571428571, 41: 0.2857142857142857, 40: 0.047619047619047616, 42: 0.5476190476190477, 35: 0.03571428571428571, 48: 0.047619047619047616}
210s - 215s: {35: 0.11842105263157894, 42: 0.05263157894736842, None: 0.11842105263157894, 36: 0.3684210526315789, 43: 0.34210526315789475}
215s - 220s: {36: 0.09523809523809523, 43: 0.3238095238095238, 44: 0.34285714285714286, 45: 0.0380952380952381, 37: 0.2}
220s - 225s: {44: 0.11538461538461539, 36: 0.05128205128205128, 37: 0.07692307692307693, 45: 0.7564102564102564}
225s - 230s: {45: 0.2608695652173913, 38: 0.043478260869565216, 46: 0.6413043478260869, 39: 0.05434782608695652}
230s - 235s: {46: 0.25925925925925924, 47: 0.7407407407407407}
235s - 240s: {47: 0.3191489361702128, 40: 0.05319148936170213, 48: 0.46808510638297873, 41: 0.1595744680851064}
```

Figure 10: Eye gaze rates with I-DT for each time period-PART2

In these results, we can observe that fixation algorithms have eliminated some points with very low gaze rates (saccade points). To illustrate this more clearly, I will present the corresponding graphs.



```
With IDT  Results:
*********************************************************
[1.0, 0.9659090909090908, 0.9605263157894737, 0.8846153846153846, 0.9390243902439024, 0.898876404494382, 0.9268292682926829, 0.6086956521739131, 1.0, 1.0, 0.9444444444444444,
 1.0, 1.0, 1.0, 0.9504950495049506, 0.951219512195122, 0.9529411764705882, 0.9473684210526316, 0.9999999999999999, 0.9425287356321839, 0.4130434782608696, 0.8705882352941177,
 1.0, 1.0, 1.0, 1.0, 1.0, 0.4473684210526315, 1.0, 1.0, 0.9058823529411765, 0.9436619718309859, 0.8148148148148147, 0.8888888888888889, 0.5822784810126582, 0.8474576271186441,
 0.7978723404255319, 0.9, 0.979381443298969, 0.8947368421052633, 0.8690476190476191, 0.7105263157894737, 0.9047619047619048, 0.8717948717948718, 0.9565217391304348, 1.0,
 0.9468085106382979]
System performence accuracy is: 0.9025299937026787
*********************************************************
```

Figure 11: Calculate the accuracy ratio with I-DT

5.2 Comparing Test Results

To visually demonstrate how fixation algorithms improve accuracy, I created pie charts comparing data with applied fixation algorithms (90.25% accuracy) to data without fixation algorithms (87.73% accuracy) **for each time interval from 0s to 240s.**

The charts display the rate of each cell in its respective segment. **If there is no cell number written above a rate, it means that point is not present on the screen and will be considered as a NULL point.**

Since there are a total of 48 time intervals, instead of showing all time intervals, specific rates are displayed. The graphs for all time intervals will be uploaded to GitHub along with the code.

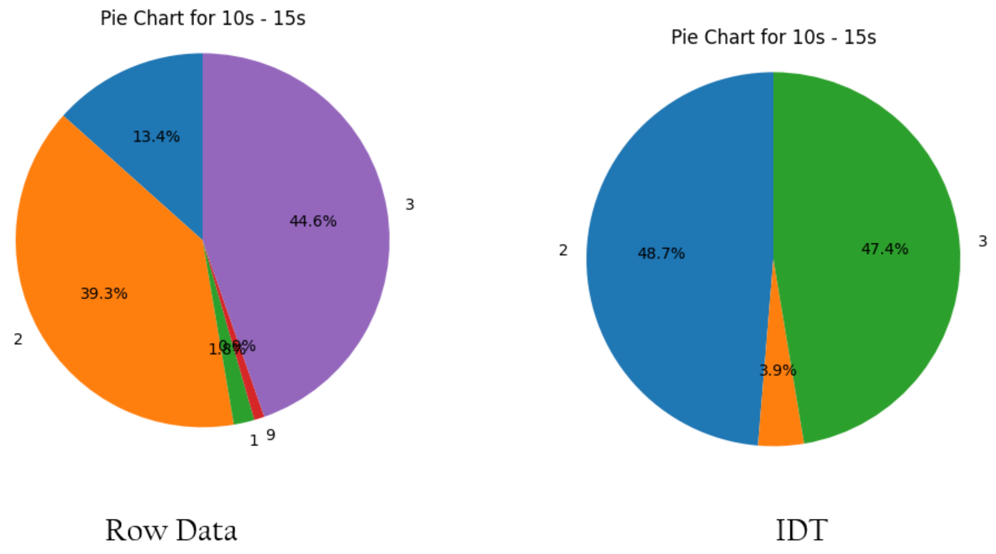
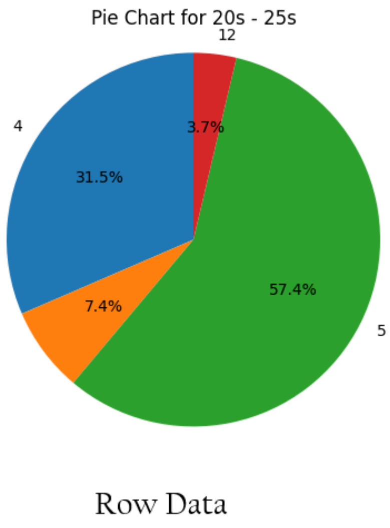


Figure 12: 10s-15s

It is necessary to focus on the 3nd cell and its neighboring cells between 10 seconds and 15 seconds. I-DT identified approximately 14% saccade points and increased accuracy by 9% in this time period.



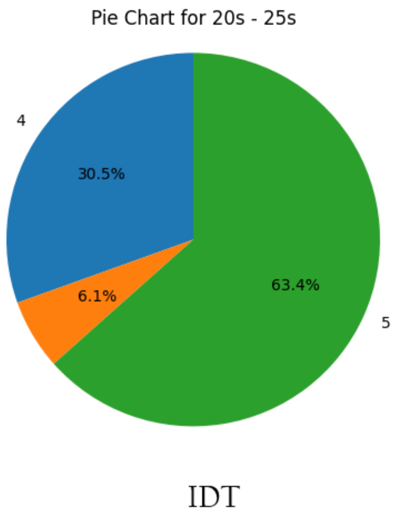


Figure 13: 20s-25s

It is necessary to focus on the 5nd cell and its neighboring cells between 20 seconds and 25 seconds. In this time interval, I-DT has classified the gaze belonging to the 12th cell as a saccade with a rate of 3.7%. Additionally, the rate for the NULL cell has decreased from 7.4 to 6.1.

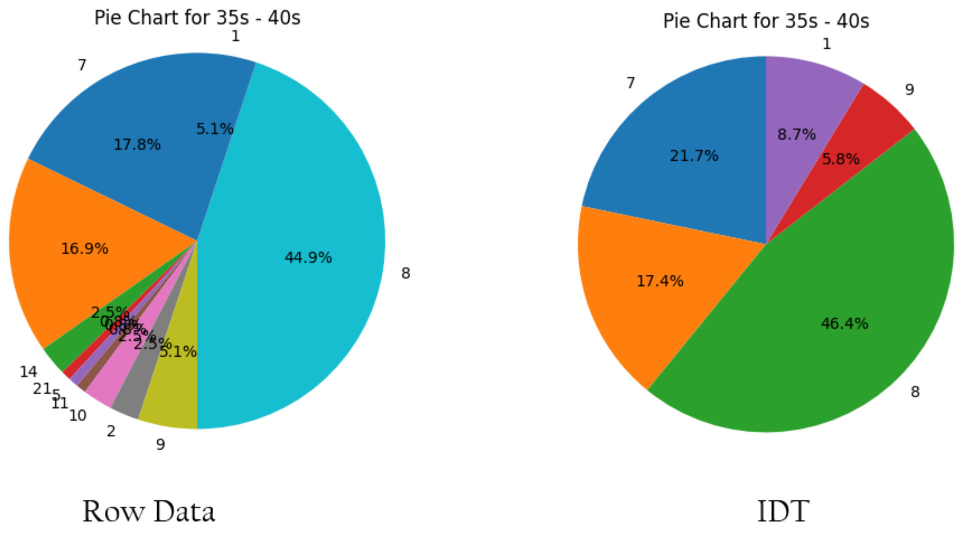


Figure 14: 35s-40s

It is necessary to focus on the 8nd cell and its neighboring cells between 35 seconds and 40 seconds. During this time interval, I-DT has eliminated numerous saccade points. In this time interval **without I-DT, a total of 11 points are observed, while with I-DT, saccade points are eliminated, reducing the total gaze points to 5.**

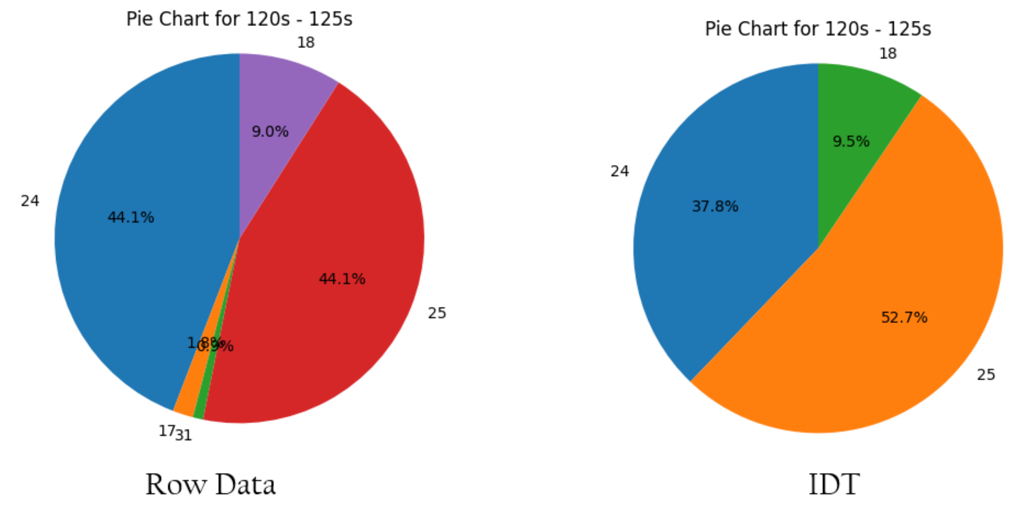
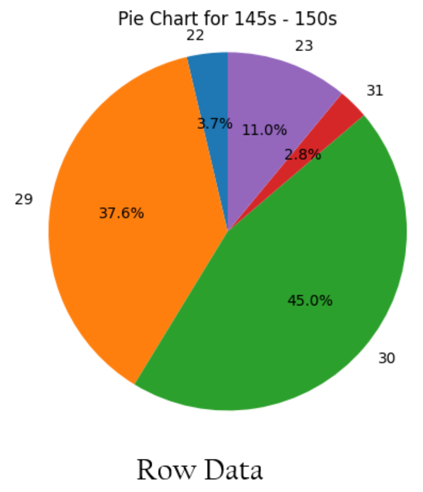


Figure 15: 120s-125s

It is necessary to focus on the 25nd cell andits neighboring cells between 120 seconds and 125 seconds. I-DT has identified and eliminated cells 17 and 31 as saccade points during this time interval.



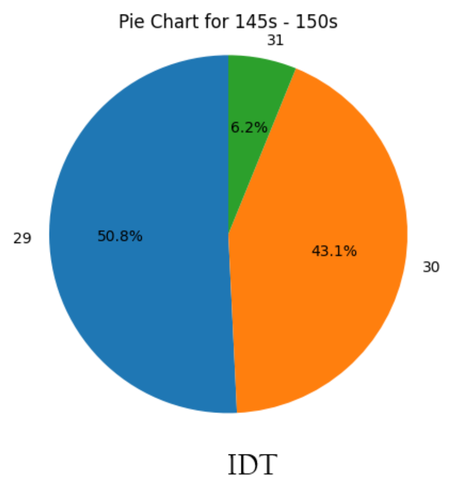


Figure 16: 140s-150s

Between the 140th and 150th seconds, it is expected to focus on cell 30 and its neighbors (cells 23, 29, 37, and 31). Although I-DT has reduced the viewing ratio of cell 30 from 45% to 43.1%, the accuracy of cell 30 and its neighboring cells is considered for calculating the system's accuracy. Since cell 22, found in the raw data, is eliminated, the success rate in this time interval has increased from 96.3% to 100%.

This interval is one of the most illustrative examples of why I include neighboring cells in the accuracy calculation. As we know, one of the biggest problems with eye trackers is the constant tremor of the cursor on the screen. This can sometimes make it appear as if we have shifted our focus slightly from where we are actually looking. Due to this effect and the small size of each cell I created, neighboring cells have been included in the accuracy calculation.

6 Future Work

- I have decided to transform the system I developed into a web-based platform, allowing us to test the accuracy of any eye-tracking device online. The aim is to create a web-based website that performs accuracy calculations for eye-tracking devices.

- By making the system web-based, we enable users to easily access and test the accuracy of their eye-tracking devices online. This platform will serve as a website where users can conveniently perform accuracy calculations for their eye-tracking devices, making the testing process more accessible and user-friendly.

- We've witnessed how fixation algorithms improved the accuracy. In the current initial test, achieving an accuracy increase from 87.3% to 90.25% is a great result. For now, the rate of improvement is calculated as 3.32%. However, with future tests, an average value will be determined.

- I want to introduce a new perspective to calculate the overall accuracy of the EASIEST project. Instead of using the accuracy of WebGazer, I believe we should calculate the accuracy of fixation points generated by the fixation algorithm and combine this result with the accuracy of the ML model.

  For instance, if we plan to use an ML model with 90% accuracy in the upcoming period, we can estimate the system accuracy for the tests in this report as $90.25\% \times 90\% = 81.225\%$.