Analyze and Visualize Eye-Tracking Data

Aymen Sekhri
aymensek@.uef.fi

Emmanuel Kwabena Frimpong
emmafrim@uef.fi

Bolaji Mubarak Ayeyemi
mayeyemi@uef.fi

Aleksi Hirvonen
alekshi@uef.fi

Matias Hirvonen
matiashi@uef.fi

Tedros Tesfay Andemichael
tedandem@uef.fi

ABSTRACT

Fixation identification, which involves isolating and identifying fixations and saccades in eye-tracking protocols, is an important aspect of eye-movement data processing that can have a big impact on higher-level analyses. However, fixation identification techniques are frequently discussed informally and rarely compared in any meaningful way. With two state-of-the-art algorithms, we will implement fixation detection and analysis in this work. The velocity threshold fixation algorithm is the first algorithm, and it identifies fixation based on a threshold value. For eye movement detection, the second approach is U'n' Eye, a deep neural network algorithm. The goal of this project is to analyze and visualize eye-tracking data from an eye gaze dataset that has been provided. The data was collected in a scenario in which individuals were shown photos and asked whether or not they recognized them. The results of the two-fixation detection approach are contrasted and visualized in this paper.

1. Introduction

Over the last decade, eye tracking has become increasingly popular as a doorway into observers' visual and cognitive processes. Eye-tracking has been used to examine behavior in areas including image scanning [1], arithmetic[2], driving[3], analogies, and reading. Fixations (pauses over informative regions of interest) and saccades are commonly used by researchers in these and other disciplines to describe eye movements (rapid movements between fixations). Natural settings, simple artificial stimuli, webpages, user interfaces, and, increasingly, information visualizations have all been studied using eye movement analysis. Eye-tracking has been widely utilized in human-computer interaction (HCI) to assess system usability and investigate the related subject of interface design [4], [5]. Duchowski [4] presents an overview of many eye-tracking applications in fields ranging from industrial engineering to marketing.

Fixation or gaze durations, saccadic velocities, saccadic amplitudes, and various transition-based parameters between fixations and/or regions of interest are common analysis metrics.

Fixation identification (or simply identification) is the process of translating raw eye-movement data points into fixation positions (and implicitly the saccades between them) on the visual presentation required for fixation and saccade research. Fixation identification considerably decreases the size and complexity of the eye-movement procedure by removing raw saccade data points and combining raw fixation points into a single representative tuple. This lowering is beneficial for at least two reasons. To begin with, a saccade may achieve little or no visual processing [6], therefore the actual pathways used during saccades are frequently irrelevant for many research applications. Second, during fixations, tiny eye movements such as tremors, drifts, and flicks [7] are usually neglected in the higher-level analysis. As a result, fixation identification provides a simple way to reduce the complexity of eye-tracking data while keeping the most important qualities for studying cognitive and visual processing behavior.

A statistically-based description of observed eye movements is known as fixation identification. While it is commonly understood that fixations involve visual and cognitive processing [8], it is less clear when fixations begin and end. As a result, the identification problem remains a subjective process, despite the precision and flexibility of identification algorithms. As a result, comparing the created fixations to the subjective impressions of an observer is an excellent method of validating these algorithms.

In the eye-movement literature, notably in the research on the relationship between eye movements and higher-level cognition, fixation identification and its various implementations have gotten a lot of attention. Identification, on the other hand, is frequently a necessary aspect of eye-movement data processing, with significant consequences for later studies. A fixation is a collection of points with a distance between them less than a predetermined value and a temporal interval more than a predetermined period. It's been thought of as a collection of places where a subject's gaze has been concentrated[9]. In the literature, the minimum time and dispersion distance to define a fixation have been hotly contested. It has long been assumed that the minimum fixation time is more than 0.1 seconds [10]. The work the subject is doing determines the subject's minimum time. For tasks such as reading and visual search, the minimum fixation period is 0.225 seconds and 0.275 seconds, respectively. The average fixation duration for eye-hand coordination tasks was found to be 0.4 seconds [9]. In conclusion, the mean fixation period has been calculated to be 0.15 to 0.65 seconds [11]. Fixation dispersion angles have yet to be determined, but they are usually adjusted to less than 2° [9]. Apart from the widespread under-specification of identification algorithms, there has been little effort put into assessing and comparing alternative possibilities. In many circumstances, academics and practitioners lack sufficient information to determine which algorithm to apply in a given situation. This issue, expected, leads to the haphazard use of multiple methods, making it difficult to compare results acquired using various identification strategies.

In this paper, we used velocity-threshold identification (I-VT) and U'n' Eye, deep neural network algorithms to identify fixation on the provided dataset. The result of both methods is compared, and we provide a review of metrics that can be used for quantitative comparison. Moreover, we discuss and visualize ways in which different metrics can be used to present the algorithm results and determine the effectiveness of the algorithms.

2. Method

2.1 Dataset

The data was collected in a circumstance where individuals have presented photos and asked to identify whether they recognized them. The dataset is made up of rows of raw gaze positions x,y recorded in time (one row per session and person). In the dataset, subjects have several rows/samples. The dataset is available in two formats: CSV (a text file with one row for each sample) and Matlab (a spreadsheet with one row for each sample). The following features are included in it:

$\boldsymbol{sid} - subject\ identifier\ (sXX)$

$\boldsymbol{known} - decision\ of\ the\ subject\ concerning\ the\ observed\ image(true\ =\ recognized\ imaged, false\ =\ did\ not\ recognize).$

$\boldsymbol{x_i} - the\ i\ th\ alue\ of\ the\ recorded\ horizontal\ eye\ gaze\ point\ (raw\ data)$

$\boldsymbol{y_i} - the\ i\ th\ value\ of\ the\ recorded\ vertical\ eye\ gaze\ point\ (raw\ data)$

The values for x and y are 0,0 for the screen's center point, positive for the screen's right and lower side, and negative for the screen's left and upper side. Every row has a different number of values! (Because participants' reactions may take varying amounts of time.) Assume a 1000Hz sampling rate.

2.2 Velocity-Threshold Identification (I-VT)

Velocity-threshold fixation identification is the simplest of the identification processes to comprehend and apply (I-VT). Based on their point-to-point velocities, the I-VT method separates fixation and saccade locations[12]. Fixation velocity algorithms combine a duration criterion and an eye velocity-based stillness condition. Due to micro-movements in the eye and noise from the eye-tracker, the eye velocity is rarely at absolute zero. As a result, users of this technology must decide on a fixation velocity upper limit[13]. Fixation length is the average amount of time between fixations. Typically, the fixation duration is between 150 and 300 milliseconds [14]. The velocity profiles of saccadic eye movements show low velocities for fixations (i.e., 100 degrees per second) and high velocities for saccades (i.e., >300 degrees per second)[15]. I- The I-VT approach's pseudocode is shown in Table 1. I-VT requires one parameter, the velocity threshold. The point-to-point velocity threshold can be approximated using an acceptable angular velocity threshold [6] if angular velocities can be determined (i.e., the distance between the eye and the visual stimuli is known).

Table 1: Pseudocode for the I-VT algorithm.

```
fixation_velocity(protocol,velocity_threshold,clustering_distance):
    Calculate point-to-point velocities for each point in the protocol.

    Label each point below velocity_threshold as a fixation point, otherwise as a saccade point.

    Collapse consecutive fixation points into fixation groups, removing saccade points.

    Map each fixation group to fixation at the centroid of its points.

    Collapse fixation points that the distance between them is below clustering_threshold, and combine them to fixation at the center of centroids.

    Return fixations, duration of each fixation
```

I-VT requires one parameter, the velocity threshold. The point-to-point velocity threshold can be approximated using an acceptable angular velocity threshold [6] if angular velocities can be determined (i.e., the distance between the eye and the visual stimuli is known). For example, Sen and Megaw [16] used a 20-degree-per-second barrier. When only point-to-point velocities are available, an appropriate velocity threshold value must be established based on data collection factors (e.g., sample frequency) and exploratory data analysis. We have added one parameter to the basic I-Vt algorithm, ie. Clustering distance. The modification was made to the velocity algorithm to improve its results. The modification is combining any two neighboring fixations points if the distance between them is less than a certain threshold. The reason for this modification is that the original velocity algorithm is sensitive to the velocity, so it gives many fixations (with small durations) in very close points, but clustering these close fixations will result in one fixation with a higher duration.

2.3 U'n'Eye deep neural network

U'n'Eye is a deep neural network model that can be used for the detection of saccades and other eye movements. It was constructed with a convolutional neural network (CNN) to automatically detect saccades at human-level performance accuracy. The application outperforms the present state of the art, according to prominent performance metrics, and will aid researchers in better understanding the neurophysiological [17]. In addition to saccades and microsaccades, the network was processed that governs saccade production and visual processing able to predict other forms of eye movements, such as blinks and PSOs. U'n'Eye performed well when trained on a single type of data with only one coder's labels and when trained on two datasets with two coders' labels, according to the study. Prediction of eye movement can be possible with the already trained model, but it is recommended to train the network on your dataset to achieve optimal performance. U'n'Eye is free and accessible to the public, with a user-friendly interface and a web API that allows users to upload data and get categorization results. No parameter changes are necessary for training (e.g., learning rate, etc.) because the default parameters have been found to work well across datasets. An experimenter only requires a few hundred seconds of tagged data to train the network.

Predict saccades in recordings with pre-trained weights. Arrange the data into a matrix of *samples*timebins* or input them as a vector of length *timebins*.

$$Prediction, Probability = model.pred(\mathrm{X}, \mathrm{Y})$$

Input parameters:

X,Y : horizontal and vertical eye positions, {array-like, shape: {(nsamples, nbins) or (nbins)}

Output:

Prediction: eye movement class prediction for each time bin, the same shape as input

Probability: softmax probability output of network, shape: {n_samples,classes,time) or (classes,time)}

The weight used for the model is "weights_dataset3" with the sampling frequency of 1kHZ

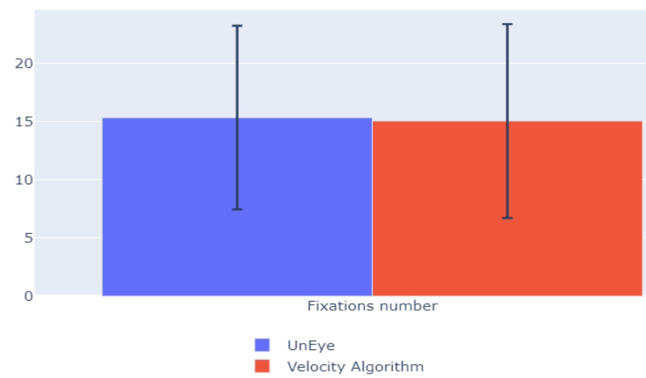
3. Results and Discussion

The results of velocity threshold identification and the U 'n'Eye deep learning model with various metrics that can be utilized for collective eye movement analysis over the dataset are shown and discussed in this part.

Mean fixation duration measures the average fixation time per subject. Considering table 2, the mean fixation duration for all subjects when the person identifies image ranges from 0.104s to 0.185s, and the mean fixation duration for all subjects when a person could not able identify image ranges from 0.101s to 0.205s Moreover, the maximum overall mean fixation duration in all subjects is 0.195s. According to different research results, the fixation duration should be between 150 and 300 milliseconds.

Therefore, subjects whose mean fixation threshold is not in the range of 150 and 300 milliseconds can be considered as involuntary (the eyes move there without a conscious decision). The standard deviation of fixation points for all subjects is presented in the table that defines how far do members of a group stray from the mean value of the group. Saccadic amplitude is the distance traveled by the eye between two fixation locations. The overall mean saccade amplitude for the subjects ranges from 1768.767 to 6182.264 and their overall standard deviation ranges from 842.032 to 3503.537. From this, you can comprehend there is a considerable difference between the amplitude saccade of the subjects.

Figure 1: comparison by number of fixations b/n algorithms



The above figure demonstrates the number of fixations identified by velocity threshold identification and the U 'n'Eye deep learning model. Both algorithms capture almost the same number of fixations.

Table 2: metrics used in velocity threshold identification algorithm

subject	MFD _true	MFD_SD _true	MFD _false	MFD_SD _false	MSA _true	MSA_SD _true	MSA _false	MSA_SD _false	MFD _overall	MFD_overall _SD	MSA_overall	MSA_overall_ SD
S2	0.172	0.053	0.144	0.061	5435.837	3221.201	6621.338	3449.283	0.154	0.060	6182.264	3414.938
S4	0.157	0.049	0.167	0.052	7027.879	5607.174	4708.222	1520.309	0.164	0.051	5411.149	3503.537
S8	0.118	0.024	0.101	0.015	2889.662	749.630	3220.273	1130.792	0.111	0.022	3034.305	949.958
S10	0.104	0.012	0.129	0.030	4977.361	993.046	4571.209	1512.967	0.126	0.030	4605.055	1480.910
S16	0.185	0.011	0.199	0.035	4984.970	1972.415	2710.702	1033.678	0.195	0.032	3235.533	1624.150
S22	0.131	0.032	0.142	0.044	2830.419	741.837	1638.873	622.990	0.139	0.042	1956.619	842.032
S23	0.149	0.015	0.205	0.055	2024.028	936.453	1699.151	1043.371	0.193	0.055	1768.767	1030.065
S30	0.121	0.043	0.115	0.020	3590.710	703.209	5107.893	2562.173	0.116	0.025	4863.186	2428.384

Figure 2: comparison by mean fixation duration b/n algorithms

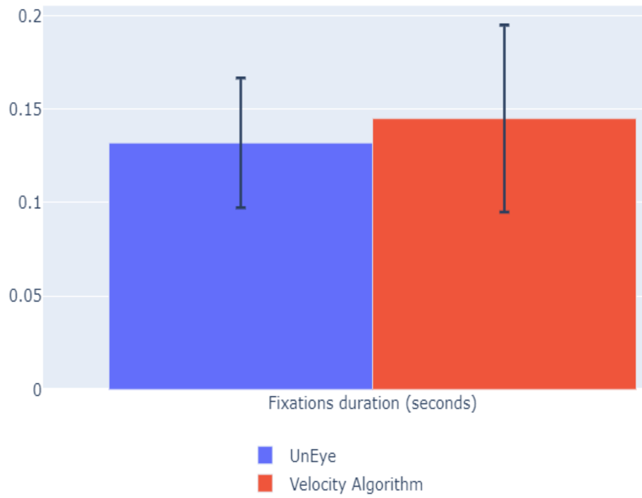
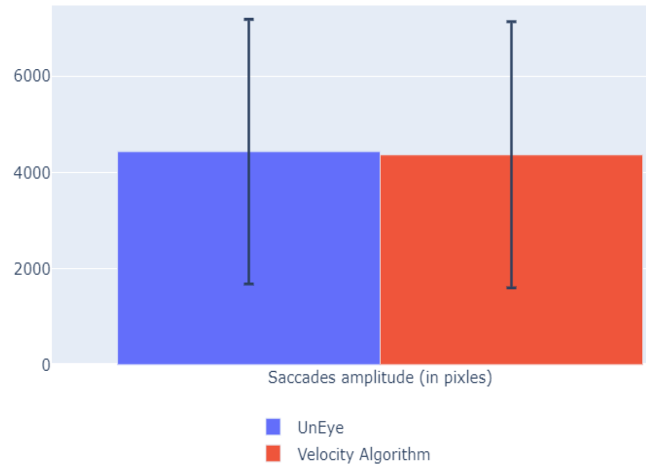


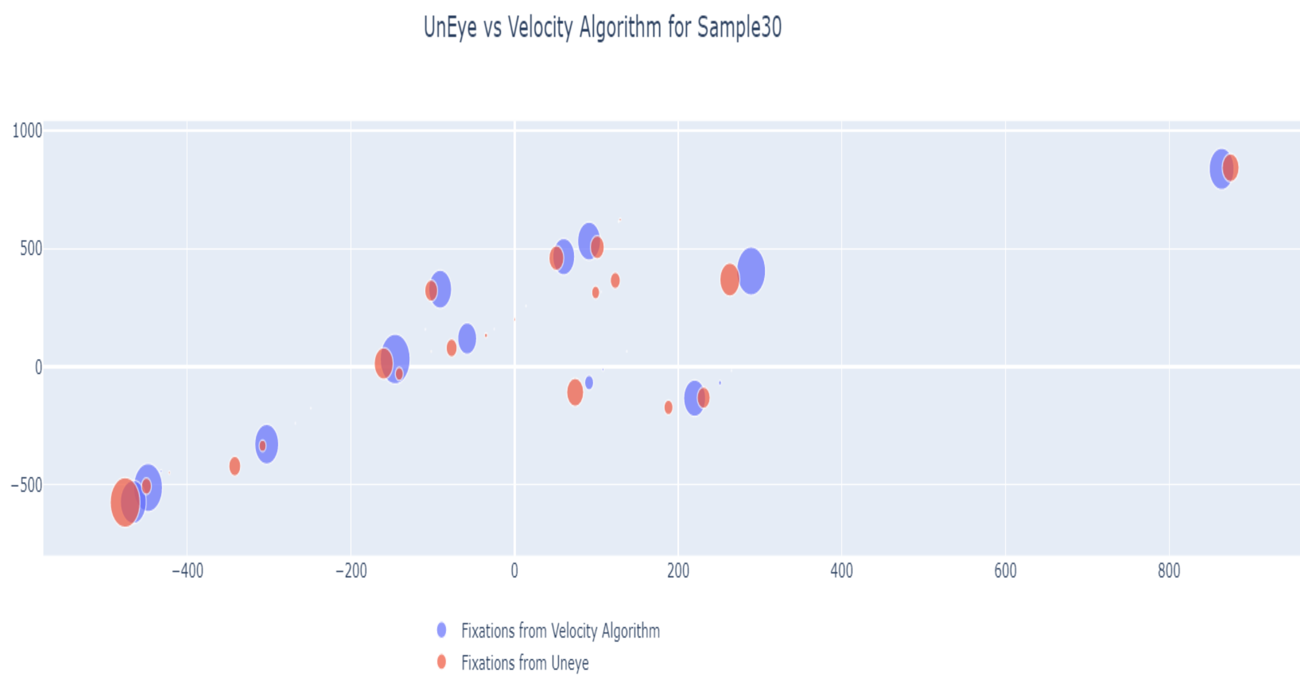
Figure 2 presents the mean fixation duration returned by velocity threshold identification and the U 'n'Eye deep learning model. The mean fixation duration of the velocity threshold identification algorithm is a little bit higher than the deep learning model.

Figure 4: comparison by saccades amplitude b/n algorithms



The above figure demonstrates the saccade amplitude of velocity threshold identification and the U 'n'Eye deep learning model. Almost the amplitude saccade of both algorithms is the same.

Figure 3: visualization of fixation identified by both methods in a two-dimensional plane



The above figure demonstrates the position of the centroid of fixation points in a two-dimensional plane. As in the above figure, most of the fixation points identified by both algorithms are positioned almost in the same location. This implies that both methods work well in identifying the centroid of fixations.

Figure 5: comparison by mean fixation duration b/n recognized and non-recognized images

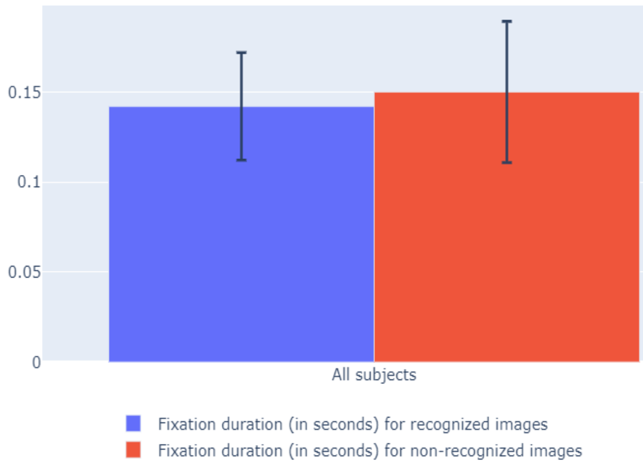
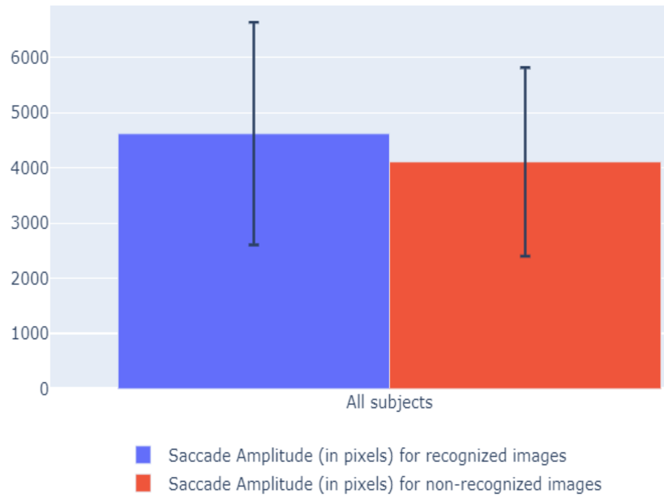


Figure 6: comparison by saccade amplitude b/n recognized and non-recognized images



Figures 5 and 6 present the mean fixation duration and saccade amplitude for recognized images and non-recognized images respectively. The mean fixation duration for both recognized images and non-recognized images is almost close to 1ms.

Figures 7 and 8 depict about mean fixation duration and saccade amplitude for recognized images and non-recognized images respectively of all subjects. Subject id-23 has the highest fixation duration compared to others.

Figure 7: visualization of mean fixation duration for recognized image and non-recognized images for every subject

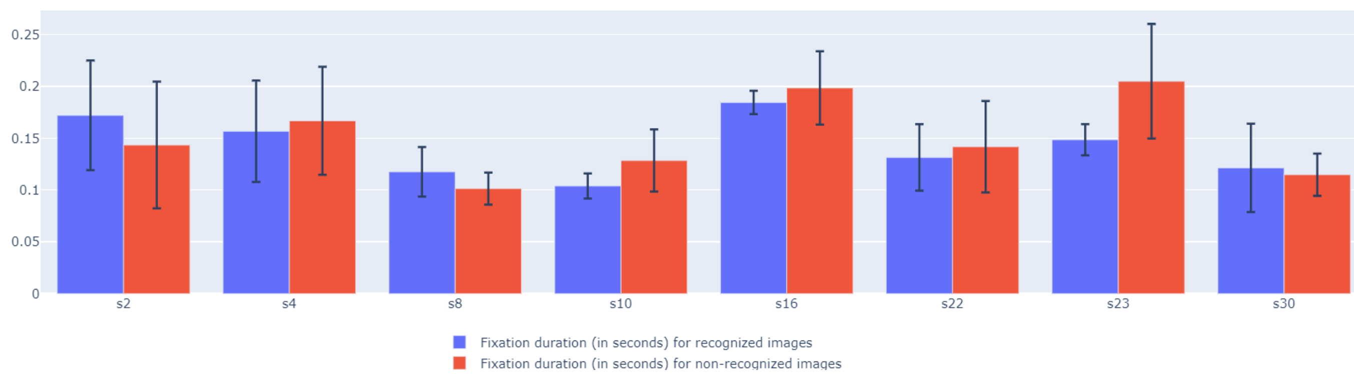
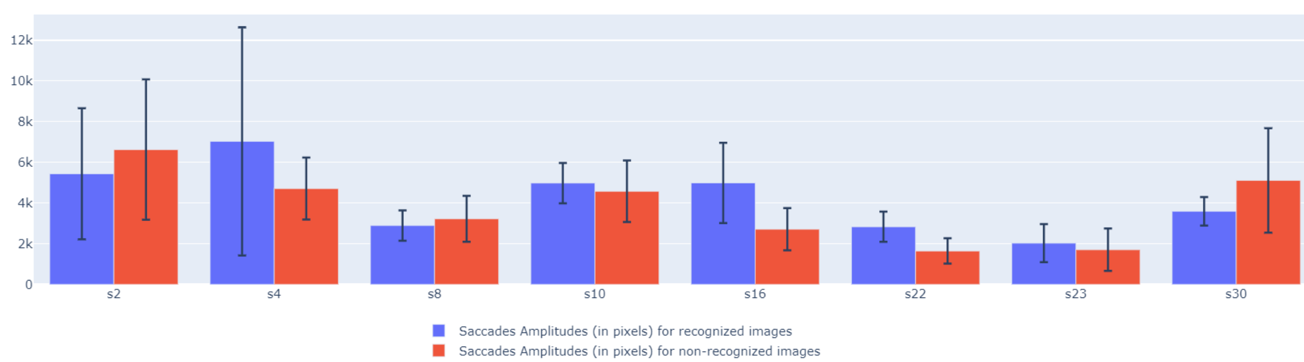


Figure 8:visualization of saccade amplitude for recognized image and non-recognized images for every subject



4. Conclusion

The results of eye movement analysis could help researchers better understand human cognitive and perceptual processes, as well as visual content design principles. Automated design predictions will be made in the future. In the conclusion, in this work, we use velocity threshold identification and U'n'Eye deep neural network model to capture fixation from the recorded eye-gaze dataset. We present the results of the algorithms with different metrics in a tabular and visual format. Moreover, both fixation identification algorithms are compared in terms of the number of fixations, mean fixation duration, and saccade amplitude. The Velocity-threshold fixation identification was calibrated with different threshold values to achieve better results. Velocity-threshold fixation identification has a higher mean fixation duration compared to the U'n'Eye deep neural network model and both algorithms capture almost the same number of fixations. The locations of centroids of the captured fixations points in both methods are placed closer to each other.

REFERENCES

[1] D. Noton and L. Stark, "Scanpaths in saccadic eye movements while viewing and recognizing patterns," *Vision Research*, vol. 11, no. 9, pp. 929-IN8, Sep. 1971, doi: 10.1016/0042-6989(71)90213-6.

[2] Suppes P, "Eye-movement models for arithmetic and reading performance.".

[3] M. F. Land and D. N. Lee, "Where we look when we steer," *Nature*, vol. 369, no. 6483, pp. 742–744, 1994, doi: 10.1038/369742a0.

[4] A. T. Duchowski, "A breadth-first survey of eye-tracking applications," *Behavior Research Methods, Instruments, & Computers*, vol. 34, no. 4, pp. 455–470, 2002, doi: 10.3758/BF03195475.

[5] J. H. Goldberg and X. P. Kotval, "Computer interface evaluation using eye movements: methods and constructs," *International Journal of Industrial Ergonomics*, vol. 24, no. 6, pp. 631–645, 1999, doi: https://doi.org/10.1016/S0169-8141(98)00068-7.

[6] A. F. Fuchs, "THE SACCADIC SYSTEM," 1971.

[7] M. ALPERN, "Type of movement," *New York: Academic Press*, pp. 63–151, 1962.

[8] P. A. C. Marcel Adam Just, *Using Eye Fixations to Study Reading Comprehension*. 1984.

[9] K. Rayner, "Eye movements in reading and information processing: 20 years of research".

[10] G. E. Manor BR, "Defining the temporal threshold for ocular fixation in free-viewing visuocognitive tasks".

[11] D. E. Irwin, "Memory for position and identity across eye movements.," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 18, no. 2, pp. 307–317, 1992, doi: 10.1037/0278-7393.18.2.307.

[12] C. J. Erkelens and I. M. L. C. Vogels, "The initial direction and landing position of saccades," in *Eye movement research : mechanisms, processes and applications*, J. M. Findlay, R. Walker, and R. W. Kentridge, Eds. Amsterdam: Elsevier, 1995, pp. 133–144. doi: 10.1016/S0926-907X(05)80012-1.

[13] M. N. R. A. R. D. H. J. J. van de W. Kenneth Holmqvist, *Eye Tracking A Comprehensive Guide to Methods and Measures*. 2011.

[14] B. A. Tom Tullis, *Behavioral and Physiological Metrics*. 2013.

[15] D. D. , & G. J. H. Salvucci, "Identifying fixations and saccades in eye-tracking protocols.," 2000.

[16] J. A. Hartigan, "Clustering algorithms," *New York: John Wiley & Sons.*, 1975.

[17] J. B. H. N. Z. M. H. P. B. Marie E. Bellet, "Human-level saccade detection performance using deep neural networks," *Neurophysiology*, 2018.