

ERASMUS UNIVERSITY ROTTERDAM

ERASMUS SCHOOL OF ECONOMICS

Master Thesis Business Analytics and Quantitative Marketing

Combining Supervised and Unsupervised Learning for Fixation Detection in Eye Tracking Data

Student name:

Babette de Leede

Student ID number:

508282

Supervisor:

Alfons, A. dr.

Second assessor:

Groenen, P.J.F. dr.

Date final version: April 10, 2024

The content of this thesis is the sole responsibility of the author and does not reflect the view of the supervisor, second assessor, Erasmus School of Economics or Erasmus University.

Abstract

Eye movement data yield insights into the underlying cognitive processes that occur when examining a visual stimulus. The aim of eye movement research is to establish which areas of the stimulus attract the most visual attention. This is done by classifying eye movements into fixations and saccades. For this purpose, a large number of fixation identification algorithms have been proposed. This paper is the first to use a hierarchical combination of supervised and unsupervised machine learning to detect fixations in eye tracking data. An unsupervised learning method first identifies the latent data structure and labels eye movements as fixations or saccades, after which a supervised learning algorithm utilises these identified data patterns for further classification. The four combinations of unsupervised and supervised learning methods used in this research result in similar eye movement statistics and sample classification, while the unsupervised techniques independently vary in both aspects. Hence, the proposed method is promising to identify fixations in eye tracking data in a general applicable way.

Contents

1	Introduction	3
2	Literature	6
2.1	Background	6
2.2	Machine Learning	7
3	Data	9
3.1	Data description	9
3.2	Cleaning	10
4	Methodology	11
4.1	Unsupervised Learning	12
4.1.1	Binocular-Individual Threshold algorithm	12
4.1.2	Eye Movements Metrics & Visualisations algorithm	14
4.2	Combining Supervised and Unsupervised Learning	16
4.2.1	Feature extraction	16
4.2.2	Random Forest	18
4.2.3	Convolutional Neural Network	21
4.3	Evaluation	23
5	Results	24
5.1	Unsupervised Learning	24
5.2	Combined Unsupervised-Supervised Learning	27
5.2.1	Random Forest	27
5.2.2	Convolutional Neural Network	33
5.2.3	Final predictions	38
6	Conclusion	40
A	Choice Tasks	51
B	Data Statistics	51
C	Comparison Fixation Centers Unsupervised Learning Methods	51
D	Random Forest	52
D.1	Grid search	52

D.2 Feature Importance	52
----------------------------------	----

1 Introduction

When people examine a visual stimulus, for example an advertisement, it is of interest to establish which areas of the stimulus attract the most visual attention. Visual attention is focused on the most informative areas of a stimulus (Loftus and Mackworth, 1978; Rayner, 1998). Eye movements reflect the underlying cognitive processes that occur when viewing a stimulus (Rayner, 1978), and therefore enable establishment of principles of human information processing (Radach et al., 2004). In other words, the underlying cognitive processes of visual perception rely on the acquisition and processing of visual information. Consequently, eye movement data yield insights into unobservable processes that are difficult to obtain otherwise. Eye movements are discontinuous when observing a visual stimulus, with distinct areas being fixated sequentially (Yarbus, 2013). The question of how one can determine which areas of a visual stimulus get the most visual attention, i.e. are the most informative, is addressed in eye movement research.

Eye movement data are collected by eye tracking devices. These devices measure where one is looking (point of gaze), sample the location that each eye is focused on, and track its movement multiple times per second. Typically in eye movement research, eye movements are classified into fixations and saccades. Fixations are defined as periods between eye movements when the eyes are relatively motionless and focused. During fixations, the eyes are aimed at a specific area of a visual stimulus. Saccades are fast eye movements between fixations, in which a viewer's eye is directed to a visual target (Rayner, 1998). Cognitive and visual information processing occur during fixations, whereas vision is essentially suppressed during a saccade (Rayner, 1998). The frequency of fixations is an indication of the level of informativeness, whereas the duration of fixations indicates the complexity and difficulty of visual display (Fitts et al., 2004).

The aim of eye movement research is to robustly classify eye movement events, such as fixations and saccades, from the stream of raw eye movement data points obtained from an eye tracker device. Since limited visual processing occurs during saccades, the the principal interest of eye movement research is to distinct fixations from saccades. Robust, efficient, and accurate identification of fixations provides valuable information on what areas of visual stimuli attract the most attention for various applications. In the medial research for instance, eye movement data have been used for detection of developmental disorders including dyslexia (Rello and Ballesteros, 2015) and autism (Vabalas et al., 2020), and for disease diagnosis such as Schizophrenia (O'Driscoll and Callahan, 2008), Parkinson (Stuart et al., 2016) and Alzheimer (Crawford et al., 2015). In the marketing domain (Wedel and Pieters, 2017), advertisements, designs, and customers' shopping behaviour are evaluated using eye tracking tools. Other common applications include human-computer interactions (Pan et al., 2004; Majaranta and Bulling, 2014; Wu et al.,

2019), entertainment (Pucihar and Coulton, 2015; Hartmann and Fox, 2021), and virtual reality (Smith and Neff, 2018; Clay et al., 2019).

To identify fixations in eye movement data, a large number of fixation identification algorithms have been developed. However, there is no standard method to detect fixations in eye movement data. In fact, the choice of algorithm may drastically affect the resulting classified fixations (Karsh and Breitenbach, 2021). In recent years, the quality of eye tracker devices has improved due to technological advancements, while various machine learning methods have successfully been proposed to identify events in eye movement data. Supervised learning methods such as Neural Networks (Yin et al., 2018), Random Forest (Zemblys et al., 2018), Support Vector Machine (Wu et al., 2010), Naive Bayes classifier (Bhattarai and Phothisonothai, 2019) show encouraging results in eye movement analysis as significant progress is made in the classification of events compared to state-of-the-art algorithms. However, training supervised deep learning models requires a large amount of labeled data. Obtaining labels for eye tracking data can be an exhaustive process. Furthermore, supervised learning tends to suffer from overfitting when data is noisy, high-dimensional and/or complex. Unsupervised classification algorithms have also been used to identify events in this context (Otero-Millan et al., 2014; Göbel and Martin, 2018; Fuhr and Kasneci, 2022). They do not need labeled input, but typically result in less accurate classification than supervised methods.

The purpose of this research is to address this problem of difficult-to-obtain labeled eye tracking data while supervised techniques generally show better accuracy in identifying events than unsupervised methods. Therefore, I propose a combination of supervised and unsupervised learning to classify fixations in eye movement data. Combined unsupervised-supervised machine learning has recently shown promising results in multiple research domains (Hashemzadeh and Azar, 2019; Kim et al., 2022; Mishra et al., 2022). This study first uses an unsupervised algorithm to label eye tracking data as either fixations or saccades by identifying the latent structure of the data. These labeled data are utilised for further classification using a supervised learning technique. This idea is visualised in Figure 1. The complementarity between the two techniques seems appealing in the context of eye tracking data, because the proposed method combines the superior performance of supervised machine learning with the ability of unsupervised learning to detect inherent data patterns. The following research question is investigated:

Is it beneficial to combine supervised and unsupervised learning to identify fixations in eye tracking data relative to the individual algorithms? Questions that arise and help answering the research question include:

- *What are the differences between the fixations identified by the proposed combined method*

and the unsupervised method (on its own)?

- *To what extent is the proposed method able to transfer learn from a previously seen task?*
- *How do different combinations of unsupervised and supervised methods affect the classified fixations?*

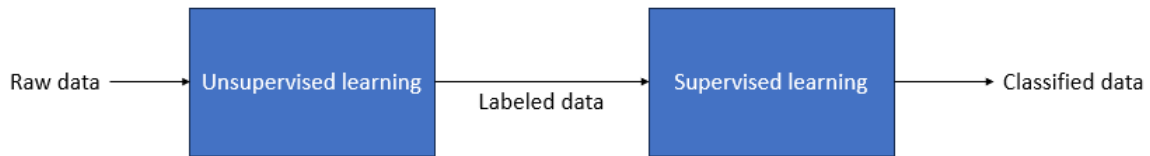


Figure 1: Visualisation of combining supervised and unsupervised learning

In this research, the velocity-based Binocular-Individual Threshold algorithm of Van der Lans et al. (2011) and the dispersion-based Eye Movements Metrics & Visualizations algorithm of Krassanakis et al. (2014) are used as unsupervised learning techniques. The supervised methods include Random Forest (Breiman, 2001) and Convolutional Neural Network. The four different combinations of the unsupervised and supervised learning algorithms are applied to an eye tracking dataset for a study on consumer choice in the Netherlands. Their performance is compared in terms of the classification of samples into fixations and saccades, and several eye movement statistics. The results show that the supervised learning methods are able to learn the latent data structure that is identified by the unsupervised methods, but adjust the labels where needed. Both the classification of samples into fixations and saccades as well as the eye movement statistics obtained by the four unsupervised-supervised methods are in line with each other, whereas the results from the unsupervised methods independently vary a lot. Using different choice tasks to train the supervised methods do not majorly influence the results, suggesting the method’s ability of transfer learning. The proposed hierarchical combination of unsupervised and supervised learning is thus a promising method to identify fixations in eye tracking data in a general applicable manner.

The remainder of this paper is structured as follows. Section 2 provides a review on the relevant existing literature on eye movement data analysis. The eye tracking data used for this research is described in Section 3. The proposed methods to answer the research questions are explained in Section 4. In Section 5, the results are presented and discussed. Finally, Section 6 concludes this research and provides ideas for future research.

2 Literature

2.1 Background

Initial eye movement research dates back to the early 1900s (Rayner, 1998). Since then a large number of different algorithms aiming to detect fixations have been developed. These fixation detection algorithms translate raw eye movement data points to fixations locations. For a long time, two classes of fixation detection algorithms were distinguished. Firstly, the velocity-based algorithms use horizontal and vertical differences between measurements to detect saccades and assume the rest to be fixations. A sequence of eye movements is defined as a saccade if eye speed exceeds the speed threshold. The most popular algorithm using the velocities of samples is the Identification by Velocity Threshold of Bahill et al. (1981) and Salvucci and Goldberg (2000). The other type of fixation detection algorithms is dispersion-based. Since fixation points tend to cluster together, these algorithms use the location of eye-samples directly to detect fixations and assume the rest to be saccades. Fixations are identified as a sequence of eye movements that do not exceed the spatial threshold for a given duration. One of the most common dispersion-based algorithms is the Identification by Dispersion-Threshold algorithm of Salvucci and Goldberg (2000) that measures the dispersion as the distance between points in the fixation that are the farthest apart. Another method is to define dispersion as the distance between data points and the fixation center (Camilli et al., 2008).

A drawback of both classes of algorithms is that they require the pre-determination of a large number of thresholds. Consequently, the fixation identification algorithms may be sensitive to the fixed, a-priori thresholds. Different choices of algorithms and their thresholds can result in systematic differences in identified fixations, and consequently in less reliable and less comparable interpretations of eye-tracking data (Shic et al., 2008). In addition, both types of algorithms usually do not account for heterogeneity in the stimulus. For example, fixations on text typically exhibit mostly side-to-side eye-movement, while fixations on an image may consist of movement in both directions. Differences in the spatiotemporal characteristics of eye movements of observers is also not accounted for, while the characteristics of fixations and their role in information processing may exhibit systematic differences between tasks and individuals (Andrews and Coppola, 1999; Rayner et al., 2007). For instance, it is shown that gender of observers influences ocular behaviour (Meyers-Levy and Maheswaran, 1991; Pan et al., 2004). To allow for differences in characteristics of fixations between both stimuli and individuals, the algorithm thresholds should vary between both individuals and tasks. A final drawback is that eye-tracking data are incorrectly classified if the algorithms are applied to data with a sampling frequency outside the intended range, or if the data contain noise, post-saccadic oscillations and smooth

pursuit (Holmqvist et al., 2011). Noise may include noise in the recording system (e.g., in the signal itself, video, or voltage) and physiological artifacts (including head movements, or changes in pupil size). Therefore, a more objective method for classifying eye-tracking data would be appropriate, accounting for both individual variation and noise variation.

Recently, event detection algorithms have been improved by introducing adaptive individual- and task-specific thresholds (Engbert and Kliegl, 2003; Nyström and Holmqvist, 2010; Mould et al., 2012). However, researchers still need to set specific parameters.

2.2 Machine Learning

In the last few years, research in the area of the detection of eye movements has been emerging with the increase in popularity of machine learning methods. The first machine learning approach introduced Hidden Markov Models that do not need fixed, a-priori thresholds based on the individual, task, and stimuli (Salvucci and Anderson, 2022). Instead, fixations are probabilistically identified based on the different distributions of velocities during fixations and saccades. Although this yields more robust fixation detection than the state-of-the-art algorithms, Hidden Markov Models are computationally unattractive and difficult to implement. Recent increases in computational resources have resulted in the development of many new machine learning algorithms. Their ability to learn from previously seen data and to handle enormous data sets make machine learning appealing for big data. Machine learning methods are typically divided into two categories: supervised learning classifies unseen data based on labeled input and output data, whereas unsupervised learning clusters data based on patterns or similarities of only input data.

For the detection of events in eye movement data, both supervised and unsupervised learning methods have been applied with great success relative to state-of-the-art algorithms. Supervised techniques include Random Forest (Zemblys et al., 2018), Support Vector Machine (Wu et al., 2010; Rello and Ballesteros, 2015), Naive Bayes classifier (Bhattarai and Phothisonothai, 2019), and Convolutional Neural Network (Hoppe and Bulling, 2016; Wang et al., 2016; Arsenovic et al., 2018; Yin et al., 2018). In contrast, unsupervised clustering algorithms aim to identify inherent patterns that can be used to cluster eye tracking data. For instance, Otero-Millan et al. (2014) cluster eye movement velocities using k -means, Krassanakis et al. (2014) propose a dispersion-based algorithm with a two-steps spatial threshold, and Fuhl and Kasneci (2022) apply a combination of k -means clustering and Principal Component Analysis to cluster eye movement data. In general, unsupervised machine learning methods provide less accurate classification than supervised ones if the training data is representative and labeled properly. Zemblys

(2017) compares the performance of ten machine learning algorithms in the identification of eye movement events, and conclude that Random Forest outperforms the other supervised and unsupervised methods. This supervised method provides accurate event detection output that is robust to noise and data sampling frequencies. However, a big drawback of supervised machine learning in the context of eye movement data is the large amount of labels that are required to train supervised methods. Since there is no golden standard or objective truth on when a fixation starts or ends (Andersson et al., 2017), these labels are usually hard to obtain. Some studies use human-made labels but they require extensive effort and may be biased, or inconsistent due to human errors (Kothari et al., 2020). For example, Hooge et al. (2018) compared the classified fixations of twelve human coders, and found that different thresholds and selection rules were used among the coders, resulting in substantial differences between fixation duration and number of fixations. Additionally, supervised learning methods tend to suffer from overfitting when the data is noisy, high-dimensional and/or complex. Overfitted supervised learning algorithms cannot generalise because they memorise noise in the training data as concepts.

I propose a method that hierarchically combines supervised and unsupervised learning. First, an unsupervised algorithm labels eye tracking data into fixations and saccades by identifying inherent data patterns. Thereafter, a supervised technique learns the identified pattern between the two clusters and provides further classification. Combining supervised and unsupervised learning has recently shown great potential in several research domains. In medical research for instance, Mishra et al. (2022) uses unsupervised K-Means clustering to identify outlier attributes. The resultant data are used as input to the supervised Naive Bayes classifier to determine chronic disease risks' presence. Gatidis et al. (2015) train a Support Vector Machine classifier using labeled prostate cancer data obtained from a spatially constrained Fuzzy C-Means algorithm. Kim et al. (2022) also use Fuzzy C-means clustering as preprocessing unit of the supervised fuzzy max-min Neural Network to diagnose diabetes. They find that the hierarchical combination of unsupervised and supervised learning may solve overfitting issues since the labels obtained by unsupervised learning can function as an intermediate concept or noise filtering scheme for the supervised method. The proposed method of Hashemzadeh and Azar (2019) first uses Fuzzy C-Means clustering to extract the thick and clear blood vessels, after which a Decision Tree extracts the thin vessels only from non-vessel regions detected in the previous step. Compared to the individual supervised method, the hierarchical combination of supervised and unsupervised learning deals much better with the problem of intra-class heterogeneity in vessel appearance. Training a supervised method on data where the same features vary a lot within a vessel class may reduce the overall efficiency, because the training data does not contain sufficient data to

cover the range of diversity. The proposed method was successful when it was trained on a dataset and tested on another dataset. In additional research areas, Shah and Murtaza (2000) and Du Jardin (2021) use a Neural Network based clustering method to predict bankruptcy, and conclude that combining supervised and unsupervised learning yields robust and accurate results. Ippolito et al. (2021) combine Self-Organising Maps and Random Forests to classify facies, and find this method to be more accurate than using these algorithms independently. In the area of human activity recognition, the combination of K-Means clustering with Graph Convolutional Networks of Budisteanu and Mocanu (2021) outperforms the individual supervised method. The unsupervised learning method extracts the salient information in the data and provides valuable insights for the supervised method.

To conclude, the hierarchical combination of unsupervised and supervised learning shows encouraging results in the existing literature. The unsupervised learning algorithm uses the latent structure inherent in a dataset to provide the supervised method with informative labels. This yields more accurate results than both methods independently, prevention of overfitting, a solution to heterogeneity issues, and transfer supervised learning. To the best of my knowledge, combining supervised and unsupervised learning has not been applied in the context of eye movement data. In this field in particular, the complementarity between the two techniques seems appealing: the superior performance of supervised machine learning is combined with the ability of unsupervised learning to identify the latent data structure.

This paper contributes to the field of fixation identification in eye tracking data in three aspects. Firstly, a hierarchical combined supervised-unsupervised learning clustering algorithm is proposed to identify fixations in eye tracking data. Secondly, the proposed method reduces the effort required to label eye tracking data for the supervised learning method. The labels are based on the latent data structure rather than on subjective expert judgement. Thirdly, the proposed algorithm automatically adjusts for the differences in eye movements for participants and tasks, and hence is generally applicable to any dataset.

3 Data

3.1 Data description







In order to empirically examine whether combining supervised and unsupervised learning is beneficial for the identification of fixations, eye tracking data for a study on consumer choice is used. This study was done at Tilburg University in the Netherlands in 2016 by Martinovici (2019). 446 students from Tilburg University were asked to make brand choices in five product

categories on simulated websites: toothbrush (practice task), light bulb (task 1), travel mug (task 2), TV (task 3), fridge (task 4). The complete set of product attributes are included in Appendix A. The participants were stimulated to make choices that align with their preferences by both a compensation for participation and by a lottery which prize was one of the chosen products. The participants were informed that for each of the five choice tasks, they were shown three slides. The first slide displays a description of the task and an example of the website. The description contains information about the quality/price ratio, the goal of the choice task, and the amount of time for which the next slide is shown. On the second slide, the website and the product descriptions are shown for a fixed amount of time. The third slide shows the four brand names and asks the participant to choose the brand he/she would buy. The first task (toothbrush) is used as an example to familiarise the participants with the tasks. All slides were projected on a (320 x 88 mm) screen and participants continued to the next slide by making one click, apart from the slides showing product descriptions for a fixed amount of time.

There are eight experimental conditions consisting of the goal of the task (environmentally friendly and performance), the duration of the task (low time pressure and high time pressure), and the presented order of the brands (ABCD and DCBA). The participants were randomly assigned to one of the experimental conditions. So in total, there are $5 \times 8 = 40$ files of binocular data. Two examples of the different conditions and tasks are depicted in Figure 2. The eye movements were recorded using a Tobii T60XL eye tracker (www.tobii.com) with a sampling frequency of 60 Hz. This implies that eye movements were recorded every 16.67 ms. Participants were seated in front of the screen at a distance of approximately 625 mm. Stimuli were displayed at a resolution of 1920 x 1200 pixels. Prior to the choice tasks, the eye tracker was calibrated for each participant. The raw data consist of a total of approximately 607,800 samples for each task. A sample refers to a time-stamped (x, y) coordinate pair for each of both eyes. Additionally, the data contain participant keys, the experimental condition, and the distance to the screen for both eyes.

3.2 Cleaning

Raw eye tracking data may contain outliers and missing data points due to eye blinks and recording issues. To reconstruct the missing data points, the Piecewise Cubic Hermite Interpolating Polynomial (Pchip) (Kahaner et al., 1989) is used. This interpolation method is accurate for eye tracking data because it takes into account its continuity and slow variation over time by preserving monotonicity (Dan et al., 2020). A cubic Hermite interpolating polynomial $P(x)$ is performed on each subinterval $x_k \leq x \leq x_{k+1}$ for the given data points. At the interpolation

	BRAND 1	BRAND 2	BRAND 3	BRAND 4
				
	Sylvania	Osram	Philips	Megaman
Bulb type	Halogen	LED	Halogen	Energy saver
Energy efficiency class				
Wattage	28 W	10 W	42 W	11 W
Voltage	220-240 V	220-240 V	220-240 V	220-240 V
Light output (lumens)	345 lm	630 lm	803 lm	630 lm
Equivalent to	40 watts	60 watts	60 watts	52 watts
Colour	Warm white	Warm white	Warm white	Daylight
Average lifetime	2000 hours	25000 hours	2000 hours	15000 hours





	BRAND 1	BRAND 2	BRAND 3	BRAND 4
				
	Zuperzazial	Grace	Monbento	Aladdin
Volume	400 ml	470 ml	500 ml	350 ml
Size	9.5 x 9.5 x 14.5	8.7 x 12.7 x 20.3	6.4 x 6.4 x 19.4	8.0 x 7.5 x 20.0
Material	Bamboo	Thermo Plastic & Double Glass	Plastic	Thermo Plastic
Recycled material	Yes	No	No	Yes
Weight	200 g	186 g	120 g	270 g

Figure 2: Light bulb and travel mug example

points, the derivatives (slopes) of $P(x)$ are specified, respecting the monotonicity of the data. As a result, $P(x)$ is monotonic on intervals where the data points are monotonic, and $P(x)$ has a local extremum at points where the data have a local extremum. This method has no overshoots if the data is not smooth.

Since the ability of interpolation methods to accurately infill data decreases proportionally as the number of consecutive missing values increases (Kornelsen and Coulibaly, 2014), missing eye locations are interpolated if the duration of periods of missing data does not exceed 100 milliseconds, and if at least 50% of the data for the participant for the task are available. Observations for which there are still no gaze points after interpolation are removed. If the remaining samples for a participant take up less than 5 seconds for a task, the data are considered to be unreliable and are deleted. These datasets do not yield sufficient information as they either have too many deleted observations, or the participant did not spend enough time looking at the choice task. Since the gaze position for the left and right eyes are expected to be similar (Zhao et al., 2013), observations for which the difference in location between two eyes exceeds 500 pixels are expected to suffer from measurement errors and are deleted. The above-mentioned cleaning procedure results in discarding 9.2% of the observations ending up with 429-433 participants per task. On average, there are approximately 1,300 observations per task per participant, which corresponds to 21.7 seconds. Some more specific statistics on the data per task can be found in Appendix B.

4 Methodology

To detect fixations in eye tracking data, a combination of unsupervised and supervised machine learning is applied. This approach first uses an unsupervised learning algorithm to label the data

points as either fixations or saccades. Thereafter, a supervised learning algorithm is trained on the labeled data to identify fixations. I use two unsupervised and two supervised learning algorithms yielding four hierarchical combinations of unsupervised and supervised learning methods. The performance of the four combinations are compared to each other and to the performance of the unsupervised methods independently.

4.1 Unsupervised Learning

The main objective of the unsupervised learning method is to cluster similar activities together by identifying inherent latent structures in the data. To label the eye tracking data, two different unsupervised learning methods are used: the velocity-based Binocular-Individual Threshold (BIT) algorithm of Van der Lans et al. (2011), and the dispersion-based Eye Movements Metrics & Visualizations (EMMV) algorithm of Krassanakis et al. (2014).

4.1.1 Binocular-Individual Threshold algorithm

Van der Lans et al. (2011) proposed the velocity-based BIT algorithm for parameter-free fixation detection using eye tracking data of both eyes. Based on the natural within-fixation variability of both eyes, the algorithm automatically identifies velocity thresholds that are specific to each of the eyes, to directions of eye movements, to tasks and to individuals. Samples corresponding to velocity that exceeds the within-fixation variability are labeled as candidate saccades. The BIT algorithm utilises the fact that both eyes are often directed at the same location to distinguish saccades from noise. If both eyes show a peak in velocity simultaneously, this velocity peak is likely to be a real movement rather than noise. The BIT algorithm automatically eliminates eye blinks and other recording abnormalities before identifying fixations. This pre-processing procedure consists of determining correctly measured samples, valid distances between gaze points from both eyes, and potential outliers.

The observed eye tracking data at time sample $t = 1, \dots, T$ $z_t = (x_{t,l}, y_{t,l}, x_{t,r}, y_{t,r})$ consist of the x-y locations of the left and right eye. Taking the first difference $\Delta z_t = z_t - z_{t-1}$ yields the velocities of each eye tracking data sample. The within-fixation variability is assumed to have a (multivariate normal) distribution with individual-, eye- and task-specific means and covariance matrices. To estimate the means and covariances, the (fast) Minimum Covariance Determinant method (Rousseeuw, 1984; Rousseeuw and Driessen, 1999) is used. This statistical method bases its estimates on the subset of observations with the smallest determinant of the covariance matrix. In this way, outliers are robustly detected such that they do not affect the estimates extravagantly. Increases in Δz_t are considered as outliers relative to the within-fixation

distribution of Δz_t . To detect saccades, i.e., data points with ‘extreme’ velocities relative to the within-fixation variability, a multivariate Shewhart control chart procedure is used which assumes that data is generated from a multivariate distribution with mean μ and covariance Σ . In the context of eye tracking data, μ indicates the mean fixation variability, whereas Σ represents the within-fixation variation. Extreme data points Δz_t that are unlikely to be generated from this distribution are flagged. For large T , it follows from the normal distribution that the variable $w_t = (\Delta z_t - \mu)' \Sigma^{-1} (\Delta z_t - \mu)$ is approximately χ^2 -distributed with four degrees of freedom. For each individual and task, the robustly estimated mean and covariance are computed. For each velocity Δz_t , w_t is used to determine whether the corresponding data point is a candidate saccade. The control ellipse represents the velocities Δz_t for which $p(w_t|\mu, \Sigma)$ equals χ . Hence, velocities that lie inside the control ellipse, i.e., velocities for which $p(w_t|\mu, \Sigma) > \chi$, are consistent with the within-fixation variability for that individual and task. On the other hand, velocities that lie outside the control ellipse, i.e. velocities for which $p(w_t|\mu, \Sigma) < \chi$, are unlikely to be due to within-fixation variability and are classified as candidate saccades. The unlikely variations in these velocities will not always correspond to saccades, but may also be due to blinks and other anomalies. Samples are qualified as saccades if at least two consecutive velocities are outside the control ellipse. The control ellipse is thus determined from the observed within-fixation variability. This allows for different, automatically set thresholds that may vary in the x - and y -direction and that may vary across both tasks, eyes, and individuals. In this way, the statistical information available in the variability of both eyes is exploited. A more detailed description of the algorithm can be found in Van der Lans et al. (2011).

One of the parameters that need to be set is the minimum fixation duration. It is stated by Manor and Gordon (2003) that setting the threshold for minimum fixation duration to 100 milliseconds is in line with theory, and yields a balance between the risk of identifying false fixations resulting from a too low threshold and the risk of missing fixations because of a too high threshold. The parameters representing the height and width of the screen that displays the choice tasks are adjusted to 1920 x 1200 pixels. The control percentage in quality control is set to its default value of $1 - \sqrt{0.001}$, which corresponds to a control limit of 0.001 that is usually used in quality control as explained in Van der Lans et al. (2011). The last parameter includes the maximum number of consecutive samples that are not tracked (i.e., blinks or missing data points) within a fixation. However, after the data cleaning process described in Section 3, there are no missing sample values. Hence, this parameter value is redundant and its default value of 3 is used (i.e., 50 ms).

4.1.2 Eye Movements Metrics & Visualisations algorithm

The second unsupervised algorithm used in this research is the dispersion-based EMMV algorithm of Krassanakis et al. (2014). This is a two-step spatial fixation detection algorithm that can be used as a spatial noise filtering approach during the detection of fixation. The proposed algorithm is based on spatial and temporal constraints that define the spatial characteristics of fixations. The parameters include two spatial dispersion thresholds $t1$ and $t2$, and a threshold for minimum fixation duration v . The dispersion is computed by applying a “two-steps” spatial threshold. In both steps, the Euclidean distance between data point $z_t = (x_t, y_t)$ and the fixation’s mean point (m_x, m_y) is compared to a threshold. This implies that the spatial threshold is defined through a circle rather than through a rectangle, as usually done in Identification by Dispersion-Threshold (I-DT) algorithms (Salvucci and Goldberg, 2000; Nyström and Holmqvist, 2010). The EMMV algorithm is visualised in Figure 3, and described in Algorithm 1. Since eye movement characteristics vary across both individuals and tasks (Andrews and Coppola, 1999), this algorithm is applied per task per participant. The gaze points from the left and right eye are averaged as is usually done in eye tracking research (Hooge et al., 2019).

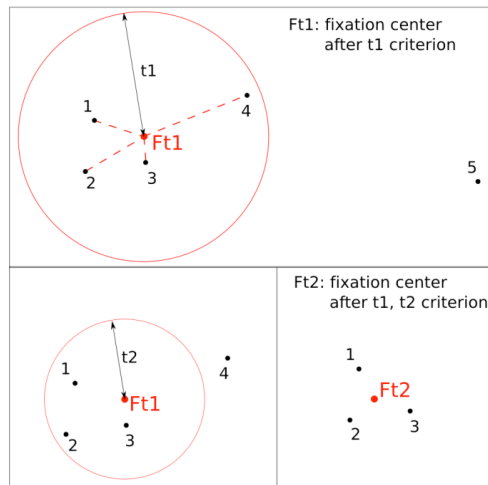


Figure 3: Visualisation EMMV - source: Krassanakis et al. (2014)

The first spatial parameter $t1$ corresponds to the maximum distance between a gaze point and the center of the fixation cluster. It describes the limited spatial distribution of fixations as it takes into account the relative stationarity of human eyes. This dispersion threshold should include at least 0.5° visual angle (Salvucci and Goldberg, 2000). In this way, the risk of a too low threshold leading to exclusion of fixations of people with a large amount of tremor is balanced with the risk of a too high threshold resulting in misclassification of saccades as fixations or incorrectly merged fixation clusters. For the data described in Section 3, 0.5° visual angle corresponds to 75 pixels,

hence $t1$ is set to 75. The implementation of the second spatial parameter $t2$ ensures consistency among the raw data of each fixation cluster by removing the noise that was produced during the recording process. Recording noise is one of the sensitive points in the performance of state-of-the-art I-DT algorithms as discussed in Section 2. With a sampling frequency of 60Hz, noise including ranges up to 16.67 ms is added to the recording process (www.coonect.tobii.com). The average (absolute) eye movement (aem) during 16.67 ms is defined as the average (absolute) eye movement between two consecutive samples: $aem = \frac{1}{2(T-2)} \sum_{t=2}^T |\Delta x_t| + |\Delta y_t|$ where $\Delta x_t = x_t - x_{t-1}$ and $\Delta y_t = y_t - y_{t-1}$. As suggested in Krassanakis et al. (2014), the statistical interval of 3 is implemented to calculate $t2$, i.e., $3aem$. This results in $t2 = 45$. The minimum fixation duration is set to $v = 100$ ms following the same reasoning as for the BIT algorithm.

Algorithm 1 EMMV algorithm

Require: data in the form of $(x, y, time)$, and

Require: set values of parameters $(t1, t2, v)$

```

1: for  $z_t = (x_t, y_t)$ ,  $t = 1, \dots, T$  per task per participant do
2:   while  $\|(m_x, m_y), (x_t, y_t)\| < t1$  do
3:     compute the mean value of horizontal and vertical coordinates  $m_x$  and  $m_y$ 
4:     if  $\|(m_x, m_y), (x_t, y_t)\| > t1$  then
5:       generate a new fixation cluster and go to Step 2
6:     end if
7:   end while
8: end for
9: for cluster  $k = 1, \dots, K$  do
10:  for  $z_t \in k$ ,  $t = 1, \dots, T$  do
11:    if  $\|(m_x, m_y), (x_t, y_t)\| > t2$  then
12:      remove  $z_t$ 
13:    end if
14:  end for
15:  Compute fixation's coordinates as  $k$ 's mean point
16:  Compute fixation's duration as the difference of the time between the last and first record
17:  if  $k$ 's fixation duration  $< v$  then
18:    remove  $k$ 
19:  end if
20: end for

```

4.2 Combining Supervised and Unsupervised Learning

Two supervised learning methods are trained on the labeled data obtained from the unsupervised learning algorithm, namely Random Forest and Convolutional Neural Network.

4.2.1 Feature extraction

For equal comparison between the methods, the same features are used for both RF and CNN. It is shown that velocity-based features work well with high-quality data, and spatial features work better for noisy and low sampling rate data. Since the eye tracking dataset described in Section 3 contains noisy and low sampling rate data, spatial features are expected to be important in deciding whether a sample belongs to a fixation or saccade. Both *velocity* and *dispersion* are included as features. The most common way to measure *dispersion* is $(x_{max} - x_{min}) + (y_{max} - y_{min})$ over a 100 ms window (Salvucci and Goldberg, 2000). As suggested in Nyström and Holmqvist (2010), *velocity* is calculated by means of a Savitzky–Golay (SG) smoothing filter (Savitzky and Golay, 1964) with polynomial order 2 and a window size of 50 ms. The SG smoothing filter aims to increase the precision of the data by reducing high frequency noise and preserving high frequency signal. The filter fits the polynomial function that best describes the raw data in each 50 ms window, differentiates the polynomial analytically, and resamples it to the original sampling frequency.

Additionally, Zemblys et al. (2018) found three features that were most important in deciding whether a sample belongs to a saccade or fixation for their RF. Feature importance was based on both univariate feature selection as well as mean decrease impurity and mean decrease accuracy. The main two features include the distance between both the mean and the median gaze in a 100 ms window preceding and succeeding the sample: *meandif* and *mediandif* proposed by (Olsson, 2007). These spatial features represent movement, but are unaffected by noise as their values are only large in the case of a real movement. The third most important feature in Zemblys et al. (2018) is the standard deviation *stdev* of the eye location in a 100 ms window centered on a sample. This is a common measure to describe eye tracker noise (Holmqvist et al., 2011). Furthermore, the differences between this noise measure are calculated for 100 ms windows preceding and succeeding the current sample. This feature *stddif* is used in Zemblys et al. (2018) and was inspired by Olsson (2007). The largest differences in *stddif* should correspond to the start and end of saccades. The last feature that is included, is *rayleightest* which is suggested by Larsson et al. (2015). It represents the probability of the sample-to-sample directions in a 50 ms window being uniformly distributed around the unit circle. The remaining features of Zemblys et al. (2018) are not included, because they are highly correlated with included features

and were less important in deciding whether a sample belongs to a fixation or a saccade. As a result of leaving out the less important features, the features that are highly correlated with the left out features will provide more unique information to the classifier.

To get insight in the extent to which features will add extra information compared to the other features because of high correlation, Spearman’s rank correlation between the features is calculated as $\frac{\sum_{t=1}^T (u_t - \bar{u})(v_t - \bar{v})}{\sqrt{\sum_{t=1}^T (u_t - \bar{u})^2 \times \sum_{t=1}^T (v_t - \bar{v})^2}}$, where $u_t = \text{rank}(x_t)$ and $v_t = \text{rank}(y_t)$ (Spearman, 1961). The results are displayed in Figure 4. Correlations between features that describe similar properties of the data are strong. For example, it can be seen that *stdev*, *dispersion* and *mediandif* are highly correlated with each other ($r \in [0.94, 1]$). Since *stdev* and *dispersion* are both measures of precision, their high correlation implies that they provide very similar information. The reason for high correlation between *mediandif* and the two precision measures is probably that all three features represent the amount of sample-to-sample movement or spatial spread of the data. For the same reason is *meandif* relatively strong correlated with *meandif* with *stdev*, *dispersion* and *mediandif* ($r \in [0.78, 0.82]$), since *meandif* also describes movement. The finding that the correlations of *stddif* with all other features are close zero implies that this feature provides some unique information that is not reflected in any of the other features. Although correlations of *rayleightest* with the other features are a bit higher ($|r| \in [0.019, 0.47]$), this feature still holds a significant amount of unique information. *velocity* is moderately correlated with all the other features ($r \in [-0.47, 0.67]$).

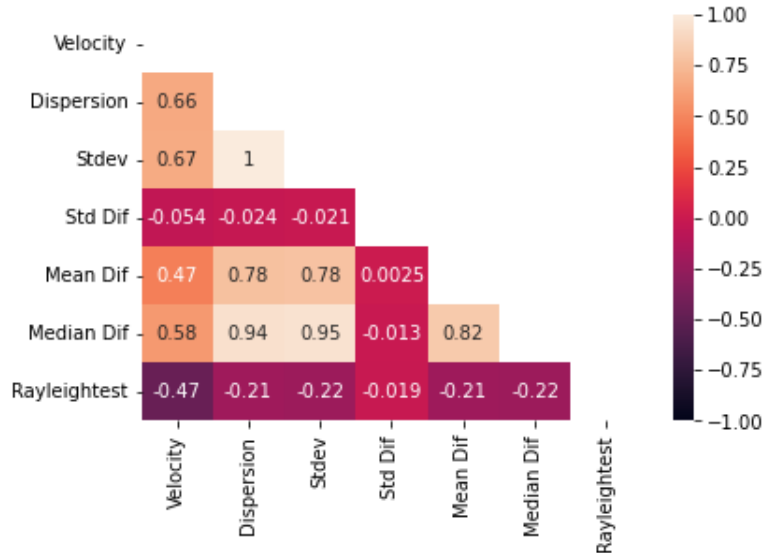


Figure 4: Spearman’s rank correlation between features

4.2.2 Random Forest

The motivation behind the choice for applying Random Forest (RF) (Breiman, 2001) to eye tracking data with the purpose of classifying fixations, is that it is shown that RF outperforms both state-of-the-art detection algorithms (Zemblys et al., 2018) and other supervised learning methods including Support Vector Machine (Dong et al., 2016), Naive Bayes (Zemblys, 2017) and Recurrent Neural Network (Kothari et al., 2020) in classifying fixations.

RF consists of multiple uncorrelated decision trees that operate as an ensemble. A decision tree consists of a series of decisions. Trees are grown according to the bagging procedure, i.e., a random subset of data points are drawn without replacement. For each so called bootstrap sample, a randomly selected subset of features is considered. As a result, the correlation between trees is reduced. Decision trees aim to split the data at so-called nodes. A node is a condition on a feature to split the data. To place a split, the algorithm determines which feature and what cut-off value maximises the heterogeneity within the partitions created by the split. As long as there is sufficient heterogeneity among partitions, the tree continues to grow by splitting the data. When we eventually arrive at the leaf node, the tree decides whether the sample belongs to a fixation or a saccade. In classification problems, the RF's final prediction is the class that obtains the majority vote of the decision trees. This procedure is summarised in Algorithm 2. As a result of the reduced correlation between trees, the variance of the RF's prediction decreases. Bagging usually yields a slight increase in bias, but this is compensated by the decrease in variance.

To examine the ability of transfer learning, the eye tracking data are split into training and test data according to the following procedure. For each participant, one choice task is used to train the model. The trained RF is then used to classify fixations for this participant for each of the remaining four tasks. As an example, the labeled data on task 1 for the first participant are used to train the RF. Thereafter, the trained RF is used to classify fixations for task 2, 3, 4 and 5 of the first participant. The same process is applied using the other tasks as training data, and for each participant. The results are averaged across participants. By applying this procedure per participant, the significant differences in eye movements across individuals are accounted for. Transfer learning implies that a model trained on a task is reused on another, related task. The knowledge that was learned from the first task is transferred to the new task. If the results are not affected by training using different tasks, this favours the RF's ability of transfer learning. Through training with the labeled data obtained from unsupervised learning, the Random Forest learns to detect fixations by identifying combinations of features. To optimise the RF, the hyperparameters of the RF are tuned by means of k -fold cross validation. Because of time constraints,

Algorithm 2 Random Forest for Classification

Require: data in the form of $(x, y, time)$

Require: parameter values m, n_{min}

```
1: for  $b = 1$  to  $B$  do
2:   (a) Draw a bootstrap sample  $Z^*$  of size  $N$  from the training data
3:   while the minimum node size  $n_{min}$  is not reached do
4:     Grow a tree  $T_b$  to the bootstrapped data by recursively repeating the following steps for
       each terminal node of the tree
5:       i. Randomly select  $m$  variables from the  $p$  features
6:       ii. Pick the best variable/split-point among the  $m$ 
7:       iii. Split the node into two daughter nodes
8:   end while
9:   (b) The class prediction of the  $b^{th}$  tree at a new point  $x$  is  $\hat{C}_b(x)$ 
10: end for
11: The prediction of the RF at a new point  $x$  is  $\hat{C}_{rf}^B(x) = majority\ vote\{\hat{C}_b(x)\}_1^B$ 
```

this procedure is only applied on the first task, after which the most frequent hyperparameter values are used in the RF. The complete process is summarised in Algorithm 3, where lines 1-5 describe the cross validation for the first task, and lines 6-14 explain transfer learning for all tasks.

I created a random search using the Python method 'RandomizedSearchCV' of the package by Buitinck et al. (2013). This technique randomly selects and tests combinations from the grid of hyperparameter values, and is therefore more efficient than considering all possible combinations. The number of different combinations to consider in each grid search is set to 100. The number of folds to use for cross validation for each RF is set to 3 to ensure that each fold contains both fixations and saccades. The combinations of hyperparameters are evaluated based on their accuracy classification score. The hyperparameters and their ranges are depicted in Table 17. The number of trees in the RF $n_estimators$ ranges from 50 to 200 with step size of 50. This range of trees is suggested by Oshiro et al. (2012) as it yields a good balance between performance, computational efficiency, and memory usage. The *maximum number of features* that are considered at each split range from 1 to 6. The maximum number of levels in a tree is denoted by *maxdepth* and ranges from 10 to 25 with step size 5. The minimum number of samples that is required to split a node is indicated by *minsamplesplit*, whereas the minimum number of samples that is required at each leaf node is denoted by *minsamplesleave*.

The remaining hyperparameters are set as follows. Due to the nature of the human visual system,

Algorithm 3 Random Forest Transfer Learning

Require: create grid by specifying possible values of the hyperparameters of the RF

Require: set the number of combinations to consider in each grid and the number of folds

```
1: for each participant do
2:   (a) Train the random forest on Task 1 using  $k$ -fold cross validation
3:   (b) Save the hyperparameter values of the best fit
4: end for
5: Create random forest with the most frequent hyperparameter values from the cross validation
6: for each choice task  $c = 1, \dots, 5$  do
7:   for each remaining task  $r$  do
8:     for each participant do
9:       i. Train RF on  $c$  and predict  $r$ 
10:      ii. Save predictions and evaluation measures
11:     end for
12:     Average evaluation measures over all participants
13:   end for
14: end for
```

the majority of eye tracking samples belongs to fixations rather than saccades. For example, Hooge et al. (2018) find that 71.1% of samples belong to fixations in their dataset, whereas Tinker (1928) reports an average of 94.4%. As a consequence of this imbalance in the data, some bootstrap samples might contain few or even no saccades which results in poor classification due to bias towards the majority class (i.e., fixations) (Chen et al., 2004). To deal with imbalanced data, the balanced subsample weighting method is used. This method calculates weights that are inversely proportional to the class frequencies in each bootstrap sample: $\frac{N}{2 \times n_{fix}}$, where N represents the bootstrap sample size, and n_{fix} the number of fixations in the sample. Lastly, the *criterion* is a function used to measure the class label distribution in a node, representing the quality of a node split. Since different criteria hardly result in different decisions made by the tree (Raileanu and Stoffel, 2004), the Gini impurity measure is used because it is computational fast. It is calculated as $1 - p_{fix}^2 - p_{sac}^2$, where p_{fix} denotes the fixation frequency in a node, and p_{sac} the saccade frequency.

After training on one task, the test data consisting of the four remaining tasks is used to evaluate the model’s generalisation to unseen data. The evaluation process is described in the next subsection.

Table 1: Grid search RF

Hyperparameter	Grid
<i>n_estimators</i>	{50, 100, 150, 200}
<i>max_features</i>	{1, 2, 3, 4, 5, 6}
<i>max_depth</i>	{10, 15, 20, 25}
<i>min_samples_split</i>	{2, 4, 6, 8, 10}
<i>min_samples_leave</i>	{1, 2, 3, 4, 5}
<i>bootstrap</i>	{True, False}

4.2.3 Convolutional Neural Network

The Convolutional Neural Network (CNN), first introduced by LeCun et al. (1989), is a deep learning neural network that is particularly satisfactory at finding patterns in data that have a grid-like topology. CNN suits the nature of eye tracking data as it resembles the part of the human brain that is responsible for organising and processing visual information. CNN has successfully been applied in classifying eye movements (Anantrasirichai et al., 2016; Hoppe and Bulling, 2016; Wang et al., 2016; Arsenovic et al., 2018; Yin et al., 2018).

A CNN typically consists of three layers: a convolutional layer, a pooling layer, and a fully connected layer.

The convolutional layer, also known as feature extractor layer, is the core building block of a CNN. This layer extracts patterns from the input data by applying filters to the input data. The input data are the features described in the first subsection. Convolution is the process of a pattern detector checking if the pattern is present. A linear operation performs element-wise multiplication of the array of input data and an array of weights, called a filter or a kernel. The kernel is applied systematically to each overlapping part or filter-sized patch of the input data. Multiple kernels enable the CNN to learn to detect a variety of patterns from the input data. Due to time constraints, the CNN consists of one convolutional layer with 32 kernels. These kernels have size 1x3 which implies that the kernels slide or convolve over each 1x3 block of the input data. This size generally yields accurate results (Ahmed et al., 2020). If kernels detect a specific pattern at a given spatial position of the input, activation occurs. The activation function defines the output value of kernel weights. The convolution layer uses the Rectified Linear Unit (ReLU) function as the activation function, which is the most widely used activation function in CNN. It is defined as $f(u) = \max(0, u)$.

The convolution layer is followed by the pooling layer. This layer performs downsampling or dimensionality reduction to prevent overfitting and to reduce computation time and memory. In

doing so, the most important information is retained. The pooling size is set to 1x2, implying that the patterns are summarised in 1x2 data blocks. Because the input data contain features along one direction, i.e., time direction, one-dimensional convolutional and pooling layers are used. An advantage of one-dimensional CNNs over two-dimensional CNNs is that one-dimensional CNNs generally use compact CNN architectures consisting of 1-2 layers (Kiranyaz et al., 2021).

The final layers of CNN include a fully connected or dense layer, and the output layer that performs the final classification task. A flatten layer is used between the pooling layer and the fully connected layer to reduce the patterns to a single one-dimensional vector. The fully connected layer activates the (batch normalised and pooled) output of the convolutional layer using the ReLU function. The output layer uses the SoftMax activation function to produce a probability distribution over the two class labels: $\sigma(\mathbf{z})_i = \frac{\exp z_i}{\sum_{j=1}^K \exp z_j}$ for $i = 1, \dots, K$, where \mathbf{z} is an $1 \times K$ input vector. Samples are predicted as fixations if their probability of belonging to a fixation exceeds 0.5.

The CNN architecture is shown in Figure 5. Using the patterns extracted through the previous layers, the network learns the optimal filters through backpropagation and stochastic gradient descent using the efficient Adam optimiser (Kingma and Ba, 2014). For the optimisation, the sparse categorical cross-entropy loss function is used.

As with the Random Forest, the CNN is trained by means of transfer learning. Every task is used to train each of the remaining tasks, for each participant. For efficiency reasons, a batch size of 1028 is used in fitting the CNN and in predicting the class probabilities.

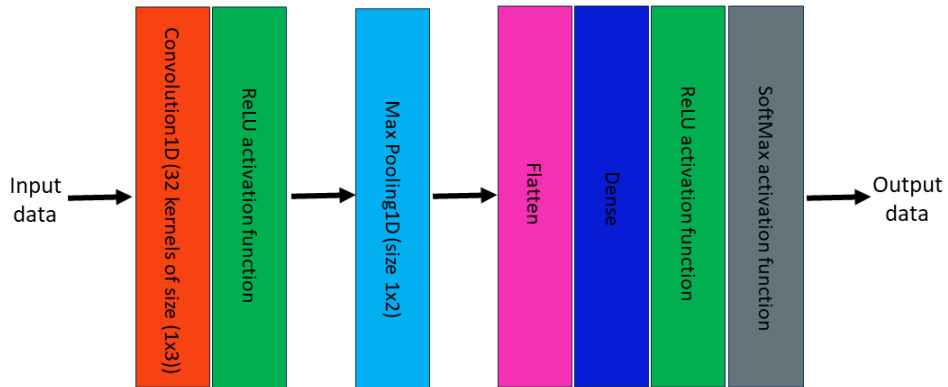


Figure 5: CNN architecture

4.3 Evaluation

Combining the two unsupervised algorithms with both supervised techniques yields four hierarchical combinations. Firstly, the obtained labels from the two unsupervised algorithms are compared and evaluated. Then, the final classification of the four combined methods are compared with each other but also with the results from the individual unsupervised method. Evaluation is done in terms of eye movement statistics, event matching, and classification metrics.

Startsev and Zemblys (2023) describe and analyse evaluation methods and measures employed in the field of eye movement event identification. Event-level quality metrics compare eye movement statistics provided by classification. Following Andersson et al. (2017), evaluation is based on the number of identified fixations, the durations of fixations (i.e. how many consecutive samples), and the variance of these fixations durations. Van der Lans and Wedel (2017) consider the minimum ratio of identified fixations to saccades to be 80%. Typically, this ratio ranges to 94.5% (Tinker, 1928). Rayner (1998, 2009); Andrews and Coppola (1999) report average fixation durations of 150–250 ms during (silent) reading, 180–275 ms during visual search, and 200–400 ms during scene viewing.

Furthermore, Startsev and Zemblys (2023) discuss evaluation based on event matching. The classification of samples into fixations and saccades is compared among the different methods. For each comparison between the classification of two models, the following percentages are calculated: the percentage of samples that is classified as fixations by both models, the percentage of samples that were classified as saccades by both models, and the percentages of samples that were classified as fixations by one model and as saccades by the other model.

To get insight into the amount of information the supervised learning methods absorb from the unsupervised learning methods, a few classification metrics on the validation data are calculated. The most widely used threshold metric for classification problems is accuracy, including in the field of eye movement event identification (Anantrasirichai et al., 2016; Hoppe and Bulling, 2016; Andersson et al., 2017). Accuracy represents the ratio between samples classified as the same events by the unsupervised and supervised learning method, and the total number of samples. One of its limitations is that this metric favours the majority class (Chawla et al., 2004). However, this issue is not too serious since the minority class will be well-represented as this research only distinguishes between fixations and saccades. To get insight into the differences between both classes, fixation accuracy is distinguished from saccade accuracy.

Secondly, the performance of the eye movement event detection is evaluated by means of the *F1*-score as suggested by Hooge et al. (2018) and as used in Bellet et al. (2019); Startsev et al. (2019) (0.83-0.96). This threshold measure represents the harmonic mean of precision and recall,

and is a popular metric for imbalanced datasets (He and Ma, 2013). The $F1$ -score ranges from 0 to 1, where 1 represents a model that classifies each observation into the class that was predicted by the unsupervised learning method, and 0 corresponds to a model that disagrees with every label obtained from the unsupervised method.

The area under the ROC curve (ROC AUC) (Fawcett, 2006) is a common ranking type metric that evaluates classifiers based on their ability to distinguish between the classes. The ROC AUC plots the true positive rate versus the false positive rate under different thresholds and measures the area under the curve. A no skill classifier predicts the majority class under all threshold values resulting in a score of 0.5, whereas a classifier that predicts the labels obtained from the unsupervised learning technique will have a score of 1.0. Scores below 0.5 imply that the algorithm classifies most of the samples that were labeled as fixations by the unsupervised learning technique as saccades, and vice versa. The ROC AUC has been used as a metric to compare eye movement event classification algorithms in Otero-Millan et al. (2014); Hoppe and Bulling (2016).

The last metric is Cohen’s Kappa Score, which measures agreement between the model’s predictions and the labels obtained from unsupervised learning, adjusted for chance (Cohen, 1960). This metric ranges from -1 to 1, where 1 corresponds to perfect agreement and 0 corresponds to a model that is no better than random guessing.

5 Results

Before evaluating the performance of combining unsupervised and supervised learning techniques, the classification results of both unsupervised methods independently are discussed.

5.1 Unsupervised Learning

Eye movement statistics of both unsupervised learning methods are summarised in Table 2. The total number of fixations per task are displayed in the third column. The average number of fixations and their standard deviation per participant for each task can be found in the fourth and fifth column of Table 2 respectively. The last three columns represent the percentage of samples that are classified as fixations, the mean fixation duration and its standard deviations respectively.

For each task, the BIT algorithm detects more fixations than the EMMV algorithm: the total number of fixations per task for BIT is around 35,000, whereas for EMMV this varies from 25,800 to 27,500. On average BIT identifies 83 fixations per participant per task relative to 62 fixations by EMMV. However, the durations of the fixations identified by EMMV are 50-70 ms

Table 2: BIT and EMMV results

		Number of fixations			Fixation duration		
		Total	Mean	Stdev	% fixations	Mean (ms)	Stdev
Task 1	BIT	35,069	81.37	27.96	80.39	225.71	145.64
	EMMV	27,289	63.32	21.50	83.97	276.29	187.17
Task 2	BIT	35,895	83.09	29.13	80.49	220.41	118.39
	EMMV	26,889	62.24	20.91	84.48	280.45	179.44
Task 3	BIT	35,823	82.73	28.58	80.21	219.72	115.26
	EMMV	27,463	63.42	21.50	83.63	270.94	179.40
Task 4	BIT	35,735	82.91	29.62	80.34	220.26	112.99
	EMMV	26,042	60.42	21.28	84.69	287.87	183.50
Task 5	BIT	35,569	82.91	29.13	79.00	217.14	118.53
	EMMV	25,800	60.14	21.13	83.65	285.78	212.08

longer than those identified by BIT, resulting in a slightly higher percentage of samples identified as fixations by EMMV (84% relative to 80%). The fixation durations identified by BIT (around 220 ms) are more in line with theory. The difference between the identified fixations by the two unsupervised learning methods is visualised in Figure 6. This Figure displays the identified saccade samples in blue, the identified fixation samples in yellow, and the fixation centers in red. The left panels show the results of BIT, while the right panels display the EMMV results. The top two panels correspond to the data for participant 1 and task 2, the bottom panels to participant 430 and task 4. From the numbers of red dots, it can be seen that BIT identifies more fixations than EMMV. The groups of blue samples in the bottom right panel of Figure 6 seem to suggest that EMMV might not be able to detect all fixations. Figure 6 also shows that both algorithms sometimes identify the same fixations, but often they are slightly different. It is interesting to notice that although the number of identified fixations is higher for BIT, the number of samples identified as fixations is higher for EMMV.

Lastly, the results of the unsupervised learning methods are evaluated based on event matching. Table 3 shows the percentage of samples that are classified as fixations or saccades by both algorithms relative to the total number of samples for each task. For example, from the top left Table it can be seen that 5.99% of the samples of Task 1 are labeled as fixations by BIT and as saccades by EMMV. The Table shows that for all tasks, the number of samples that are labeled as fixations by EMMV but as saccades by BIT is slightly higher than the number of samples that are labeled as fixations by BIT but as saccades by EMMV. This agrees with the visualisation in Figure 6. Although the differences per task are minor, it seems that the algorithms disagree

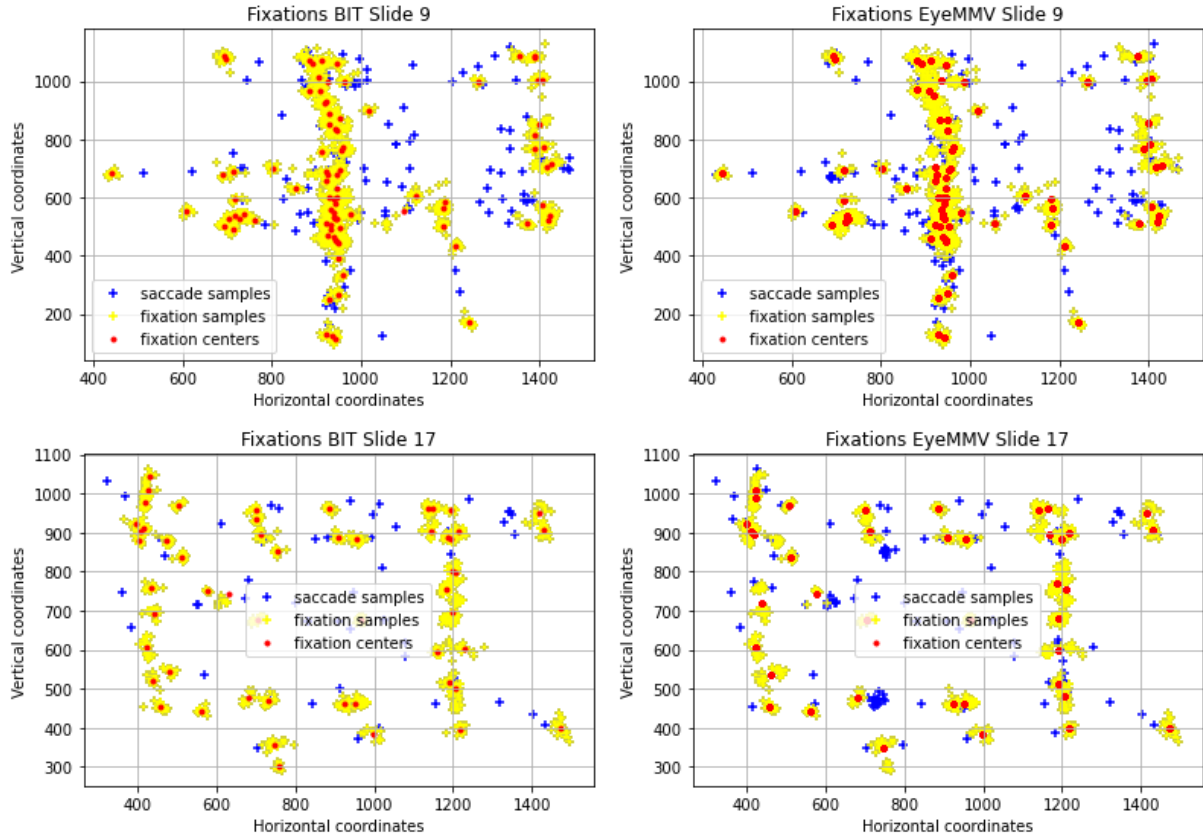


Figure 6: Examples BIT (left) and EMMV (right) results

the most for task 5. Additionally, the identified fixation centers are compared across the two unsupervised learning methods in Appendix C.

Table 3: Event matching - sample classification

Task 1		EMMV		Task 2		EMMV	
		fix	sac			fix	sac
BIT	fix	77.35%	5.99%	BIT	fix	77.94%	6.01%
	sac	6.49%	9.86%		sac	6.45%	9.29%
Task 3		EMMV		Task 4		EMMV	
		fix	sac			fix	sac
BIT	fix	77.08%	6.40%	BIT	fix	77.98%	6.14%
	sac	6.52%	9.94%		sac	6.67%	9.07%
Task 5		EMMV					
		fix	sac	fix	sac		
BIT	fix	76.71%	6.54%				
	sac	6.86%	9.67%				

5.2 Combined Unsupervised-Supervised Learning

5.2.1 Random Forest

The results of Random Forest (RF) combined with BIT and EMMV are displayed in Table 4 and 5. First of all, it is worth mentioning that the random grid search yields the same hyperparameters values for both BIT and EyeMMV labels. The values are included in Appendix D.1. The sample-level evaluation metrics as discussed in Section 4 are shown in columns 3-8. Recall that these metrics refer to the amount of information the RF absorbs from the unsupervised learning method. The last six columns display eye movement statistics. The rows indicate the task that was used to train the random forest, and the task that was predicted respectively. For example, 2-1 corresponds to the predicted labels of task 1, when the random forest was trained on task 2. The results are averaged across all participants.

To evaluate the ability of transfer learning, the blocks are arranged per predicted task with the remaining tasks used for training. For RF with EMMV labels, it can be seen that training the random forest on task 4 yields relatively large numbers of identified fixations and relatively small fixation durations. For the remaining tasks, the eye movement statistics are similar when different training data were used, and differences in statistics across tasks are detected. For example, for task 1 and 3, more fixations are identified than for task 2, 4, and 5 regardless the training task that was used. The mean fixation duration is the shortest for task 3 (around 265 ms) and the longest for task 4 (around 290 ms). This suggests that the results are in favour of the ability of transfer learning of the proposed method. For RF with BIT labels such patterns are slightly less clear as the results across the different training tasks vary more than for RF with EMMV. However, BIT-based RF also detects the highest number of fixations for task 1 and 3, but in addition also for task 2. The mean fixation duration is again shortest for task 3 (around 280 ms), and the longest for task 4 (around 305 ms). Training the RF on task 4 does not yield divergent results as with EMMV-based RF. The sample-level metrics are similar across tasks generally. However, for both RFs, the saccade accuracy is lower for the last two tasks than the first three tasks, indicating that the random forest absorbed less saccade labels from EMMV and BIT for the last two tasks. All in all, both supervised methods distinguish differences in eye movement statistics regardless of the task that was used to train the random forest. The unsupervised methods did not notice these differences, but the theory suggests that eye movements differ across tasks.

Compared to the results of the unsupervised learning algorithms independently in Table 2, the differences in eye movement statistics between BIT and EMMV are smaller after the Random Forest is applied. RF based on BIT still identifies more fixations than RF with EMMV labels.

However, the mean number of fixations for RF based on BIT (74-83) is slightly lower compared to applying solely BIT (81-83), while the mean number of fixations for EMMV has changed from 60-63 to 67-73 after applying the RF. This seems to suggest that the random forest corrects for too little identified fixations from solely the EMMV algorithm, and for slightly too many identified fixations from solely the BIT algorithm. Furthermore, the percentage of samples identified as fixations is similar for BIT and EMMV after applying the random forest. This percentage is higher than solely applying the unsupervised learning methods, and more in line with theory. The mean fixation durations are similar or slightly lower for EMMV, but have increased for BIT labels after the RF is applied. This might indicate that the RF adjusts for the too short fixation durations identified by the BIT algorithm independently.

Looking at the sample-level metrics, the high fixation accuracy implies that the labels of fixations agree more often than the labels of saccades which can be explained by the previously discussed imbalance in this ratio. Both the fixation accuracy and the saccade accuracy are higher for RF based on EMMV. This implies that the RF takes over less BIT labels than labels obtained from the EMMV algorithm. For example, the RF only agrees for just over 50% with the saccades classified by BIT independently. This explains the lower $F1$, Kappa, and ROC AUC scores for RF based on BIT labels.

Table 4: Random Forest results (1)

Train-test task	Sample-level metrics					Number of fixations			Fixation duration				
	Acc	Fix acc	Sac acc	F1	Kappa	ROC	AUC	Total	Mean	Stdev	%Fix	Mean (ms)	Stdev
2-1	0.870	0.940	0.546	0.920	0.518	0.743		35,358	82.04	48.72	86.37	286.85	227.83
	0.928	0.978	0.674	0.957	0.714	0.826		30,466	72.02	24.94	87.46	273.37	192.45
3-1	0.876	0.946	0.555	0.925	0.531	0.750		34,493	80.03	43.89	86.78	281.37	220.66
	0.928	0.978	0.680	0.957	0.715	0.828		30,647	72.11	25.33	87.24	273.40	194.21
4-1	0.885	0.947	0.577	0.929	0.561	0.762		33,777	78.37	39.81	86.32	280.58	216.98
	0.926	0.975	0.680	0.956	0.709	0.828		30,908	73.07	25.23	87.13	269.61	191.92
5-1	0.880	0.946	0.563	0.926	0.543	0.755		33,796	78.78	41.50	86.63	283.60	223.24
	0.926	0.974	0.680	0.955	0.707	0.827		30,592	72.67	25.81	87.05	271.12	192.71
1-2	0.873	0.939	0.558	0.922	0.525	0.749		35,022	81.26	50.15	86.30	280.24	201.98
	0.929	0.978	0.674	0.958	0.712	0.826		29,521	69.79	23.53	87.71	278.89	187.90
3-2	0.875	0.943	0.554	0.924	0.525	0.748		34,428	79.69	48.43	86.83	286.70	206.33
	0.929	0.977	0.675	0.957	0.711	0.826		29,634	69.40	24.02	87.61	279.21	188.29
4-2	0.872	0.939	0.555	0.921	0.520	0.747		34,465	79.97	47.10	86.39	285.61	208.71
	0.927	0.975	0.677	0.957	0.707	0.826		29,906	70.87	24.08	87.47	274.33	183.98
5-2	0.875	0.942	0.555	0.923	0.526	0.749		33,889	79.00	46.81	86.64	290.97	212.42
	0.927	0.974	0.677	0.956	0.705	0.826		29,468	70.16	24.33	87.41	277.25	187.10
1-3	0.873	0.942	0.557	0.922	0.529	0.749		34,217	79.39	42.78	86.36	275.52	201.35
	0.924	0.976	0.670	0.954	0.705	0.823		30,773	72.41	24.49	87.05	266.22	181.31
2-3	0.868	0.940	0.542	0.918	0.515	0.741		34,151	79.05	42.68	86.36	290.48	212.67
	0.924	0.977	0.666	0.954	0.704	0.821		30,766	72.05	24.84	87.20	267.89	183.09

Table 5: Random Forest results (2)

Train-test task	Sample-level metrics						Number of fixations				Fixation duration		
	Acc	Fix acc	Sac acc	F1	Kappa	ROC AUC	Total	Mean	Stdev	%Fix	Mean (ms)	Stdev	
4-3	BIT	0.878	0.944	0.562	0.925	0.540	0.753	33,862	78.57	40.84	86.36	278.13	206.43
	EMMV	0.924	0.974	0.676	0.954	0.705	0.825	31,157	73.14	25.09	86.83	262.92	178.65
5-3	BIT	0.880	0.945	0.565	0.926	0.545	0.755	33,358	77.76	38.38	86.35	281.56	208.38
	EMMV	0.922	0.973	0.675	0.953	0.701	0.824	30,845	73.09	25.13	86.74	263.63	180.168
1-4	BIT	0.884	0.951	0.541	0.929	0.538	0.746	31,886	73.98	40.78	87.74	300.35	223.59
	EMMV	0.925	0.979	0.641	0.956	0.687	0.810	28,467	67.30	24.84	88.46	293.13	201.85
2-4	BIT	0.868	0.941	0.514	0.918	0.487	0.728	32,576	75.58	44.55	87.42	315.30	241.34
	EMMV	0.926	0.980	0.637	0.956	0.687	0.808	28,271	66.99	25.11	88.59	295.06	203.66
3-4	BIT	0.878	0.947	0.530	0.925	0.517	0.739	32,308	74.97	41.77	87.52	307.95	234.01
	EMMV	0.926	0.978	0.645	0.956	0.690	0.812	28,795	67.59	25.34	88.31	292.13	201.35
5-4	BIT	0.885	0.951	0.546	0.930	0.542	0.748	31,327	73.02	35.93	87.40	301.28	228.48
	EMMV	0.924	0.975	0.648	0.955	0.684	0.812	28,751	67.97	25.74	88.10	289.97	200.82
1-5	BIT	0.873	0.944	0.539	0.921	0.525	0.742	32,750	76.34	40.60	86.70	285.44	227.32
	EMMV	0.919	0.977	0.634	0.952	0.678	0.806	28,894	68.63	24.78	87.82	284.30	218.65
2-5	BIT	0.867	0.943	0.520	0.918	0.500	0.731	33,042	77.02	44.02	87.01	296.56	239.51
	EMMV	0.920	0.979	0.631	0.952	0.680	0.805	28,586	68.06	24.23	88.03	286.88	221.33
3-5	BIT	0.879	0.949	0.538	0.926	0.530	0.743	32,387	75.49	40.15	87.09	289.11	235.08
	EMMV	0.919	0.977	0.636	0.951	0.678	0.806	29,028	68.79	24.79	87.72	284.27	217.97
4-5	BIT	0.883	0.950	0.554	0.928	0.548	0.752	32,024	74.65	34.56	86.82	287.04	231.23
	EMMV	0.919	0.975	0.640	0.951	0.679	0.808	29,383	69.46	25.04	87.49	280.22	214.76

The identified fixations for the first participant for task 2 trained on each of the remaining four tasks are visualised in Figure 7. The fixation centers displayed in red are calculated as the average eye location of each consecutive series of predicted fixation samples. The left panels show the results from the random forest with labels obtained from BIT, whereas the results from the random forest with EMMV labels are displayed in the right panels. The evidence in favour of transfer learning is visually supported as the different training tasks do not seem to influence both the fixation centers and the classification of samples. It can be seen that the random forest based on EMMV identifies more fixations than the random forest with BIT labels. However, the classification of samples into fixations and saccades is similar across both methods and across the different training tasks. Additionally, the sample classification between the two methods is more similar than applying the unsupervised algorithms independently.

Random Forest provides insight in how useful the different features are for correctly classifying eye movement data into events using mean decrease impurity. This measure indicates how much each feature decreases the weighted impurity in a tree. The feature importances are included in Appendix D.2.

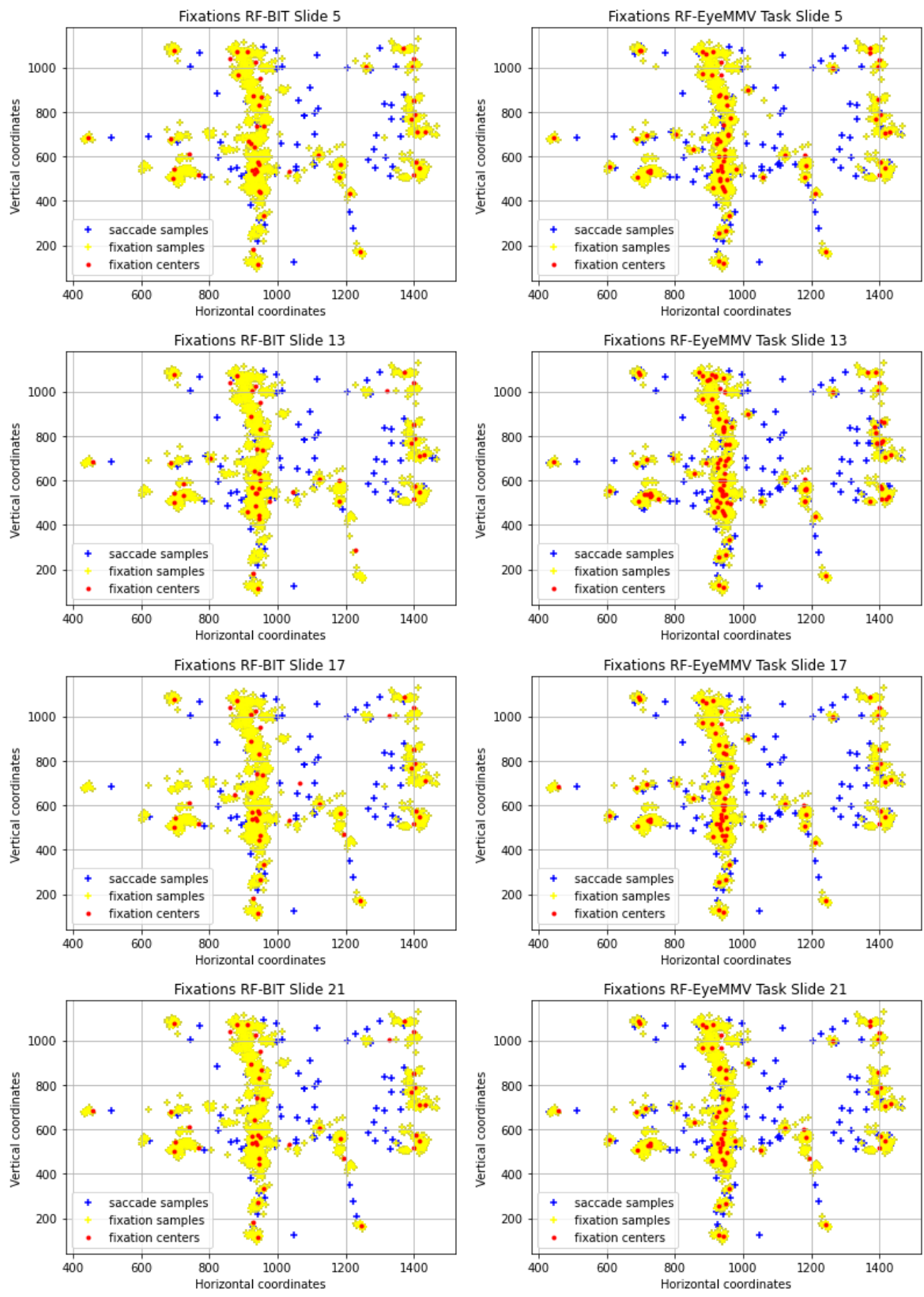


Figure 7: Examples RF-BIT (left) and RF-EMMV (right) results

5.2.2 Convolutional Neural Network

The results of CNN are presented in Table 6 and 7 in a similar manner as the RF results. Regarding the ability of transfer learning, CNN with EMMV labels shows more varying results than RF with EMMV labels. In particular, when the EMMV-based CNN is trained on task 4, less fixations with longer fixation durations are identified for task 1, 2, and 3 compared to training the model on the other tasks. Training EMMV-CNN on task 3 yields more fixations with longer durations for task 4, and less fixations with longer durations for task 5 compared to training on the other tasks. Yet, a clear distinction is again detected in the number of identified fixations and fixation duration across tasks. As with RF based on EMMV, CNN with EMMV labels also detects more fixations for task 1 and 3, and the least fixations for task 4 and 5. The average fixation duration is again the longest for task 4 (around 285 ms) and the shortest for task 4 (around 260 ms). The percentage of samples classified as fixations is the highest for task 2 and 4. Looking at the sample-level metrics, the saccade accuracy is again lower for the last two tasks. Although some predictions seem to be slightly off, in general the results seem promising for the ability of transfer learning. Both supervised methods identify the same pattern in eye movement statistics for the different tasks.

The results of CNN with BIT labels are quite consistent for each task regardless of what task is used as training data. Again, the most fixations are identified for the first three tasks, the fixation duration is longest for task 4 (around 350 ms), and shortest for task 3 (around 315 ms). This again suggests that the results are in favour of the ability of transfer learning. All four hierarchical combinations of unsupervised-supervised machine learning methods find similar patterns regarding eye movement statistics despite the task that is used for training.

Similarly to RF, the differences in eye movement statistics between BIT and EMMV are smaller after the CNN is applied relative to the results of the unsupervised learning algorithms independently in Table 2. Interestingly, CNN based on BIT identifies less fixations than CNN with EMMV labels. The mean number of fixations for BIT-CNN is 61-67 compared to 74-83 from BIT-RF and 81-83 from solely BIT. For EMMV-CNN, the mean number of fixations is 68-74 relative to the 67-73 from EMMV-RF and the 60-63 from EMMV on its own. Using EMMV labels, both supervised methods identify similar numbers of fixations. CNN corrects for too many fixations identified by BIT independently, but maybe gets carried away slightly by doing so. Looking at the mean fixation duration, something similar can be seen: both supervised methods yield similar fixation durations when EMMV labels are used (260-295 ms), whereas the supervised methods need to adjust for too short fixation durations identified by the BIT algorithm (217-225 ms). The CNN seems to exaggerate a bit (310-335 ms) compared to the RF (278-308 ms).

Table 6: Convolutional Neural Network results (1)

Train-test task	Sample-level metrics							Number of fixations				Fixation duration		
	Acc	Fix acc	Sac acc	F1	Kappa	ROC	AUC	Total	Mean	Stdev	%Fix	Mean (ms)	Stdev	
2-1	BIT	0.905	0.980	0.549	0.944	0.601	0.765	28,771	66.75	27.08	89.77	319.01	241.54	
	EMMV	0.944	0.984	0.746	0.967	0.784	0.865	31,170	73.69	25.52	86.79	264.84	192.23	
3-1	BIT	0.902	0.979	0.538	0.942	0.589	0.758	28,920	67.10	30.70	89.76	328.34	250.34	
	EMMV	0.937	0.979	0.722	0.962	0.753	0.851	29,943	70.45	26.03	86.74	281.89	209.62	
4-1	BIT	0.905	0.979	0.548	0.944	0.600	0.764	29,187	67.72	29.06	89.65	322.38	245.45	
	EMMV	0.943	0.982	0.748	0.966	0.778	0.865	31,393	74.22	25.44	86.59	263.05	190.06	
5-1	BIT	0.907	0.981	0.552	0.945	0.608	0.766	28,973	67.54	27.92	89.67	317.74	224.59	
	EMMV	0.944	0.983	0.747	0.966	0.782	0.865	30,907	73.41	25.29	86.69	265.71	191.88	
1-2	BIT	0.902	0.975	0.551	0.942	0.590	0.763	28,809	66.84	29.61	89.51	320.13	224.90	
	EMMV	0.945	0.984	0.739	0.967	0.780	0.862	30,168	71.32	23.58	87.24	270.72	185.09	
3-2	BIT	0.901	0.976	0.537	0.942	0.581	0.757	28,522	66.02	30.84	89.89	339.12	239.14	
	EMMV	0.938	0.982	0.710	0.963	0.751	0.846	28,858	67.58	24.50	87.43	288.31	203.62	
4-2	BIT	0.899	0.974	0.539	0.940	0.577	0.756	28,897	67.05	31.71	89.63	324.79	235.62	
	EMMV	0.944	0.983	0.741	0.966	0.776	0.862	30,330	71.87	23.67	87.08	268.44	182.26	
5-2	BIT	0.900	0.975	0.542	0.941	0.582	0.759	28,418	66.24	31.37	89.66	337.89	238.88	
	EMMV	0.944	0.984	0.738	0.967	0.777	0.861	29,722	70.77	23.65	87.23	272.74	185.69	
1-3	BIT	0.900	0.973	0.558	0.940	0.598	0.766	29,179	67.70	28.24	88.96	307.32	226.15	
	EMMV	0.942	0.983	0.741	0.965	0.777	0.862	31,529	74.19	24.47	86.42	257.01	178.50	
2-3	BIT	0.903	0.981	0.538	0.943	0.594	0.760	27,930	64.65	26.41	90.00	346.74	248.47	
	EMMV	0.941	0.983	0.737	0.964	0.776	0.860	31,515	73.81	24.84	86.51	258.34	181.03	

Table 7: Convolutional Neural Network results (2)

Train-test task	Sample-level metrics						Number of fixations			Fixation duration			
	Acc	Fix acc	Sac acc	F1	Kappa	ROC AUC	Total	Mean	Stdev	%Fix	Mean (ms)	Stdev	
4-3	BIT	0.903	0.977	0.552	0.942	0.599	0.764	28,883	67.01	28.74	89.25	315.94	232.78
	EMMV	0.936	0.980	0.725	0.961	0.756	0.853	30,554	71.72	24.49	86.44	267.34	191.29
5-3	BIT	0.902	0.977	0.552	0.942	0.597	0.764	28,639	66.76	26.79	89.28	314.82	233.46
	EMMV	0.940	0.982	0.740	0.964	0.773	0.861	31,298	74.17	24.94	86.39	257.51	178.55
1-4	BIT	0.902	0.978	0.528	0.942	0.581	0.753	27,218	63.15	28.70	90.24	338.46	252.43
	EMMV	0.941	0.984	0.711	0.965	0.758	0.848	29,120	68.84	25.00	87.85	283.46	196.89
2-4	BIT	0.902	0.981	0.517	0.943	0.572	0.749	26,547	61.59	25.34	90.69	348.39	255.74
	EMMV	0.941	0.984	0.708	0.965	0.756	0.846	28,984	68.68	24.92	87.88	284.36	198.94
3-4	BIT	0.899	0.976	0.511	0.940	0.559	0.743	26,965	62.564	29.71	90.22	399.49	293.93
	EMMV	0.916	0.960	0.670	0.946	0.676	0.815	30,301	71.13	38.68	86.41	296.92	220.47
5-4	BIT	0.905	0.981	0.517	0.943	0.583	0.749	26,207	61.09	25.66	90.57	355.77	264.55
	EMMV	0.940	0.984	0.708	0.964	0.754	0.846	28,952	68.44	24.62	87.88	284.67	196.49
1-5	BIT	0.898	0.976	0.532	0.939	0.580	0.754	27,354	63.76	27.65	89.51	326.70	267.45
	EMMV	0.936	0.983	0.706	0.962	0.752	0.845	29,562	70.22	24.86	87.08	274.17	214.39
2-5	BIT	0.898	0.979	0.521	0.940	0.574	0.750	26,845	62.58	25.74	89.99	339.59	275.23
	EMMV	0.937	0.984	0.705	0.962	0.752	0.844	29,310	69.79	24.80	87.18	276.86	216.02
3-5	BIT	0.899	0.977	0.519	0.939	0.574	0.748	26,866	62.62	28.20	89.82	345.48	283.15
	EMMV	0.926	0.977	0.673	0.955	0.710	0.825	28,028	66.42	26.35	87.06	362.62	246.83
4-5	BIT	0.901	0.978	0.526	0.941	0.583	0.752	27,198	63.40	27.33	89.78	338.57	279.91
	EMMV	0.933	0.980	0.699	0.959	0.739	0.840	29,341	69.36	24.30	86.94	278.00	217.60

As a result of the longer fixations, the percentage of samples classified as fixations is relatively high for BIT-CNN (around 90%) compared to the 87% of EMMV-CNN, EMMV-RF, and BIT-RF.

Noticeable about the sample-level metrics is the higher accuracy rates compared to those obtained from RF. Both the fixation and saccade accuracy are higher, implying that CNN absorbs more of the latent data structures identified by the unsupervised learning methods. This explains why the other metrics are also slightly higher for CNN compared to RF. Similarly to RF, the CNN only agrees on the saccade labels identified by the BIT algorithm for just over 50%. The metrics are again similar across the tasks for both methods.

The results for the first participant for task 2 using each of the other tasks as training for the CNN are visualised in Figure 8 in a similar manner as for the RF in Figure 7. It can be seen that BIT-based CNN identifies less fixations than CNN based on EMMV labels. The visual results suggest the CNN's ability of transfer learning since the identified fixation centers and the classification of samples are not affected by the different training tasks. Although the fixation centers are not in line across the two versions of CNN, the classification of samples into fixations and saccades is similar across both CNNs. Furthermore, the classification seems to be similar to the classification done by the RF in Figure 7. This is further elaborated in the next subsection.

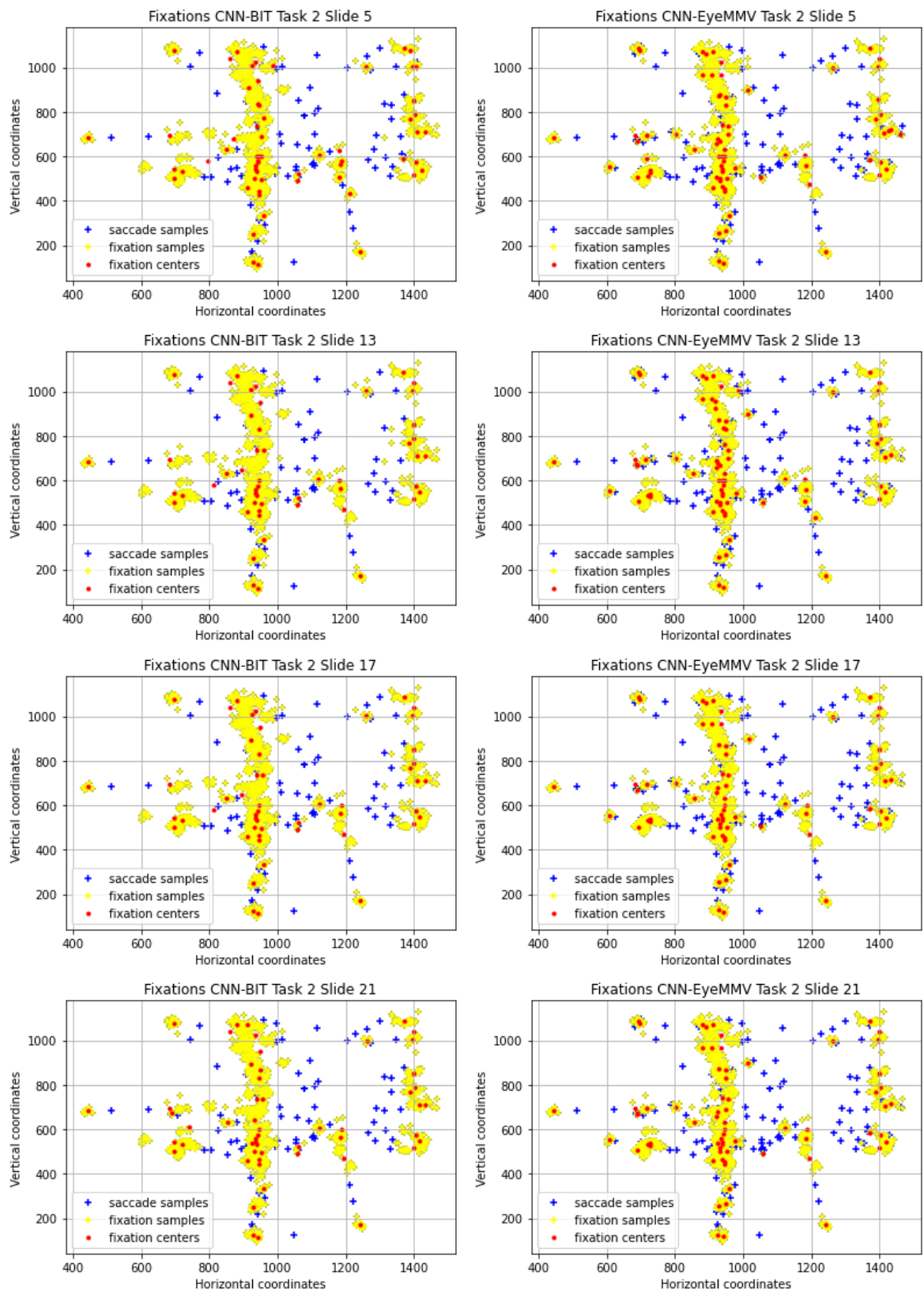


Figure 8: Examples CNN-BIT (left) and CNN-EMMV (right) results

5.2.3 Final predictions

The final predictions are defined as the event that was most frequently predicted across the four different training tasks for each sample. The predicted event of each sample obtained from the four combined unsupervised-supervised learning methods are compared to each other, and to the labels obtained from the unsupervised learning algorithms. The sample comparison results are displayed in Table 8 and 9 similarly to those of the unsupervised learning methods independently in Table 3. For every comparison between two methods, the percentage of samples that are classified as fixations or saccades by both algorithms are shown relative to the total number of samples for each task. The first observation is that the percentages vary across tasks, implying that the hierarchical combination of unsupervised and supervised machine learning algorithms accounts for the differences in tasks.

The last two columns show the comparisons of the four hierarchical combinations with the unsupervised learning methods independently. EMMV-based combinations seem to disagree more with BIT labels than the extent to which BIT-based combinations disagree with EMMV labels. All four combinations agree take over more EMMV fixations than BIT fixations. It can be seen that CNN based on BIT labels takes over the vast majority of the fixation labels identified by the BIT and the EMMV algorithm: the percentages of BIT predicting a fixation and BIT-CNN predicting a saccade are low for every task (0.78%-1.87%), and the percentages of EMMV predicting a fixation and BIT-CNN predicting a saccade are between 1.29%-2.09%. BIT-CNN seems to classify roughly half the samples that BIT labeled saccades, as fixations

In the first columns, it can be seen that the sample classification of the unsupervised-supervised learning combinations are more similar than comparing the two unsupervised learning algorithms: for the unsupervised learning methods, Table 3 shows that the diagonal elements for each task sum up to agreement of 87% for all samples, whereas the diagonal elements sum up to 92%-97%. EMMV-CNN and EMMV-RF agree the most on the sample classification for each task (97%), whereas BIT-RF and EMMV-CNN 'only' agree for 92% of the samples. To conclude, applying a supervised learning method to labels obtained from unsupervised learning techniques reduces the differences in sample classification, and that is a promising step in the development of a general applicable method to identify fixations in eye tracking data.

To conclude, all four combinations of unsupervised-supervised algorithms yield similar sample classifications, regardless of the unsupervised algorithm that was used, the supervised method that was used, and the task that was used to train the model. This implies that the proposed method of hierarchically combined unsupervised and supervised machine learning methods is promising to identify fixations in eye tracking data in a general applicable manner.

Table 8: Sample classification comparison (1)

Task 1		BIT-CNN		EMMV-RF		EMMV-CNN		BIT		EMMV	
		fix	sac	fix	sac	fix	sac	fix	sac	fix	sac
BIT-	fix	86.14%	1.91%	85.33%	3.17%	84.34%	4.15%	80.13%	8.46%	82.06%	6.52%
RF	sac	3.90%	8.05%	3.68%	7.83%	3.83%	7.68%	3.29%	8.13%	3.79%	7.63%
BIT-	fix			86.78%	3.83%	86.44%	4.17%	82.09%	8.58%	84.11%	6.56%
CNN	sac			2.23%	7.17%	1.73%	7.67%	1.32%	8.01%	1.75%	7.58%
EMMV-	fix					85.80%	2.00%	78.32%	9.74%	82.99%	5.08%
RF	sac					1.15%	11.05%	3.98%	7.96%	1.31%	10.62%
EMMV-	fix							77.94%	9.10%	83.10%	3.95%
CNN	sac							4.36%	8.59%	1.20%	11.75%
Task 2		BIT-CNN		EMMV-RF		EMMV-CNN		BIT		EMMV	
		fix	sac	fix	sac	fix	sac	fix	sac	fix	sac
BIT-	fix	86.13%	2.01%	86.08%	3.08%	85.33%	3.84%	82.24%	7.00%	82.57%	6.67%
RF	sac	4.07%	7.78%	3.01%	7.82%	3.37%	7.47%	2.75%	8.01%	3.25%	7.51%
BIT-	fix			87.01%	3.71%	86.86%	3.87%	83.64%	7.16%	84.05%	6.74%
CNN	sac			2.08%	7.19%	1.84%	7.44%	1.35%	7.85%	1.77%	7.44%
EMMV-	fix					86.45%	1.82%	80.31%	8.12%	83.49%	4.94%
RF	sac					1.17%	10.56%	3.54%	8.03%	1.25%	10.32%
EMMV-	fix							80.07%	7.69%	83.69%	4.07%
CNN	sac							3.78%	8.46%	1.04%	11.19%
Task 3		BIT-CNN		EMMV-RF		EMMV-CNN		BIT		EMMV	
		fix	sac	fix	sac	fix	sac	fix	sac	fix	sac
BIT-	fix	85.88%	2.07%	84.67%	3.35%	83.88%	4.14%	80.55%	7.54%	80.94%	7.15%
RF	sac	3.94%	8.11%	4.07%	7.92%	4.38%	7.61%	3.82%	8.09%	4.24%	7.67%
BIT-	fix			86.25%	3.84%	86.11%	3.98%	82.49%	7.66%	83.09%	7.06%
CNN	sac			2.48%	7.43%	2.14%	7.77%	1.87%	7.98%	2.09%	7.76%
EMMV-	fix					85.49%	2.06%	79.52%	8.37%	82.69%	5.20%
RF	sac					1.26%	11.20%	3.58%	8.52%	1.21%	10.89%
EMMV-	fix							79.17%	7.90%	82.82%	4.25%
CNN	sac							3.93%	9.00%	1.08%	11.85%

Table 9: Sample classification comparison (2)

Task 4		BIT-CNN		EMMV-RF		EMMV-CNN		BIT		EMMV	
		fix	sac	fix	sac	fix	sac	fix	sac	fix	sac
BIT-	fix	87.23%	1.78%	87.09%	2.33%	86.42%	3.00%	82.71%	6.661%	84.90%	4.41%
RF	sac	3.73%	7.25%	3.83%	6.75%	4.12%	6.46%	3.05%	7.64%	4.07%	6.61%
BIT-	fix			88.88%	3.21%	88.72%	3.37%	84.91%	7.14%	87.27%	4.78%
CNN	sac			2.04%	5.87%	1.81%	6.10%	0.84%	7.11%	1.70%	6.25%
EMMV-	fix					86.87%	1.97%	81.00%	9.33%	86.58%	3.75%
RF	sac					1.07%	10.09%	4.40%	5.27%	1.08%	8.59%
EMMV-	fix							80.64%	9.07%	86.62%	3.09%
CNN	sac							4.48%	5.52%	1.03%	9.24%
Task 5		BIT-CNN		EMMV-RF		EMMV-CNN		BIT		EMMV	
		fix	sac	fix	sac	fix	sac	fix	sac	fix	sac
BIT-	fix	86.67%	1.95%	86.99%	3.09%	86.25%	3.83%	82.94%	7.09%	83.68%	6.35%
RF	sac	3.74%	7.65%	2.76%	7.16%	2.92%	7.00%	1.93%	8.04%	2.67%	7.31%
BIT-	fix			87.91%	3.55%	87.71%	3.76%	84.09%	7.31%	85.05%	6.35%
CNN	sac			1.84%	6.70%	1.46%	7.08%	0.78%	7.81%	1.29%	7.31%
EMMV-	fix					86.27%	2.04%	79.00%	10.18%	84.56%	4.62%
RF	sac					1.18%	10.51%	4.75%	6.07%	0.91%	9.91%
EMMV-	fix							78.77%	9.62%	84.62%	3.77%
CNN	sac							4.98%	6.63%	0.85%	10.76%

6 Conclusion

The identification of fixations in eye movement data provides valuable information on cognitive processes that occur when viewing a stimulus. Accurate fixation classification is necessary to understand how visual stimuli are examined, and consequently, to determine what areas of visual stimuli attract the most attention. Many different algorithms have been developed which aim to distinguish fixations from saccades in eye tracking data. In the last decade, machine learning methods have become popular for this purpose. Various approaches have been presented based on both unsupervised and supervised machine learning, with both categories having its advantages and shortcomings. In this study, fixations in eye tracking data are detected by combining the superior performance of supervised learning with the ability of unsupervised learning to identify the latent data structure. I combine the unsupervised (velocity-based) Binocular-Individual Threshold and (dispersion-based) Eye Movements Metrics & Visualisations algorithms with su-

pervised Random Forest and Convolutional Neural Network because of their good performances to identify fixations in eye tracking data. However, there are unlimited possible combinations of machine learning methods to use. The hierarchical combination of individual unsupervised and supervised learning algorithms turns out to be a promising method to identify fixations in eye tracking data. The supervised method is able to learn the latent data structure that was identified by the unsupervised method, but it adjusts where needed. Applying a supervised method on eye tracking data using labels obtained by an unsupervised technique yields classification results that are more in line with each other compared to applying the unsupervised methods independently. Furthermore, all combinations detected the same differences in eye statistics across tasks that the unsupervised techniques on their own did not identify. These results were not affected by the task on which the model was trained, implying the proposed method's ability of transfer learning. To conclude, the hierarchical combination of unsupervised and supervised machine learning algorithms could be a promising step in developing a generally applicable method to identify fixations in eye tracking data regardless of the data, the unsupervised learning technique used to label the data, and parameter values that need to be set.

Future research could be focused on the incorporation of the identified fixations into marketing decision models. It would be interesting to see whether consumer choices could be predicted and/or explained by the fixations that were identified by the algorithm. This could also be used as a method to evaluate the detected fixations.

Another idea would be to adjust the supervised models to the context of eye tracking data specifically. Because of time constraints, the supervised models are quite general and easy to implement. On the other hand, the unsupervised learning methods used in this research are specific to eye tracking data.

Lastly, this research focussed on the identification of fixations and distinguished solely between fixations and saccades. In reality however, there might be more eye movement events such as post-saccadic oscillations, smooth pursuits, and blinks. It would be interesting to extend the proposed method in a multi-category manner to also identify these events, and to compare the results to the binary results obtained in this study.

There are a few limitations of this research that need to be addressed. Firstly, the proposed method should be applied to data obtained by different eye trackers with different sampling rates and noise levels to generalise the results. For example, Hessels et al. (2017) evaluate robustness against noise by adding noise points (saccades) to the data and compare the classification of fixations to the original data with a small noise amplitude. The proposed method is considered

to be robust to noise if the number of classified fixations and the corresponding distribution of fixation durations remain unchanged as noise increases. Robustness to different eye tracker sampling rates can also be tested by changing the sample rate and evaluating the algorithm's performance as in Yu et al. (2016).

Additionally, both unsupervised methods used in this research have their own process to deal with blinks, outliers and other types of noise. It would be interesting to see whether pre-processing the data yields more accurate comparisons, or whether this does not affect the classification results. Thirdly, the eye tracker signal may contain empty samples (also referred to as data loss). This study dealt with data loss by means of interpolation and discarding observations. However, there may be a better way to extract information from the data.

Furthermore, both unsupervised machine learning algorithms used in this research require the specification of a few parameters, for example the minimum fixation duration. It would be worthwhile researching if the supervised methods get affected by different parameter values. If the supervised methods yield the same fixation identification for different parameter values, the proposed method of hierarchically combining unsupervised and supervised machine learning techniques is generally applicable without having to set parameter values.

Lastly, after applying the supervised method to the labeled eye tracking data, the fixation centers are calculated as the average eye location of each consecutive series of fixation samples. This is done with the purpose to easily compare the identified fixation centers across the different methods. However, for more accurate fixation samples, the output of the supervised methods should be post-processed.

References

- W. S. Ahmed et al. The impact of filter size and number of filters on classification accuracy in cnn. In *2020 International conference on computer science and software engineering (CSASE)*, pages 88–93. IEEE, 2020.
- N. Anantrasirichai, I. D. Gilchrist, and D. R. Bull. Fixation identification for low-sample-rate mobile eye trackers. In *2016 IEEE international conference on image processing (ICIP)*, pages 3126–3130. IEEE, 2016.
- R. Andersson, L. Larsson, K. Holmqvist, M. Stridh, and M. Nyström. One algorithm to rule them all? an evaluation and discussion of ten eye movement event-detection algorithms. *Behavior research methods*, 49:616–637, 2017.
- T. J. Andrews and D. M. Coppola. Idiosyncratic characteristics of saccadic eye movements when viewing different visual environments. *Vision research*, 39(17):2947–2953, 1999.
- M. Arsenovic, S. Sladojevic, D. Stefanovic, and A. Anderla. Deep neural network ensemble architecture for eye movements classification. In *2018 17th International Symposium Infoteh-Jahorina (Infoteh)*, pages 1–4. IEEE, 2018.
- A. Bahill, A. Brockenbrough, and B. Troost. Variability and development of a normative data base for saccadic eye movements. *Investigative ophthalmology & visual science*, 21(1):116–125, 1981.
- M. E. Bellet, J. Bellet, H. Nienborg, Z. M. Hafed, and P. Berens. Human-level saccade detection performance using deep neural networks. *Journal of neurophysiology*, 121(2):646–661, 2019.
- R. Bhattarai and M. Phothisonothai. Eye-tracking based visualizations and metrics analysis for individual eye movement patterns. In *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 381–384. IEEE, 2019.
- L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001.
- E.-A. Budisteanu and I. G. Mocanu. Combining supervised and unsupervised learning algorithms for human activity recognition. *Sensors*, 21(18):6309, 2021.
- L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, et al. Api design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238*, 2013.

- M. Camilli, R. Nacchia, M. Terenzi, and F. Di Nocera. Astef: A simple tool for examining fixations. *Behavior research methods*, 40(2):373–382, 2008.
- N. V. Chawla, N. Japkowicz, and A. Kotcz. Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, 6(1):1–6, 2004.
- C. Chen, A. Liaw, L. Breiman, et al. Using random forest to learn imbalanced data. *University of California, Berkeley*, 110(1-12):24, 2004.
- V. Clay, P. König, and S. Koenig. Eye tracking in virtual reality. *Journal of eye movement research*, 12(1), 2019.
- J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.
- T. J. Crawford, A. Devereaux, S. Higham, and C. Kelly. The disengagement of visual attention in alzheimer’s disease: a longitudinal eye-tracking study. *Frontiers in aging neuroscience*, 7: 118, 2015.
- E. L. Dan, M. Dînşoreanu, and R. C. Mureşan. Accuracy of six interpolation methods applied on pupil diameter data. In *2020 IEEE international conference on automation, quality and testing, robotics (AQTR)*, pages 1–5. IEEE, 2020.
- Y. Dong, Y. Zhang, J. Yue, and Z. Hu. Comparison of random forest, random ferns and support vector machine for eye state classification. *Multimedia Tools and Applications*, 75:11763–11783, 2016.
- P. Du Jardin. Forecasting bankruptcy using biclustering and neural network-based ensembles. *Annals of Operations Research*, 299(1-2):531–566, 2021.
- R. Engbert and R. Kliegl. Microsaccades uncover the orientation of covert attention. *Vision research*, 43(9):1035–1045, 2003.
- T. Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.
- P. M. Fitts, R. E. Jones, and J. L. Milton. Eye movements of aircraft pilots during instrument-landing approaches. *Ergonomics: Major Writings*, page 56, 2004.
- W. Fuhl and E. Kasneci. Hpcgen: Hierarchical k-means clustering and level based principal components for scan path generation. In *2022 Symposium on Eye Tracking Research and Applications*, pages 1–7, 2022.

- S. Gatidis, M. Scharpf, P. Martirosian, I. Bezrukov, T. Küstner, J. Hennenlotter, S. Kruck, S. Kaufmann, C. Schraml, C. la Fougère, et al. Combined unsupervised–supervised classification of multiparametric pet/mri data: application to prostate cancer. *NMR in Biomedicine*, 28(7):914–922, 2015.
- F. Göbel and H. Martin. Unsupervised clustering of eye tracking data. In *Spatial Big Data and Machine Learning in GIScience, Workshop at GIScience 2018*, pages 25–28. Spatial Big Data, 2018.
- T. Hartmann and J. Fox. Entertainment in virtual reality and beyond: The influence of embodiment, co-location, and cognitive distancing on users’ entertainment experience. 2021.
- M. Hashemzadeh and B. A. Azar. Retinal blood vessel extraction employing effective image features and combination of supervised and unsupervised machine learning methods. *Artificial intelligence in medicine*, 95:1–15, 2019.
- H. He and Y. Ma. Imbalanced learning: foundations, algorithms, and applications. 2013.
- R. S. Hessels, D. C. Niehorster, C. Kemner, and I. T. Hooge. Noise-robust fixation detection in eye movement data: Identification by two-means clustering (i2mc). *Behavior research methods*, 49:1802–1823, 2017.
- K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van de Weijer. *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford, 2011.
- I. T. Hooge, D. C. Niehorster, M. Nyström, R. Andersson, and R. S. Hessels. Is human classification by experienced untrained observers a gold standard in fixation detection? *Behavior Research Methods*, 50:1864–1881, 2018.
- I. T. Hooge, G. A. Holleman, N. C. Haukes, and R. S. Hessels. Gaze tracking accuracy in humans: One eye is sometimes better than two. *Behavior Research Methods*, 51(6):2712–2721, 2019.
- S. Hoppe and A. Bulling. End-to-end eye movement detection using convolutional neural networks. *arXiv preprint arXiv:1609.02452*, 2016.
- M. Ippolito, J. Ferguson, and F. Jenson. Improving facies prediction by combining supervised and unsupervised learning methods. *Journal of Petroleum Science and Engineering*, 200:108300, 2021.
- D. Kahaner, C. Moler, and S. Nash. *Numerical methods and software*. Prentice-Hall, Inc., 1989.

- R. Karsh and F. W. Breitenbach. Looking at looking: The amorphous fixation measure. In *Eye movements and psychological functions*, pages 53–64. Routledge, 2021.
- K. B. Kim, H. J. Park, and D. H. Song. Combining supervised and unsupervised fuzzy learning algorithms for robust diabetes diagnosis. *Applied Sciences*, 13(1):351, 2022.
- D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman. 1d convolutional neural networks and applications: A survey. *Mechanical systems and signal processing*, 151:107398, 2021.
- K. Kornelsen and P. Coulibaly. Comparison of interpolation, statistical, and data-driven methods for imputation of missing values in a distributed soil moisture dataset. *Journal of Hydrologic Engineering*, 19(1):26–43, 2014.
- R. Kothari, Z. Yang, C. Kanan, R. Bailey, J. B. Pelz, and G. J. Diaz. Gaze-in-wild: A dataset for studying eye and head coordination in everyday activities. *Scientific reports*, 10(1):2539, 2020.
- V. Krassanakis, V. Filippakopoulou, and B. Nakos. Eyemmv toolbox: An eye movement post-analysis tool based on a two-step spatial dispersion threshold for fixation identification. *Journal of Eye Movement Research*, 7(1), 2014.
- L. Larsson, M. Nyström, R. Andersson, and M. Stridh. Detection of fixations and smooth pursuit movements in high-speed eye-tracking data. *Biomedical Signal Processing and Control*, 18:145–152, 2015.
- Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.
- G. R. Loftus and N. H. Mackworth. Cognitive determinants of fixation location during picture viewing. *Journal of Experimental Psychology: Human perception and performance*, 4(4):565, 1978.
- P. Majaranta and A. Bulling. Eye tracking and eye-based human–computer interaction. In *Advances in physiological computing*, pages 39–65. Springer, 2014.

- B. R. Manor and E. Gordon. Defining the temporal threshold for ocular fixation in free-viewing visuocognitive tasks. *Journal of neuroscience methods*, 128(1-2):85–93, 2003.
- A. Martinovici. Revealing attention-how eye movements predict brand choice and moment of choice. 2019.
- J. Meyers-Levy and D. Maheswaran. Exploring differences in males’ and females’ processing strategies. *Journal of consumer research*, 18(1):63–70, 1991.
- S. Mishra, H. K. Thakkar, P. Singh, and G. Sharma. A decisive metaheuristic attribute selector enabled combined unsupervised-supervised model for chronic disease risk assessment. *Computational Intelligence and Neuroscience*, 2022, 2022.
- M. S. Mould, D. H. Foster, K. Amano, and J. P. Oakley. A simple nonparametric method for classifying eye fixations. *Vision Research*, 57:18–25, 2012.
- M. Nyström and K. Holmqvist. An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior research methods*, 42(1):188–204, 2010.
- P. Olsson. Real-time and offline filters for eye tracking, 2007.
- T. M. Oshiro, P. S. Perez, and J. A. Baranauskas. How many trees in a random forest? In *Machine Learning and Data Mining in Pattern Recognition: 8th International Conference, MLDM 2012, Berlin, Germany, July 13-20, 2012. Proceedings 8*, pages 154–168. Springer, 2012.
- J. Otero-Millan, J. L. A. Castro, S. L. Macknik, and S. Martinez-Conde. Unsupervised clustering method to detect microsaccades. *Journal of vision*, 14(2):18–18, 2014.
- G. A. O’Driscoll and B. L. Callahan. Smooth pursuit in schizophrenia: a meta-analytic review of research since 1993. *Brain and cognition*, 68(3):359–370, 2008.
- B. Pan, H. A. Hembrooke, G. K. Gay, L. A. Granka, M. K. Feusner, and J. K. Newman. The determinants of web page viewing behavior: an eye-tracking study. In *Proceedings of the 2004 symposium on Eye tracking research & applications*, pages 147–154, 2004.
- K. Č. Pucihar and P. Coulton. Exploring the evolution of mobile augmented reality for future entertainment systems. *Computers in Entertainment (CIE)*, 11(2):1–16, 2015.
- R. Radach, A. Kennedy, and K. Rayner. *Eye movements and information processing during reading*, volume 11. Psychology Press, 2004.

- L. E. Raileanu and K. Stoffel. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41:77–93, 2004.
- K. Rayner. Eye movements in reading and information processing. *Psychological bulletin*, 85(3):618, 1978.
- K. Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372, 1998.
- K. Rayner. The 35th sir frederick bartlett lecture: Eye movements and attention in reading, scene perception, and visual search. *Quarterly journal of experimental psychology*, 62(8):1457–1506, 2009.
- K. Rayner, X. Li, C. C. Williams, K. R. Cave, and A. D. Well. Eye movements during information processing tasks: Individual differences and cultural effects. *Vision research*, 47(21):2714–2726, 2007.
- L. Rello and M. Ballesteros. Detecting readers with dyslexia using machine learning with eye tracking measures. In *Proceedings of the 12th International Web for All Conference*, pages 1–8, 2015.
- P. J. Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984.
- P. J. Rousseeuw and K. V. Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.
- D. D. Salvucci and J. R. Anderson. Tracing eye movement protocols with cognitive process models. In *Proceedings of the twentieth annual conference of the cognitive science society*, pages 923–928. Routledge, 2022.
- D. D. Salvucci and J. H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications*, pages 71–78, 2000.
- A. Savitzky and M. J. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.
- J. R. Shah and M. B. Murtaza. A neural network based clustering procedure for bankruptcy prediction. *American Business Review*, 18(2):80, 2000.

- F. Shic, B. Scassellati, and K. Chawarska. The incomplete fixation measure. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 111–114, 2008.
- H. J. Smith and M. Neff. Communication behavior in embodied virtual reality. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–12, 2018.
- C. Spearman. The proof and measurement of association between two things. 1961.
- M. Startsev and R. Zemblys. Evaluating eye movement event detection: A review of the state of the art. *Behavior Research Methods*, 55(4):1653–1714, 2023.
- M. Startsev, I. Agtzidis, and M. Dorr. 1d cnn with blstm for automated classification of fixations, saccades, and smooth pursuits. *Behavior Research Methods*, 51:556–572, 2019.
- S. Stuart, L. Alcock, A. Godfrey, S. Lord, L. Rochester, and B. Galna. Accuracy and re-test reliability of mobile eye-tracking in parkinson’s disease and older adults. *Medical engineering & physics*, 38(3):308–315, 2016.
- M. A. Tinker. Eye movement duration, pause duration, and reading time. *Psychological Review*, 35(5):385, 1928.
- A. Vabalas, E. Gowen, E. Poliakoff, and A. J. Casson. Applying machine learning to kinematic and eye movement features of a movement imitation task to predict autism diagnosis. *Scientific reports*, 10(1):1–13, 2020.
- R. Van der Lans and M. Wedel. Eye movements during search and choice. *Handbook of marketing decision models*, pages 331–359, 2017.
- R. Van der Lans, M. Wedel, and R. Pieters. Defining eye-fixation sequences across individuals and tasks: the binocular-individual threshold (bit) algorithm. *Behavior research methods*, 43: 239–257, 2011.
- K. Wang, S. Wang, and Q. Ji. Deep eye fixation map learning for calibration-free eye gaze tracking. In *Proceedings of the ninth biennial ACM symposium on eye tracking research & applications*, pages 47–55, 2016.
- M. Wedel and R. Pieters. A review of eye-tracking research in marketing. *Review of marketing research*, pages 123–147, 2017.
- Y.-S. Wu, T.-W. Lee, Q.-Z. Wu, and H.-S. Liu. An eye state recognition method for drowsiness detection. In *2010 IEEE 71st Vehicular Technology Conference*, pages 1–5. IEEE, 2010.

- Z. Wu, S. Rajendran, T. Van As, V. Badrinarayanan, and A. Rabinovich. Eyenet: A multi-task deep network for off-axis eye gaze estimation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3683–3687. IEEE, 2019.
- www.coonect.tobii.com. Tobii connect. URL https://connect.tobii.com/s/article/eye-tracker-sampling-frequency?language=en_US.
- www.tobii.com. Tobii.com. URL <https://www.tobii.com/>.
- A. L. Yarbus. *Eye movements and vision*. Springer, 2013.
- Y. Yin, C. Juan, J. Chakraborty, and M. P. McGuire. Classification of eye tracking data using a convolutional neural network. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 530–535. IEEE, 2018.
- M. Yu, Y. Lin, J. Breugelmans, X. Wang, Y. Wang, G. Gao, and X. Tang. A spatial-temporal trajectory clustering algorithm for eye fixations identification. *Intelligent Data Analysis*, 20(2):377–393, 2016.
- R. Zemblys. Eye-movement event detection meets machine learning. *BIOMEDICAL ENGINEERING 2016*, 20(1), 2017.
- R. Zemblys, D. C. Niehorster, O. Komogortsev, and K. Holmqvist. Using machine learning to detect events in eye-tracking data. *Behavior research methods*, 50:160–181, 2018.
- Y. Zhao, D. Cook, H. Hofmann, M. Majumder, and N. R. Chowdhury. Mind reading: Using an eye-tracker to see how people are looking at lineups. *International Journal of Intelligent Technologies & Applied Statistics*, 6(4), 2013.

A Choice Tasks

Table 10: Toothbrush (task 1)

Brand	Jordan	Nature	Oral-B	Preserve
Handle	Recycled plastic	Recycled wood	Plastic	
Bristles	Natural hair	Nylon		
Whitening	Yes	No		
Rubber grip	Yes	No		
Tongue cleaner	Yes	No		

Table 11: Light bulb (task 2)

Brand	Megaman	Osram	Philips	Sylvania
Bulb type	Energy saver	Halogen	LED	
Energy efficiency class	A	B		
Wattage	10W	11W	28W	42W
Voltage	220-240V			
Light output (lumens)	345lm	630lm	803lm	
Equivalent to	40 watts	52 watts	60 watts	
Colour	Warm white	Daylight		
Average lifetime	2,000 hours	15,000 hours	25,000 hours	

Table 12: Travel mug (task 3)

Brand	Aladdin	Grace	Monbento	Zuperzazial
Volume	350ml	400ml	470ml	500ml
Size	6.4x6.4x19.4	8.0x7.5x20.0	8.7x12.7x20.3	9.5x9.5x14.5
Material	Bamboo	Plastic	Thermo plastic (TP)	TP&double glass
Recycled	Yes	No		
Weight	120gr	186gr	200gr	270 gr

B Data Statistics

C Comparison Fixation Centers Unsupervised Learning Methods

Table 13: Television (task 4)

Brand	Panasonic	Philips	Samsung	Sony
Energy efficiency	A	A ⁺⁺	A ⁺⁺⁺	
Power consumption	41W	48W	57W	58W
Electricity consumption	60kWh	70kWh	83kWh	85kWh
Screen size	40 inch	42 inch		
Image quality	Full HD			
Image resolution	1920x1080			
Image motion rate	200Hz	300Hz	400Hz	500Hz
Audio power	20W RMS	24W RMS		
Dimensions	95.7x61.9x21.7	95.7x63.5x29.4	97.2x64x26.5	97.7x62.9x24

Table 14: Fridge (task 5)

Brand	Bosch	Samsung	Siemens	Whirlpool
Energy efficiency	A ⁺	A ⁺⁺	A ⁺⁺⁺	
Electricity consumption	172kWh	204kWh	293kWh	308kWh
Cooling space (litres)	225	234	245	260
Freezer space (litres)	85	98	113	
Fast chill option	Yes	No		
Freezer/fridge	2/2	2/3		
Freezer position	bottom			
Dimensions (WxHxD)	59.5x178x66.8	59.5x187.5x64	60x185x65	60x201x65
Weight (kg)	63	79	94	101

Table 15: Data statistics after cleaning

	#participants	Total #samples	Mean #samples	Min #samples	Max #samples
Task 1	431	568,248	1,318	330	1,804
Task 2	432	563,328	1,304	292	1,805
Task 3	433	562,825	1,300	296	1,805
Task 4	431	557,642	1,294	302	1,805
Task 5	429	554,726	1,293	301	1,804

D Random Forest

D.1 Grid search

D.2 Feature Importance

Table 16: Comparison fixation locations BIT and EMMV

	Within 1 pixel	Within 5 pixels
Task 1	1827	1853
Task 2	2408	2299
Task 3	1687	1860
Task 4	151	695
Task 5	249	720

Table 17: Grid search results BIT & EMMV

Hyperparameter	Grid
<i>n_estimators</i>	{50, 100, 150, 200 }
<i>max_features</i>	{1, 2, 3 , 4, 5, 6}
<i>max_depth</i>	{10, 15 , 20, 25}
<i>min_samples_split</i>	{2, 4 , 6, 8, 10}
<i>min_samples_leave</i>	{ 1 , 2, 3, 4, 5}
<i>bootstrap</i>	{True, False}

Table 18: Mean feature importance (1)

Train-test	task	<i>velocity</i>	<i>dispersion</i>	<i>stdev</i>	<i>stddif</i>	<i>meandif</i>	<i>mediandif</i>	<i>rayleigh</i>
2-1	BIT	0.318	0.162	0.166	0.085	0.074	0.098	0.096
	EMMV	0.301	0.138	0.186	0.112	0.080	0.120	0.062
3-1	BIT	0.323	0.158	0.163	0.086	0.075	0.100	0.095
	EMMV	0.301	0.139	0.183	0.112	0.083	0.118	0.062
4-1	BIT	0.308	0.164	0.170	0.088	0.078	0.102	0.090
	EMMV	0.284	0.144	0.186	0.113	0.085	0.123	0.064
5-1	BIT	0.305	0.164	0.166	0.090	0.082	0.101	0.091
	EMMV	0.277	0.143	0.187	0.115	0.088	0.126	0.065
1-2	BIT	0.318	0.161	0.169	0.084	0.075	0.100	0.092
	EMMV	0.299	0.144	0.189	0.109	0.080	0.119	0.0602
3-2	BIT	0.323	0.158	0.163	0.086	0.074	0.100	0.095
	EMMV	0.300	0.140	0.183	0.112	0.083	0.118	0.063
4-2	BIT	0.308	0.164	0.170	0.088	0.078	0.102	0.090
	EMMV	0.284	0.144	0.187	0.113	0.085	0.123	0.064
5-2	BIT	0.305	0.164	0.166	0.090	0.082	0.101	0.091
	EMMV	0.277	0.143	0.187	0.114	0.088	0.126	0.065
1-3	BIT	0.318	0.161	0.169	0.084	0.075	0.100	0.092
	EMMV	0.299	0.144	0.189	0.109	0.080	0.119	0.060
2-3	BIT	0.318	0.162	0.166	0.086	0.074	0.098	0.096
	EMMV	0.301	0.139	0.187	0.112	0.080	0.120	0.062
4-3	BIT	0.308	0.164	0.170	0.088	0.078	0.102	0.090
	EMMV	0.284	0.145	0.187	0.113	0.085	0.123	0.064
5-3	BIT	0.305	0.164	0.166	0.090	0.082	0.101	0.091
	EMMV	0.277	0.143	0.187	0.114	0.088	0.126	0.065
1-4	BIT	0.318	0.161	0.169	0.084	0.075	0.100	0.092
	EMMV	0.299	0.144	0.189	0.109	0.080	0.120	0.060
2-4	BIT	0.318	0.162	0.166	0.085	0.074	0.098	0.096
	EMMV	0.302	0.138	0.186	0.112	0.080	0.120	0.062
3-4	BIT	0.323	0.158	0.163	0.086	0.075	0.100	0.095
	EMMV	0.301	0.140	0.183	0.112	0.083	0.119	0.062
5-4	BIT	0.305	0.164	0.166	0.090	0.082	0.101	0.091
	EMMV	0.277	0.143	0.187	0.114	0.088	0.126	0.065

Table 19: Mean feature importance (2)

Train-test	task	<i>velocity</i>	<i>dispersion</i>	<i>stdev</i>	<i>stddif</i>	<i>meandif</i>	<i>mediandif</i>	<i>rayleigh</i>
1-5	BIT	0.318	0.161	0.169	0.084	0.075	0.100	0.092
	EMMV	0.299	0.144	0.189	0.109	0.080	0.119	0.060
2-5	BIT	0.318	0.162	0.166	0.085	0.075	0.098	0.097
	EMMV	0.302	0.138	0.186	0.112	0.080	0.120	0.062
3-5	BIT	0.323	0.158	0.163	0.086	0.075	0.100	0.095
	EMMV	0.301	0.140	0.183	0.112	0.083	0.118	0.063
4-5	BIT	0.308	0.165	0.170	0.088	0.078	0.102	0.090
	EMMV	0.284	0.144	0.186	0.113	0.085	0.123	0.064