Transfer of Gaze Classifiers: Towards a Distance Metric for Eye Tracking Data

Elena Gimmel
Faculty of Technology
University of Applied Sciences Emden/Leer
Emden, Germany
elena.gimmel@stud.hs-emden-leer.de

Carolin Hainke
Faculty of Technology
University of Applied Sciences Emden/ Leer
Emden, Germany
carolin.hainke@hs-emden-leer.de

Sven Janßen
Faculty of Technology
University of Applied Sciences Emden/Leer
Emden, Germany
sven.janssen@hs-emden-leer.de

Thies Pfeiffer
Faculty of Technology
University of Applied Sciences Emden/Leer
Emden, Germany
thies.pfeiffer@hs-emden-leer.de

Abstract

Eye tracking is applied in very heterogeneous settings, from reading to immersive virtual reality (VR), using devices ranging from stationary high-precision to mobile low-power devices. Classification algorithms, on the other hand, are developed and tested on specific devices in specific settings. A central question is how algorithms that have been optimized for a particular scenario (device + setting) would perform when being applied to a different scenario. In this paper, we approach the idea of a distance metric, describing the similarity between two scenarios. If the distance between two scenarios is low, the algorithms can be safely adopted between them. The higher the distance, the weaker the expected performance of the transferred algorithm. We reflect our ideas on a transfer from a 2D screen-based eye-tracking scenario to an immersive VR scenario.

CCS Concepts

• **Human-centered computing → Interaction techniques**; **Pointing devices**; **Accessibility technologies**.

Keywords

eye tracking, event classification

ACM Reference Format:
Elena Gimmel, Carolin Hainke, Sven Janßen, and Thies Pfeiffer. 2025. Transfer of Gaze Classifiers: Towards a Distance Metric for Eye Tracking Data. In *2025 Symposium on Eye Tracking Research and Applications (ETRA '25), May 26–29, 2025, Tokyo, Japan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3715669.3725896

1 Introduction

In this paper, we are taking first steps to address a problem that researchers and engineers have when trying to implement gaze-based interaction or accessibility technologies in non-standard scenarios: To apply modern, machine-learning (ML)-based algorithms, they would need extensive training data. This would be expensive to generate and sometimes, due to small target populations in particular in the domain of accessibility, would be difficult to get. Alternatives are taking a pre-trained algorithm or finding a published dataset of a similar scenario as a basis for training. However, in both cases, it is unclear, whether the original scenario is similar enough so that the performance of the algorithm will be satisfying.

We thus discuss the idea of a decision framework. Such a framework would provide a list of tools helping with the decision, whether a dataset for a certain scenario could be used for training a classifier that should later be applied to a different scenario. One first tool of such a decision framework would be a list of exclusion criteria, that can be applied and measured to decide, whether a dataset would be eligible. In a second step, a distance metric could provide an estimation of the practical difference between two scenarios based on their characteristic parameters. With such a metric, one can make an informed selection from available datasets or algorithms with proven performance in a specific scenario. In follow-up steps, it might even be possible to develop an estimator which predicts the performance on a new dataset just based on these parameters.

In our own research, we are facing the described challenges quite often, as we are targeting real-time interaction in virtual reality (VR) or mixed reality, thus studying gaze behavior in immersive 3D environments. Immersive gaze analysis offers insights previously unattainable in 2D settings and has broadened applications from psychological studies to consumer analysis [Adhanom et al. 2023].

Most research on gaze analysis, however, is focusing on 2D gaze tracking, which typically involves fixed user positions and limited stimuli (static images, videos). This often leads to a simplified data analysis based on 2D screen coordinates. In contrast, 3D scenarios, such as VR, introduce dynamic elements where both stimuli and users move. This expands the field of view and complicates gaze mapping due to additional spatial dimensions and environmental changes. User movement in 3D environments adds noise and variability that differs from the patterns observed in stationary 2D data [Lamb et al. 2022]. For example, a user following a moving point in a 2D scenario would result in observable smooth pursuit, while a moving user observing the same movements in VR would

ETRA '25, Tokyo, Japan

ACM ISBN 979-8-4007-1487-0/25/05
https://doi.org/10.1145/3715669.3725896

result in more complex gaze behavior, often referred to as a combination of smooth pursuit and vestibulo-ocular reflex (VOR) [Purves et al. 2015].

Varying scenarios also produce different gaze event distributions, and machine learning algorithms trained on 2D datasets (without VOR) may struggle to classify fixations and saccades in 3D datasets (with VOR). Consequently, established classification algorithms might underperform on 3D datasets. Researchers and engineers exploring eye tracking in new scenarios consequently face challenges in algorithm selection and machine learning applicability. They must either invest in recording and annotating substantial data or identify the most suitable existing dataset or algorithm.

In the following, we review related work, identify parameters, and classify datasets according to these parameters. We then use an example-based approach to assess the influence of these parameters on the performance of a machine learning algorithm transferred from a 2D screen-based scenario to an immersive 3D scenario. Based on these findings, we conclude with a first estimation on the relevance of parameters and thus on a first choice set of parameters for a distance metric between eye tracking datasets.

2 Related Work

First step is the identification of parameters characteristic for a scenario: The **frame rate** at which eyes are recorded varies between devices (e.g. HoloLens 2 [Microsoft 2023]: 30 Hz, Eye Link 1000 Plus [Ltd. [n. d.]]: 2000 Hz). The **field of view** covered by the stimuli varies between scenarios (e.g. a computer screen [Griffith et al. 2021] or VR glasses [Wei et al. 2023]) and can be specified in angles [Magic Leap [n. d.]], millimeters [Griffith et al. 2021] or pixels [Wei et al. 2023]. Gaze location events are given in different **units**, from pixels [Agtzidis et al. 2019] over angles [Griffith et al. 2021] and angular velocities [Kothari et al. 2020] to millimeters. Stimuli are covering different **dimensions**, typically 2D for screen-based scenarios or 3D for mobile or immersive scenarios. The **distance** between users and stimuli can be fixed or dynamic. The **setup** for eye tracking sessions can range from fixed chin-rest positions [Andersson and Larsson 2017] to free head movement in space [Kothari et al. 2020]. **Stimuli** can be dots [Andersson and Larsson 2017], texts, images, videos, complex 3D structures, animated objects [Kim et al. 2019], or articulated figures. A **task** may be given, such as reading, free viewing [Griffith et al. 2021], or performing an action [Kothari et al. 2020]. The data can have **labels** from basic fixations and saccades to such used for biometrics [George and Routray 2016], or of a finer granularity, such as in the dataset 360_em [Agtzidis et al. 2019]. Table 1 reflects the diversity of data on the example of six labeled datasets available to train and test gaze classifiers.

In our own research, we are targeting gaze classification in immersive environments. We thus concentrate on the question of classifying gaze in 3D scenarios in the following.

Recent research has explored approaches to classify eye movements in 3D eye-tracking datasets, collected particularly in virtual reality (VR) settings where both stimuli and users are mobile. With CLRGaze [Bautista and Naval 2021], a novel method for learning feature vectors of eye movements using contrastive learning was presented, that can then be used by neural networks to classify the eye-tracking data. This approach shows strong potential for handling diverse and complex datasets, such as those collected in dynamic 3D VR environments. By leveraging self-supervised learning, the method reduces the reliance on extensive labeled data while enabling effective feature representation. However, the research is restricted to eye movements obtained from viewing static images and the authors already apply several transformations to homogenize the data without an analysis of the effects of the transformations (downsampling, coordinate system transformations).

Another notable study is that of Agtzidis et al. (2019) [Agtzidis et al. 2019], who recorded and partially annotated a new dataset of eye-tracking data in 360-degree videos. They devised a two-step process for the manual annotation of eye movements, including fixations and saccades, and implemented an algorithm for the automatic classification of these movements. This study highlights the need to adapt existing classification methods to the more complex conditions of 3D environments.

Llanes-Jurade et al. (2020) [Llanes-Jurado et al. 2020] not only addressed the problem of the complexity of 3D datasets but also highlight the challenge of identifying the right parameters for the ML algorithms used to classify those datasets. They propose a dispersion-threshold identification algorithm capable of handling eye-tracking data that is recorded while the user is free to move. Additionally, the authors provide parameters for the algorithm that achieve optimal fixation detection in a dataset recorded in a VR environment.

3 Towards a Distance Metric for Eye Tracking following an Example Case

The distance metric should help to assess the similarity of scenarios based on parameters. It aims to predict machine learning performance when training and application scenarios differ. The metric should therefore take into account parameter differences and their weighted influence on event detection. This paper provides initial indications of the significance of some parameters. Results are preliminary and the weighting and significance of the parameters for the distance metric may change with further research that includes more examples of datasets and transformations.

In our example, we are considering the problem of deriving a working gaze classifier for immersive VR scenarios from an annotated dataset created in a 2D scenario. We approach this by selecting two available datasets and analyze the process of transferring a classifier trained on the first to predict outcomes on the latter. In this stage of the research, we need the second dataset for verification, in the end, the metric should allow us to predict the outcome just based on the parameters of the target scenario. The parameter values of "Dimension of use" are thus "2D" and "3D VR", respectively. In order to find out which of the parameters listed in chapter 2 are decisive in determining whether the dataset is suitable for training a machine learning model for this use case, "Field of View", "Label" distribution and the type of "Stimuli" are analyzed in a first step. The parameter "Frame Rate" is covered in a limited way.

The parameters “Unit” and “Distance to User” are equalized, to allow a machine learning algorithm to be trained on one dataset and tested on the other. The labels were also aligned for the same

Table 1: Parameter of the datasets; Datasets: 1. 360_em [Agtzidis et al. 2019], 2. GazeBase [Griffith et al. 2021], 3. NVGaze [Kim et al. 2019][Fuhl et al. 2021], 4. Andersson et al. [Andersson and Larsson 2017], 5. GazeInWild [Kothari et al. 2020] [Fuhl et al. 2021], 6. Labelled Pupils in the wild [Tonsen et al. 2016][Fuhl et al. 2021]; Label: Fix - Fixation, Sac - Saccade, SP - Smooth Pursuit, PSO - Post-Saccadic Oscillation, GF - Gaze Fixation, GP - Gaze Pursuit, GS - Gaze Shift, Blk - Blink, Undef - Undefined, Err - Error, N - Noise, Unass - Unassigned, HP - Head Persuit, VOR - Vestibulo-ocular reflex, OKN - Optokinetic nystagmus

Dataset	Frame Rate	Field of View	Unit	Dimension of use	Setup & Distance to User	Stimuli & Task	Labeling
1.	120 Hz	1440 px x 1280 px	Coordinates on 360° video surface	3D VR	Free Head Movement	Free viewing of videos	Fix, Sac, SP, HP, VOR, OKN, N, Unass
2.	1000 Hz	474 mm x 297 mm/ 1680 px x 1050 px	Degrees of visual angle	2D	Free Head Movement; Distance: 550 mm	fixation, horizontal and random oblique saccade tasks, reading, free video viewing, gaze-driven gaming	Fix, Sac, Blk
3.	120 Hz	Monitor: 27" inches	Gaze vector	2D & 3D VR	Monitor: Fixed Head Position, 530 mm distance	Rotating letter "E", changing contrast	Fix, Sac, SP, Blk, Err
4.	500 Hz	380 mm x 30 mm/ 1680 px x 1024 px	Pixel	2D	Fixed Head Position, Chin and Forehead Rest, 670 mm distance	Free viewing of images, following moving objects in videos, moving dots	Fix, Sac, PSO, SP, Blk, Undef
5.	120 Hz	-	Rotational velocities (deg/s), gaze vector	3D	Free Movement	Indoor navigation, Ball catching, Object search, Tea making	GF, GP, GS
6.	120 Hz	-	Gaze vector	3D	Free Head Movement	Following moving red ball	Fix, Sac, SP, Blk, Err

reason. The influence of these parameters on the distance metric must be investigated in further research with different datasets.

3.1 Datasets

For the 2D dataset Andersson et al. [Andersson and Larsson 2017] was used. The set is further referred to as the Andersson et al. set. Data were captured in 500 Hz using a static eye tracker with a chin and a forehead rest, so the participant's head was fixed. They were supposed to observe the content on a screen freely, including images, moving dots and videos. As 3D data, the 360_em dataset [Agtzidis et al. 2019] was chosen, which has been mentioned earlier, because of the similar stimuli used (video) compared to Andersson et al.. These data were recorded by an eye tracker integrated into a VR headset, while participants watched a 360° video and were asked to move their head and look around freely. The gaze positions in the data are given as coordinates on the video sphere.

The distribution of event labels in the 360_em dataset can be seen in the left diagram of Fig. 1. To stay close to the target use case, first a subset of Andersson et al. with annotated videos was chosen. However, this subset contained an imbalance towards smooth pursuit, as shown in the central diagram of Fig. 1. This was counteracted by adding some data from the image viewing subset of Andersson et al. until a comparable distribution was reached, later referred to as "like_360_em". This data subset "like_360_em" was used for the following comparisons. The distribution of gaze in the field of view is different between the two datasets. In Andersson et al. a computer screen was used, which limits the field of view to 31.66° x 25.24°. The headset used in 360_em has a field of view of 86° x 92° [Sauer et al. 2022], which allows for a larger view compared to Andersson et al. The distribution of the view is therefore wider in 360_em, which can be seen in Fig. 3. The horizontal deflection of the eye also has a different shape in the distribution, with the gaze being concentrated in the center at 360_em. A possible explanation could be the enforced fixed head in Andersson et al., preventing users to follow their eye movements with their head to keep eyes in close-to-center positions. To further compare the datasets, the unit has to be unified to angular velocity, which will be discussed later in this work. The comparison of the velocity distribution in Fig. 2 shows that, with the exception of Smooth Pursuit, the velocities in 360_em are more widely distributed. It is striking that in 360_em the distribution of velocities found during smooth pursuit is very similar to the distribution found during fixations.

3.2 Parameters that may be aligned between datasets

Parameter Labeling: The event labels used in the dataset are highly relevant. If the training set does not contain all the labels that the application needs to detect, the dataset cannot be used. Consequently, having the required event labels is an exclusion criteria of the decision framework. If a dataset covers more labeled event types than required, they can be used with any labels not required in the target scenario collected in one "Other" class, or merged if arguable. For the selected data, labels were merged as follows: Anderson: "PSO" => "Fixation" and "Blink" or "Undefined" => "Other"; 360_em: "Noise" => "Other". However, it remains to be tested, whether it is better to merge close labels or to collect them in one rejection class. The labels for our case are thus "Fixations", "Saccades", "Smooth Pursuit" and "Other".

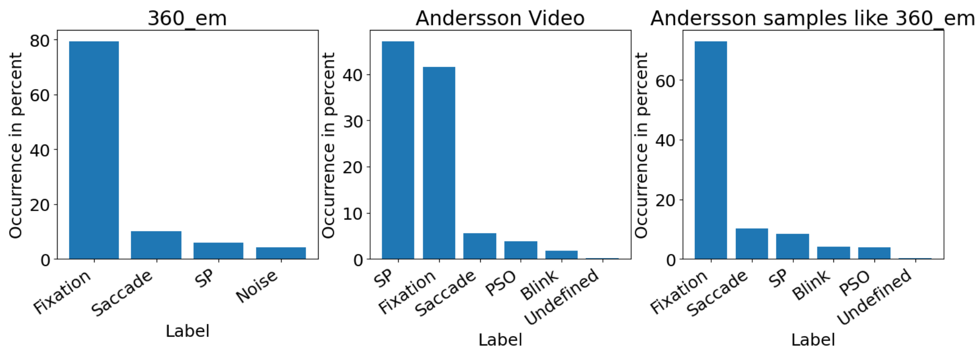


Figure 1: Event distributions for 360_em, Andersson et al. video samples, and the "like_360_em" subset of Andersson et al.

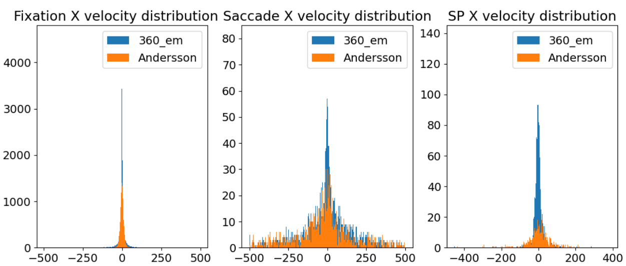


Figure 2: X axis velocity distribution for 360_em and the Andersson et al. "like_360_em" subset

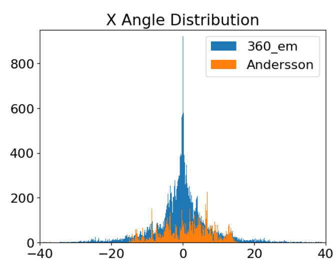


Figure 3: Angle distribution of the X component of 360_em and the Andersson et al. "like_360_em" subset

Parameter Unit: A common unit to represent movement information is required. The often reported pixel dimensions are not very suitable for this, as interpretation depends on many factors, such as screen resolution, display dimensions, field of view and distance to the user. For this reason, angular velocity (angles in degrees per second) was chosen as unit in this paper because it is independent of these parameters. Converting the Andersson et al. set to angular velocity meant to convert the pixel positions into millimeters, constructing position vectors based on these coordinates as well as the distance to the user and calculating the angles between consecutive vectors. Finally, the angular velocity was calculated in degrees per second based on the frame rate. Converting the 360_em [Agtzidis et al. 2019] set meant to port the original code from Matlab to Python. In this code directional vectors were created for the positions on the video sphere and head movements were compensated by adjusting the directional vectors accordingly. Afterwards, the angle between two consecutive vectors was calculated, and angular velocity was determined on the basis of the frame rate. However, if a dataset is using a unit such as pixels but does not provide the required information to transform the unit into angular velocity, the dataset cannot be used. Thus incomplete documentation of a dataset also is an exclusion criterium.

Parameter Frame Rate: Having the same frame rate between the source and target scenario is decisive. For fixing a difference, resampling one dataset is an option. The two datasets used here have different frame rates, so the frame rate of the Andersson et al. set was downsampled from 500 Hz to 120 Hz, to match the frame rate of the 360_em dataset. We evaluated seven different approaches for downsampling, which are listed in the table 2. Two different application orders were also compared. In the first, the gaze positions were interpolated first and then the velocity was calculated. In the second, it was the other way around, so the velocity was interpolated. To determine the similarity of the interpolated data to the original, the velocity distribution was compared using cosine similarity. The results in table 2 show that the interpolation of the velocity in combination with the Samplerate package with the converter "zero_order_hold" leads to the most similar results compared to the original. For the reassignment of the label to the resampled data two different methods were tested. In the first method the label was assigned using the label closest in time to the original data (close in time). In the second method the label was assigned using the label with the closest velocity (close in value). For this purpose, the velocities of the original samples, from which the new resampled sample was created, were compared with the velocity of the new sample. The label of the sample with the most similar velocity was assigned. To determine which label assignment method comes closest to the original, data was split by events and the distribution of velocities for each event label were compared again with the original data using cosine similarity. Table 3 shows the results, from which it can be seen that assigning the label by similar velocity resulted in a velocity distribution more similar to the original.

3.3 Parameters with an effect on performance

For a first investigation regarding the effect of parameter differences on performance, we selected a CNN with the structure of

Table 2: Interpolation packages, classes and functions with cosine similarity results

Name	Description	Position Interpolation	Velocity Interpolation
Signal.resample [community 2025d]	Part of scipy, uses the Fourier method	0.9545	0.9762
Samplerate (zero_ order_ hold) [Wagner 2017]	Resampler with a zero order hold converter of the Samplerate package	0.9461	0.9994
Samplerate (linear) [Wagner 2017]	Resampler with a linear interpolation of the Samplerate package	0.9392	0.9957
Interp [Developers 2024]	Part of numpy, uses linear interpolation	0.942	0.9416
CubicSpline [community 2025b]	Part of scipy, interpolates with a piecewise cubic polynomial	0.9476	0.9993
PchipInterpolator [community 2025c]	PCHIP = Piecewise Cubic Hermite Interpolating Polynomial; Part of scipy, uses a monotonic cubic interpolation	0.9451	0.9978
Akima1DInterpolator [community 2025a]	Part of scipy, uses a sub-spline that is continuously differentiable	0.9455	0.9979

Table 3: Label assignment using original label close in time or close in value

Label	close in time	close in value
Fixation	0.997	0.998
Saccade	0.752	0.775
SP	0.994	0.994
Other	0.989	0.991

Biwaro and Kasprowski [Birawo and Kasprowski 2022]. This algorithm had already proven itself in the corresponding paper in comparison with other algorithms. "The network is composed of different layers, precisely three convolutional layers with a gradually increasing number of filters (32, 64 and 128) with a kernel size of 3, a batch normalization operation before activation and an output layer. Input to the network is a sequence of gaze samples of shape 100 x 2." [Birawo and Kasprowski 2022]. Like in Biwaro and Kasprowski [Birawo and Kasprowski 2022] sequences of the two dimensional samples are used as input. To be able to use the datasets with the CNN, the data must be unified in the same unit and with the same frame rate as described earlier in this chapter.

First, a CNN was trained with positive component velocities (absolute values) from the dataset "like_360_em" of Andersson et al.. Then a second was trained with positive and negative velocities on the same data. The velocities were negative when a backward movement took place. The CNNs were tested on the same dataset as trained (different subsets as usual). Comparing the confusion matrices of the two CNNs (Fig.. 4 and Fig. 5) it can be seen that the event detection for each label has improved or remained the same, except for Smooth Pursuit. The average event detection increased from 51% to 56%.

After compensating the head movement from the 360_em dataset, it becomes clear that SP is difficult to impossible to detect. The confusion matrix of the CNN trained on angular velocity showed a detection rate of 98% for fixations, 75% for saccades and 82% for noise, but 0% for smooth pursuit on the same dataset. This is due to a low occurrence of SP of 6.02%, with 46% overlapping with vestibulo-ocular reflex (VOR) and 17% overlapping with VOR and optokinetic nystagmus (OKN). If these are canceled out by removing head movements from the angular velocities, only 2.22% of SP labels

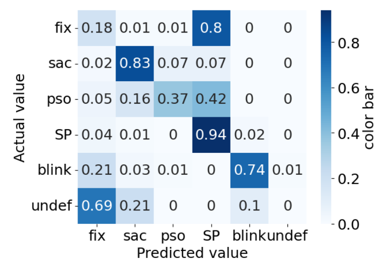


Figure 4: Confusion matrix of the CNN trained on "like_360_em" with only positive component angle velocity

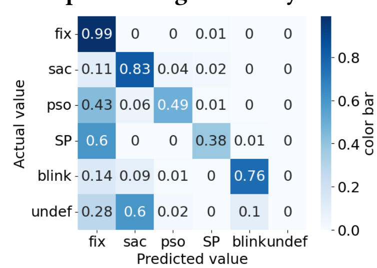


Figure 5: Confusion matrix of the CNN trained on "like_360_em" with positive and negative angle velocity

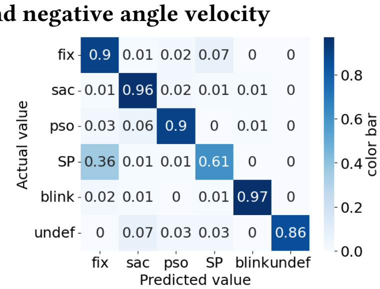


Figure 6: Confusion Matrix of a CNN trained on even balanced amount of the labels Fixation, Saccade and Smooth Pursuit

remain to train the CNN, which could explain the low detection, but further research is needed to make a general statement.

Parameter Label: To achieve better results, the input data was adjusted. The number of event labels to be detected, i.e. fixation, saccade and SP, was adjusted using undersampling. To preserve the natural gaze flow, the sequences were undersampled instead of the samples themselves. As it can be seen in Fig. 6 in comparison to Fig. 5, the average detection rate has increased to an average event detection of 86,67% for the same dataset. This result shows that the label distribution has a strong influence and should be weighted more heavily in the distance metric.

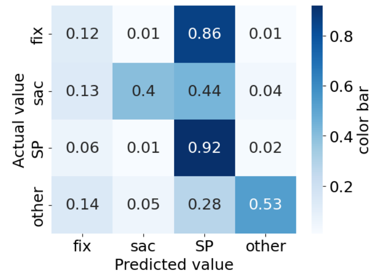
Parameter Field of View: To exclude the possibility that the different fields of view of the datasets is a factor in the detection of the events, the samples of 360_em were gradually restricted to different fields of view according to their position. The training was then performed on the angle-filtered data and tested on the complete dataset. As can be seen in table 4, fixations were still detected at a similar percentage. The detection of saccades and SP decreases slightly, but this is related to the lower occurrence. Noise was no longer detected, as this was almost completely sorted out by the angle filter. This indicates that the field of view was not a limiting factor in detection of the events with these two datasets. Therefore, the parameter will probably have a very low weight in the metric.

Parameter Stimuli & Label Distribution: The CNN model trained on Andersson et al. was finally tested on the 360_em dataset. Three different sample datasets were used for this. The first contained only samples taken from a similar stimulus (videos), the second contained samples selected to match the distributions of events found in 360_em (image and video stimuli). The last one was created by taking samples from video, image and moving dot data in such a way, that event labels are balanced. The confusion matrices of these tests are shown in Fig. 7. It shows that evenly distributed events (c) produce good results when targeting datasets with unknown distributions, even if the samples used were recorded showing different stimuli. If event frequencies of the target set are known, matching this distribution (b) might be considered if there are prominent events for which classification results should be high. Regarding the decision framework this provides further insights: it does not only matter that the relevant event labels are present in the dataset, they should also have a relevant frequency. They do, however, not need to show a similar distribution, as this can be compensated for, as demonstrated above. For the compensation to work properly, however, relevant events should not be significantly underrepresented.

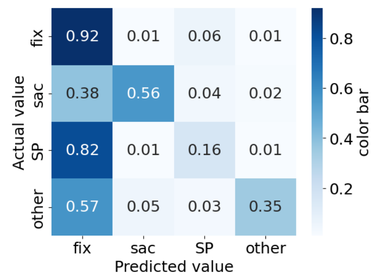
Parameter Setup & Dimension of use: The extent to which the setup or even the choice of eye tracker has an influence on the distance metric can not yet be determined. Several scenarios need to be compared for this. The impact of different "Dimension of use" must also be determined by comparing several datasets.

4 Summary

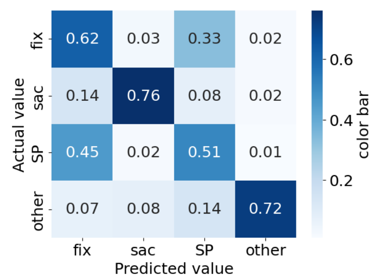
In this paper, the problem of selecting the right training dataset to train a classifier for eye movements is discussed, and first steps towards a decision framework and a distance metric to compare



(a) CNN trained on Andersson et al. with video samples



(b) CNN trained on Andersson et al. with sample set distributed like 360_em



(c) CNN trained on Andersson et al. with balanced samples

Figure 7: CNN trained on different Andersson et al. sets and tested on 360_em

datasets were taken. Parameters were introduced to assess the difference between datasets. A use case was presented to discuss the effects of different transformation strategies on data quality or classification results. Following this strategy, we could either identify a valid transformation (e.g. for frame rate downsampling between 500 Hz and 120 Hz) – rendering the parameter practically irrelevant –, show that differences in parameters did not matter (e.g. field of view), suggest a neutral way of specification (e.g. unit/angular velocities) or show that a parameter is highly relevant (e.g. distribution of relevant event labels). This shows which parameters should be weighted as strongly in the distance metric. Two exclusion criteria have also been identified: a dataset needs to provide the required event labels with a relevant frequency to be eligible and if units are not reported in a scenario agnostic format (such as angular velocities), the datasets documentation needs to provide relevant information regarding the setup (e.g. distance of the user, size of

Table 4: Angle filtered event distribution with detection results

Angle	Fixation occurrence	Saccade occurrence	SP occurrence	Noise occurrence	Fixation detection	Saccade detection	SP detection	Noise detection
FoV 86° x 92°	79.39%	10.33%	6.06%	4.21%	97%	89%	8%	79%
1/3 FoV 28.67° x 30.67°	86.14%	8.59%	5.26%	0.01%	96%	81%	5%	0%
1/4 FoV 21.5° x 23°	87.71%	7.17%	5.10%	0.02%	94%	79%	8%	0%
1/5 FoV 17.2° x 18.4°	90.21%	6.52%	3.25%	0.03%	95%	80%	8%	0%
1/6 FoV 14.33° x 15.33°	91.99%	5.64%	2.36%	0.01%	97%	79%	1%	0%
1/6 FoV 14.33° x 15.33°	93.53%	4.76%	1.7%	0%	-	-	-	-
Andersson FoV 31.66° x 25.24°	79.39%	10.33%	6.06%	4.21%	95%	83%	9%	0%

the screen, DPI) to allow for a unit transformation. Future research will address more datasets with the goal of finally deriving the metric function. Further research will show if the outcome of the distance metric can be applied to different algorithms, especially to real-time-capable algorithms, or whether further modifications are necessary for each.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 515614446.

References

Iyasu B. Adhanom, Paul MacNeilage, and Eelke Folmer. 2023. Eye Tracking in Virtual Reality: A Broad Review of Applications and Challenges. *Virtual Reality* 27 (2023), 1481–1505. doi:10.1007/s10055-022-00738-z

Ioannis Agtzidis, Mikhail Startsev, and Michael Dorr. 2019. 360-degree Video Gaze Behaviour: A Ground-Truth Data Set and a Classification Algorithm for Eye Movements. In *Proceedings of the 27th ACM International Conference on Multimedia* (Nice, France) *(MM '19)*. Association for Computing Machinery, New York, NY, USA, 1007–1015. doi:10.1145/3343031.3350947

R. Andersson and K. et al. Larsson, L.and Holmqvist. 2017. One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms. *Behav Res* 49 (2017), 616–637. doi:10.3758/s13428-016-0738-9

Louise Gillian C. Bautista and Prospero C. Naval. 2021. CLRGaze: Contrastive Learning of Representations for Eye Movement Signals. In *2021 29th European Signal Processing Conference (EUSIPCO)*. 1241–1245. doi:10.23919/EUSIPCO54536.2021.9616181

Birtukan Birawo and Pawel Kasprowski. 2022. Review and Evaluation of Eye Movement Event Detection Algorithms. *Sensors* 22, 22 (2022). doi:10.3390/s22228810

The SciPy community. 2025a. *Akima1DInterpolator*. https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.Akima1DInterpolator.html

The SciPy community. 2025b. *CubicSpline*. https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.CubicSpline.html

The SciPy community. 2025c. *PchipInterpolator*. https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.PchipInterpolator.html

The SciPy community. 2025d. *resample*. https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.resample.html

NumPy Developers. 2024. *numpy.interp*. https://numpy.org/doc/2.2/reference/generated/numpy.interp.html

Wolfgang Fuhl, Gjergji Kasneci, and Enkelejda Kasneci. 2021. TEyeD: Over 20 Million Real-World Eye Images with Pupil, Eyelid, and Iris 2D and 3D Segmentations, 2D and 3D Landmarks, 3D Eyeball, Gaze Vector, and Eye Movement Types. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 367–375. doi:10.1109/ISMAR52148.2021.00053

Anjith George and Aurobinda Routray. 2016. A score level fusion method for eye movement biometrics. *Pattern Recognition Letters* 82 (2016), 207–215. doi:10.1016/j.patrec.2015.11.020 An insight on eye biometrics.

H. Griffith, D. Lohr, and E. et al. Abdulin. 2021. GazeBase, a large-scale, multi-stimulus, longitudinal eye movement dataset. *Sci Data* 8, 184 (2021). doi:10.1038/s41597-021-00959-y

Joohwan Kim, Michael Stengel, Alexander Majercik, Shalini De Mello, David Dunn, Samuli Laine, Morgan McGuire, and David Luebke. 2019. NVGaze: An Anatomically-Informed Dataset for Low-Latency, Near-Eye Gaze Estimation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3290605.3300780

R. Kothari, Z. Yang, and C. et al. Kanan. 2020. Gaze-in-wild: A dataset for studying eye and head coordination in everyday activities. *Sci Rep* 10, 2539 (2020). doi:10.1038/s41598-020-59251-5

M. Lamb, M. Brundin, Luque E. Perez, and E. Billing. 2022. Eye-Tracking Beyond Peripersonal Space in Virtual Reality: Validation and Best Practices. *Frontiers in Virtual Reality* 3 (2022), 864653. doi:10.3389/frvir.2022.864653

Jose Llanes-Jurado, Javier Marín-Morales, Jaime Guixeres, and Mariano Alcañiz. 2020. Development and Calibration of an Eye-Tracking Fixation Identification Algorithm for Immersive Virtual Reality. *Sensors (Basel, Switzerland)* 20, 17 (2020). doi:10.3390/s20174956

SR Research Ltd. [n. d.]. *EyeLink 1000 Plus*. https://www.sr-research.com/eyelink-1000-plus/

Inc. Magic Leap. [n. d.]. *Field of View (FOV)*. https://developer-docs.magicleap.cloud/docs/device/hardware/fov/

Microsoft. 2023. *Eye tracking on HoloLens 2*. https://learn.microsoft.com/en-us/windows/mixed-reality/design/eye-tracking

D. Purves, GJ. Augustine, and D. Fitzpatrick. 2015. *The Neurology of Eye Movements: Types of Eye Movements and Their Functions*. https://www.ncbi.nlm.nih.gov/books/NBK10991/?utm_source=chatgpt.com Accessed: 2025-01-17.

Yannick Sauer, Alexandra Sipatchin, Siegfried Wahl, and Miguel García García. 2022. Assessment of consumer VR-headsets' objective and subjective field of view (FoV) and its feasibility for visual field testing. *Virtual Reality* 26 (2022), 1089–1101. doi:10.1007/s10055-021-00619-x

Marc Tonsen, Xucong Zhang, Yusuke Sugano, and Andreas Bulling. 2016. Labelled pupils in the wild: a dataset for studying pupil detection in unconstrained environments. In *Proceedings of the ninth biennial ACM symposium on eye tracking research & applications*. 139–142.

Tino Wagner. 2017. *python-samplerate*. https://python-samplerate.readthedocs.io/en/latest/index.html

Shu Wei, Desmond Bloemers, and Aitor Rovira. 2023. A Preliminary Study of the Eye Tracker in the Meta Quest Pro. In *Proceedings of the 2023 ACM International Conference on Interactive Media Experiences* (Nantes, France) *(IMX '23)*. Association for Computing Machinery, New York, NY, USA, 216–221. doi:10.1145/3573381.3596467