scientific reports

Check for updates

OPEN

Efficient diagnostic classification of diverse pathologies through contextual eye movement data analysis with a novel hybrid architecture

Alae Eddine El Hmimdi[1,2], Themis Palpanas[2] & Zoi Kapoula[1,2]✉

The analysis of eye movements has proven valuable for understanding brain function and the neuropathology of various disorders. This research aims to utilize eye movement data analysis as a screening tool for differentiation between eight different groups of pathologies, including scholar, neurologic, and postural disorders. Leveraging a dataset from 20 clinical centers, all employing AIDEAL and REMOBI eye movement technologies this study extends prior research by considering a multi-annotation setting, incorporating information from recordings from saccade and vergence eye movement tests, and using contextual information (e.g. target signals and latency of the eye movement relative to the target and confidence level of the quality of eye movement recording) to improve accuracy while reducing noise interference. Additionally, we introduce a novel hybrid architecture that combines the weight-sharing feature of convolution layers with the long-range capabilities of the transformer architecture to improve model efficiency and reduce the computation cost by a factor of 3.36, while still being competitive in terms of macro F1 score. Evaluated on two diverse datasets, our method demonstrates promising results, the most powerful discrimination being Attention & Neurologic; with a macro F1 score of up to 78.8%; disorder. The results indicate the effectiveness of our approach in classifying eye movement data from different pathologies and different clinical centers accurately, thus enabling the creation of an assistant tool in the future.

Keywords Time serie, Eye movement, Deep learning, Classification, Saccade, Vergence, Hybrid

Eye movement data is a valuable tool for gaining insights into brain disorders and pathologies. Extensive research has been conducted on recording and analyzing eye movement trajectories to develop early detection methods and interventions for these conditions. For instance, studies by Ward and Kapoula have revealed various abnormalities in the eye movements of individuals with dyslexia during reading tests as well as stimulus-driven tests. Additionally, supervised learning algorithms have been explored in multiple studies to screen for such pathologies[1–7].

However, many studies have collected data in controlled laboratory settings using the same acquisition protocol. This can result in differences between the eye movement time series distributions of these datasets and real-world scenarios in terms of variability and signal-to-noise ratio. In addition, the studies often focused on screening for pathology by comparing it to a healthy control group.

This approach may introduce bias since trained models should ideally be able to screen for target pathologies not only among control subjects but also among populations with different pathologies. Furthermore, there is a limitation regarding the visual tasks used to construct the dataset. Most of these studies utilized either context-free image exploration tasks or reading tasks to record eye movements and primarily analyzed saccadic eye movements exclusively.

To bridge the gap between controlled laboratory eye movement studies and real-world contexts, our previous work[7] focused on developing and training a Convolutional Neural Network (CNN) architecture.

[1]Orasis Eye Analytics and Rehabilitation, Paris, France. [2]Laboratoire d'Informatique Paris Descartes,LIPADE, French University Institute (IUF) Universitá de Paris, 45 Rue Des Saints-Peres, 75006 Paris, France. ✉email: zoi.kapoula@orasis-ear.com

CNN architectures have gained significant interest since the introduction of LeNet in 1990[8] for reading digits and zip codes. Subsequently, several variants have improved performance by incorporating new methods. For instance, AlexNet[9] significantly outperformed other models in the 2012 ImageNet ILSVRC challenge[9]. VGGNet[10] demonstrated the crucial role of network depth for performance, GoogLeNet[11] introduced the inception module to optimize parameter count relative to AlexNet, and ResNet[12] introduced skip connections.

Our proposed CNN aimed to predict school learning disorders by analyzing 5-second segments of eye movement time series data (consisting of 1024 points) recorded during various visual tasks like saccades, vergence, and reading. CNNs are deep learning algorithms designed to autonomously learn structured hierarchies of features from input data, excelling in pattern recognition and classification tasks. They utilize convolutional layers to capture patterns using adaptable kernels and pooling layers to extend the receptive field of these learned kernels.

The proposed experimental design aimed to closely reflect the clinical reality of detecting eye movement pathologies by gathering data from 20 clinical centers, using the same technologie (REMOBI and AIDEAL), with 1575 patients representing 18 different pathologies.

In addition to including healthy subjects in the control population, individuals with pathologies other than dyslexia and scholar learning disorders were also included. This real-world approach demonstrated superior performance compared to previous methods.

Exploiting the potential of attention mechanism

In addition to CNNs, another significant class of deep learning architectures, known as Transformers[13], has emerged. These transformers have achieved state-of-the-art performance across various domains, including natural language processing[14–16] and computer vision[17–20]. The notable effectiveness of Transformer architectures is attributed to their attention mechanism, which demonstrates superior capability in learning various tasks, and capturing complex patterns, compared to CNN layers. However, when training on small datasets, these architectures often struggle to generalize as effectively as CNNs, which excel due to implicit regularization techniques like weight sharing.

Consequently, a recent trend has emerged that leverages the strengths of both layers by combining them into a hybrid architecture. Such approaches aims to enhance the expressive power of the model through the utilization of the attention mechanism. These studies can be broadly grouped into two categories: The first category investigates incorporating convolutions within the attention block to effectively capture local and global features and performing the attention mechanism in two-dimensional space[21–24], while the second category concentrates on combining both architectures by sequentially incorporating attention blocks after convolutional blocks[25–29].

This second strategy, has been investigated across a wide range of tasks in diverse domains, as evidenced by various studies. For instance, Sakorn et al.[25] developed a hybrid architecture that achieved up to 98.8% recognition performance on three different time series datasets. Additionally, in image classification, Guoqiang et al.[26] also proposed a hybrid architecture as a solution to mitigate the loss of dataset size. They evaluated their architecture on Cifar10 and Cifar100 datasets. On the other hand, Yihua et al.[27] focused on gaze estimation from images and reported an angular error of 3.08 degrees using their method. For tumor segmentation in brain imaging, Hatamizadeh et al.[28] introduced an hybrid architecture dubbed (SWIN UNETR) trained with both Hierarchical Vision Transformer using Shifted Windows (SWIN) encoder and CNN decoder components achieving a mean dice score of 0.891, which outperformed the baseline CNN-based architectures. Similarly, Philippi et al.[29] applied this same SWIN Hybrid Architecture for retinal lesion segmentation resulting in a Mean dice score up to 0.485. In comparison, the newly explored model performed better compared to traditional Convolutional Neural Network approaches.

To take advantage of our recently proposed CNN architecture, in the current study, we adopt the same hybrid approach. Furthermore, to our knowledge, this technique has not been applied to the classification of eye movement time-series diseases.

The present study

In this study, the aim was to extend our previous model by enhancing its capabilities in handling more complex classification problems. The scope of our research expands beyond the binary classification of scholar disorders and other pathologies. We now address a multi-annotation classification problem involving 8 distinct groups of pathology.

Furthermore, instead of relying on just one segment for predicting pathology, we explore using 10 different segments and aggregating the information to improve prediction accuracy. This approach has increased the resilience of our model to noisy segments.

Consequently, each epoch's duration and GPU memory required per batch have significantly increased by approximately a factor of 10. To optimize runtime efficiency while managing these constraints effectively, we conduct a thorough review and make necessary revisions to our previous HTCE architecture.

Additionally, we present a novel hybrid architecture that combines a lightweight HTCE for constructing embeddings for each segment and a VIT encoder to learn the classification task using these stacked embeddings.

Finally, we explore the inclusion of contextual information time series, such as eye movement latency, stimulus coordinates on the optical axis, and confidence intervals for each point position coordinate. Our findings demonstrate that adding these features improves both model accuracy and its ability to handle noise inherent in the eye tracking pipeline.

The contributions are as follows:

- We establish the feasibility of screening 8 different pathology groups on two datasets and we propose the idea of aggregating 10 segment information into one accurate prediction using three strategies: Max and mean pooling, and utilizing the VIT encoder.
- We review our proposed HTCE architecture to reduce computation and memory costs in order to achieve reasonable epoch duration and memory cost, then we introduce an Hybrid architecture that combines the advantages of HTCE feature extractor's local inductive bias with increased model capacity through a transformer-based classifier.
- We perform ablation studies to evaluate the various model choices, encompassing considerations such as the number of segments and the selection of time series features.
- We conduct a comprehensive comparison of our three proposed architecture with four distinct baselines, which include Recurrent networks, Temporal Convolutional Networks, ROCKET classifier, and CNNs trained on image spectrograms.

Material

Eye movement recording

Eye movements are recorded using the Pupil Core head-mounted video-oculography device[30]. This device provides angular position estimations along both the vertical (y) and horizontal (x) axes at a frequency arround 200 Hz. Note that the x and y axes lie in a plane perpendicular to the optical axis.

This is meta-analysis, of the data coming from 20 clinical centers using the REMOBI technology (patent WO2011073288), and AIDEAL technology (patent PCT/EP2021/062224), who consent that the eye movement trajectories can be used anonymously for further research aiming to improve the technology , ensuring compliance with the General Data Protection Regulation (GDPR). These are data collected during routine clinical protocols for eye movement analysis are utilized for research purposes aimed at enhancing the technology. These technologies are employed to elicit and analyse saccade and vergence eye movements.

Eye movement saccade and vergence

In this study, we have examined two eye movement: the saccade and the vergence. These two movements, are the most important elementary eye movements used daily to explore the world. Saccadic movements control direction, with both eyes moving in the same direction. In Supplementary Figure [1], we present an example of left and right eye position time series on the horizontal axis when performing the saccade test. In contrast, vergence movements control depth by moving in opposite directions, as depicted in Supplementary Figure [2]. Finally, elegant throughout presentations of eye movements and their neurology can be found in[31] and in french on the thesis on Marine VERNET's thesis[32].

REMOBI and AIDEAL

To analyze eye movement, various clinical centers use the REMOBI device, presented in Supplementary Figure [3], to conduct diffrents task including the saccade and vergence tasks. Each subject is positioned such that Remobi is at eye level, and the first layer of LED is positioned 30cm from the eye. Each test has a duration of approximately 2 minutes. The synchronization of the target signal and the eye-tracked data is implicitly achieved using the Unix timestep time series generated by each of the two systems-the eye tracker and the Remobi. The eye movement analysis is conducted relatively to the Remobi time series data using the AIDEAL software, which implements a velocity-based criterion to detect the start and end of the movement. Once the trials are identified, the software computes several eye movement parameters such as amplitude, velocity, drift, as well as saccade disconjugacy. For instance, several studies analyse in depth the saccade and vergence eye movements by studying different eye movement parameters such as duration, latency, and velocity[33–35]. Furthermore, the same experimental setting are to analyze these parameters, in several studies, using statistical frameworks[36–39] as well as machine learning algorithms[1,2].

The goal of the saccade task is to analyze eye movements and fixation after movement. Participants are instructed to visually focus on randomly appearing stimuli along a horizontal axis. In the vergence task, we aim to observe both convergence and divergence eye movements. This is accomplished by fixing a stimulus presented at various positions and durations over the optical axis. Finally, to prevent participants from predicting motion, the duration and position of LEDs are randomized in both tests. Each test comprises 40 trials: 20 leftward and 20 rightward for the saccade test, as well as 20 coordinated and 20 uncoordinated for the vergence test.

Dataset overview

We follow the procedure outlined in[10] to construct the Ora23 dataset using our database and extract annotations from clinician reports. Ora23 comprises data from different European countries, including 20 different European clinical centers, with a population age ranging from 5 to 75 years old. This dataset consists of 3181 subjects (92207 samples) performing the saccade visual task and 3228 subjects (95630 samples) performing the vergence visual task. Table [1] illustrates the target distribution for the two datasets, in term of subject. In addition, Supplementary Figure [4] illustrates the target distribution in terms of segments.

We obtain annotations from clinicians' reports of patient information and eye tracker records. This information includes a text description of each patient's reasons for consulting clinician. The dataset encompasses more than 100 different pathologies, including dyslexia and learning disorders, strabismus, vertigo, presbycusis, vergence-accommodation disorders, low vision, and motor restlessness. Subsequently, we cluster multiple families of disorders to form the final seven groups of pathologies. Finally, a new class 'Other' is defined, which incorporates all subjects with no clear diagnosis, missing diagnosis, or infra clinic pathologies, as well as subjects with no pathology. We provide a non-exhaustive description of each group of pathology using ICD-11

Class identifier	Corresponding disorder	Saccade dataset	Vergence dataset
0	Dyslexia	873	854
1	Reading disorder	1264	1265
2	Listening & expressing	331	321
3	Vertigo & postural	396	372
4	Attention & neurologic	1016	975
5	Neuro-Strabismus	455	511
6	Visual fatigue	678	567
7	Other pathologies	195	279

Table 1. Presentation of the patient count for the saccade and the vergence datasets.

in Supplementary Table [6]. Please note that the description may not be exhaustive for the last four classes (c5, c6, and c7) due to the grouping heuristic used to construct those groups of pathologies.

Problem statement

Our dataset, denoted as $\mathcal{D} = (X_i, y_i)$, where $i \in 1, \ldots, N$, consists of N samples. Each (X_i, y_i) pair corresponds to a multivariate time series of length T, with $X_i \in \mathbb{R}^{15\times T}$, and a corresponding target class, y_i. The input features (X_i) encompass the horizontal and vertical angular positions of both eyes for each point within the interval of length T, as well as their first and second derivatives, along with context-based information such as latency, LED coordinates over the optical axis, and confidence level. Our task involves predicting the class y_i given an input X_i.

In our previous approach[7], addressing the extended length of our records (approximately 30,000 time points), we investigated operating on segments of size S = 1024, equivalent to 5 seconds of recording. In this approach, we adopt a multi-segment approach by initially constructing embeddings for 10 different segments and then aggregating these 10 embeddings, feeding them into a neural network classifier. This approach reduces the model's sensitivity to highly noisy segments.

Eye movement preprocessing

This section presents the data preprocessing steps, following the methodology outlined in a previous study[7], with some modifications to the standardization and segmentation methods. We provide a concise summary of these key procedures. Initially, we conduct two levels of data cleansing comprising a low-pass filtering step using a Gaussian FIR filter with a cut-of frequency of 33Hz and a z-score filtering step. In the z-score filtration process, we identify and eliminate data points that have z-scores exceeding 2.5. Each time series recording is individually filtered using its own statistics, which are computed based on the entire time series recording.

In contrast to implementing the standardization technique suggested in prior research, our approach draws inspiration from techniques used in image processing. Specifically, for each centered angular coordinate, we compute the modulo angular coordinate value with 90 and then divide each coordinate by 90. This alternate strategy demonstrates lower sensitivity towards distribution shifts when compared to Z-normalization while capitalizing on the cyclic nature inherent within angular coordinates.

Random segment sampling

Our approach aims to mitigate model error variance by using a multi-segment strategy coupled with dynamic random sampling. We empirically observed an inverse correlation between the number of segments used to form each unique sample and the model's validation error. However, this improvement comes with increased complexity and memory requirements during batch fitting. Thus, we retain a trade-off of selecting 10 segments per sample, corresponding to approximately 50 seconds of recording. This decision balances the need for a diverse training set with manageable computational resources.

In our dynamic sampling process, 10 segments are randomly sampled for each sample during model fitting iterations. Initially, a batch of unique recording identifiers is selected, followed by the random selection of 10 segments from each recording ID. This strategy enhances model training regularization compared to consecutive sampling methods by increasing the diversity of the training set. For instance, consider a recording comprising 50 segments. Using consecutive sampling, we can construct 41 unique samples. In contrast, with random sampling, the theoretical number of tuple samples exceeds 10^{16}, a substantial increase in sample diversity by a factor of approximately 10^{14}. By employing a higher number of segments alongside dynamic random sampling, we aim to maximize the utilization of our dataset while promoting model generalization.

Contextual information and gaze derivatives

To improve classification accuracy and enhance analysis, we propose the integration of Gaze derivatives as well as contextual information such as Latency, LED coordinates, and confidence intervals. These variables offer additional insights into eye movement patterns, facilitating a more comprehensive analysis of the data. Thus, the final multivariate time series has 15 components as presented bellow:

- Eye Movement Gaze Signals: These signals correspond to the horizontal and vertical positions signals of the left and right eyes, constituting the initial four time series used in our previous study[7].
- Gaze Derivative: Comprising eight time series, the Gaze Derivative represents the velocity and acceleration for each of the four gaze signals mentioned above, aiding in the analysis of eye movement dynamics.
- Remobi Target Signal: Utilizing a Remobi device during recording, this signal reflects the status of different LEDs throughout the recording phase. It allows our architecture to deduce when each target LED was activated, capturing the latency of eye movement - defined as the time interval between LED activation and eye movement initiation towards a specific target.
- LED Coordinates: Representing the coordinates of LEDs along the optical axis, this signal provides valuable spatial information regarding eye movements, including convergence, divergence, and saccade disconjugacy.
- Confidence Level: Encoding confidence levels for estimated gaze positions obtained from an eye-tracking system at each timestamp, this enhancement improves the overall robustness of our architecture.

Finally, to standardize contextual features, namely Latency, LED coordinates, and the interval of confidence, we employ a min-max standardization method, which is defined as follows:

$$X_{\text{standardized}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

where X is the original value, X_{min} is the minimum value of X in the dataset, and X_{max} is the maximum value of X in the dataset.

Methods

To implement our multi-segment prediction approach, we modify the HTCE architecture proposed in our previous work[7]. We replace the segment-based Multi-Layer Perceptron classifier with a multi-segment-based MLP classifier that utilizes poling to aggregate the different segment embeddings.

Additionally, to optimize the architecture for efficient processing, we introduce several modifications to the basic building block of our architecture. Finally, we explore a more advanced aggregation strategy by constructing a hybrid architecture called Hierarchical Temporal Convolutions and Segment Attention for Eye Movement Analysis (HTCSE). This architecture employs ConvBlock to handle temporal dimensions and attention blocks for processing segment dimensions.

Pooling based aggregation

HTCE Backbone

The HTCE Backbone is based on temporal convolutions and multi-scale context aggregation. Temporal convolutions have proven effective for processing sequential data efficiently[40–43].

The feature extractor consists of 4 stages, as illustrated in Supplementary Figure [5] in the left subfigure, with each stage having a varying number of filters in the convolution layers. Furthermore, each stage consists of 3 Convblocks that share the same hyperparameter settings. Additionally, a skip connection is incorporated between the input and output of each stage, followed by layer normalization. Please refer to our previous study[7] for more details on the HTCE architecture, as well as Supplementary Table [3], where we summarize the different HTCE hyperparameters.

Reviewing the ConvBlock design

To streamline HTCE's computational complexity, we've redesigned the ConvBlock module by consolidating normalization to the input of each block, omitting normalization layers between convolutions and activations. The updated basic block architecture, detailed in Supplementary Figure [6], begins with input normalization. Subsequently, three convolutional layers, each with distinct dilation rates (1, 4, and 8) and int(d/3) filters, are then applied and their outputs merged and transformed using ReLU activation. A final convolutional layer with d filters and ReLU activation follows, along with a skip connection between input and output. Optionally, pooling is performed as the final step of the ConvBlock.

Hierarchical temporal convolutions and segment attention

In this approach, we aim to create a hybrid architecture that combines the advantages of both CNN and attention-based models. By stacking lightweight HTCE with an attention-based feature extractor, we can leverage the locality and weight-sharing inductive bias of CNNs while also benefiting from the high model expressivity provided by attention blocks.

Lightweight HTCE

To optimize memory utilization and parameter count during training, we simplified the structure of the HTCE backbone by reducing the model depth. We present an overview of this architecture in the right subfigure of Supplementary Figure [5]. Note that instead of using three convolutional blocks per stage, we reduce it to two blocks per stage by removing each first conv-blocks that do not involve temporal reduction via pooling. In addition, we reduce the number of parameters by half in the first stage. This simplification reduces computational costs, required memory for training, and the number of parameters in the model. The set of different hyperparameters for the Lightweight HTCE variant is summarized in Supplementary Table [4].

Attention encoder

In addition to the hierarchical temporal convolutions, we incorporate an attention-based feature extractor to enhance the model's ability to capture important temporal patterns in the eye movement gaze data and aggregate information across the segment as well as the temporal dimensions.

Our attention encoder is based on the attention encoder described in previous research[13,20]. It consists of three stacked attention blocks, with each block containing a multi-head attention module followed by two dense layers.

The first projection increases the feature dimension by a factor of 4 and applies Gelu activation[44] while the second projection reduces the feature space back to its initial size. Furthermore, we have utilized pre-norm residual units for both the attention blocks and the multi-layer perceptron, as recent studies have shown that this scheme is more efficient for training compared to post-residual units[20,45–47].

Overall architectures

We introduce two feature combination methods: pooling-based aggregation and attention-based aggregation. Supplementary Figure [7] presents an overview of the two proposed architectures: a pooling-based approach on the left, and an attention-based feature aggregation approach on the right. Recall that our input has a shape of (B, 10, 1024, 15), representing a batch of size B with 10 segments, each containing 15 multivariate time series of length 1024.

Pooling-based approach

In the pooling-based approach, we utilize the HTCE architecture as a feature extractor. We investigate two pooling techniques: max pooling (HTCE-MAX) and mean pooling (HTCE-MEAN) to combine segment embeddings into a unified feature map. This resulting map is then input into an MLP classifier. In Supplementary Figure [7], we present, in the left subfigure, an overview of our multi-segment architecture.

We first flatten the batch and segment dimensions, pass them through the HTCE backbone to generate embeddings for each segment, and then unflatten them back to their original dimensions.

Next, within the sample index dimension (second axis), we employ max pooling on the HTCE-MAX variant and mean pooling on the HTCE-MEAN variant (represented by the green module in the left subfigure) to compute an aggregated embedding. A global average pooling is applied to entirely remove the temporal dimension, followed by a normalization layer. The final embedding is then fed into an MLP classifier (represented by the orange module in the left subfigure).

Attention-based Approach

In the attention-based approach, we adopt a hybrid approach (HTCSE) for advanced aggregation strategies, depicted in the right subfigure of Supplementary Figure [7]. This bifurcated process divides feature extraction into two levels: the first at the temporal level, employing a lightweight HTCE, and the second at the segment level, using an attention-based encoder to capture segment-specific details.

First, the input data, undergoes processing in the first feature extractor to convert temporal information into feature dimensions. This transformation reduces temporal dimensions from 1024 to 15 and increases feature dimensions from 15 to 512 using a lightweight version of the HTCE backbone, represented by the four stages in the right subfigure, in Supplementary Figure [5]. Each layer at this level operates solely on the last two dimensions: temporal and feature dimensions, independently processing batches and segments.

After the embedding process, ten resulting embeddings are stacked along the temporal axis (the second axis). The encoder, depicted as the segment level in Supplementary Figure [7], processes each concatenated sample embedding, which has a size of 150 and a dimensionality of 512. This method facilitates precise feature extraction across various temporal granularities, generating a feature map. The CLS slice of this feature map is utilized by the final layer, represented by the orange module in the right subfigure, for classifying different pathologies.

Optimizing the memory and computation costs

Our primary objective in designing the HTCSE architecture is to maximize the model's learning capacity while maintaining a reasonable training time. By increasing the parameter size, we enable the model to capture more complex relationships, given a sufficient amount of data. We use the parameter count as an estimation of the model's capacity. However, the training duration depends on various factors such as parameter size and the computational resources associated with the architecture. The computational cost is influenced by design choices in the architecture and the operations performed during forward and backward passes. This cost can be estimated using FLOP (floating point operations) as a metric. A higher number of FLOPs in an architecture leads to longer durations for each pass. Similarly, we also considered memory costs when determining our trained architecture. Increasing the maximum memory allowed for one pass reduces the feasibility of training with larger batch sizes, resulting in longer epoch durations.

To provide a thorough comparison between the two architectures, we have included an evaluation as well as the parameter count for each of the three HTCE variations. To ensure fairness in our comparison, we set up the HTCE with base parameters of 128 and configured the Hybrid HTCE with base parameters of 64 to achieve a similar number of parameters. The measurements were conducted using an Nvidia V100, with a batch size of 32 and mixed precision.

Model fitting

Train/Test Split

To evaluate our proposed methods, we employ stratified cross-validation, a variant of the cross-validation algorithm. We use the IterativeStratification library from the skmultilearn package[48], which is specifically designed

for multi-label data. This method ensures that class proportions between training are similar and test folds throughout each iteration.

Initially, the algorithm partitions subject identifiers into three complementary subsets (folds), maintaining consistent class proportions between training and test folds. Then, for each subset, we replace each patient identifier data with its corresponding segments. Finally, the model is trained during 3 iterations, where in each iteration, data from 2 folds are used for model training, and the remaining fold is used for testing. In Supplementary Table [1], we present the sizes of the training and test sets for each fold in terms of patients as well as segments.

Model training

We use the TensorFlow package to implement the differents deep learning architectures and for model fitting. We utilize a single GPU (NVIDIA A100 80 GB) and manually optimize each model's hyperparameters. We train each architecture for 100 epochs using the AdamW optimizer, setting the learning rate to a low value of $1e^{-4}$ to mitigate instability observed with higher values. The weight decay[49] is set at $1e^{-5}$, which is 0.1 times the learning rate, to regularize model training. Furthermore, we implement early stopping by monitoring the validation F1 global score, stopping training if the metric does not improve for 10 consecutive epochs. Our choice is due to observed instability in validation scores caused by random data sampling, resulting in different training and validation sets each epoch. To address this, we employ an exponential moving average with a coefficient of 0.1 for stability. Additionally, we use a high patience waiting period for the early stopping algorithm.

For the different Deep Learning models except for the HTCSE, each iteration processes a batch of 128 samples, where each sample consists of 10 segments, totaling 1280 samples per batch. However, for the HTCSE, we leverage the increased memory capacity and thus double the training batch size. Additionally, to optimize memory usage, we adpote mixed precision training. To mitigate dataset imbalance, we use focal loss optimization and per-sample loss weighting based on class weight, with binary focal losses independently optimized for each class using a gamma value of 5. Furthermore, to achieve class balancing, different alpha values are assigned to individual classes. For a comprehensive overview of these alpha parameters and all remaining model training hyperparameters, please refer to Supplementary Table [5].

Model evaluation

To evaluate our architecture, we considered the macro F1 score for each class[50]. Additionally, we consider aggregating the different per-class F1 scores into global metrics. We present the formulas of or each defined metric in Supplementary Table [2]. Furthermore, the definitions of each metric are provided in Supplementary Table [7].

Each model is evaluated using a stratified 3-fold cross-validation approach. During training, we observe high variance in the different metrics within each epoch due to the real-time sampling of the 10 segments from each recording sample. This results in the validation set differing with each epoch, thereby affecting the performance of the early stopping technique. To mitigate this issue, an exponential moving average is applied to each metric score with a smoothing factor of 0.1, using the previous score from the last epochs. Finally, we report the mean value for each metric across the three folds. Furthermore, to ensure a fair comparison, all models are trained and evaluated using the same fold split.

Model performance analysis

Ablation study

To evaluate our design, we performed additional experiments to assess the importance of each component. First, we experimented with dilated convolution by training the three architectures: HTCE-MAX, HTCE-MEAN, and HTCSE, both with and without the dilation rate. Initially, we replaced the three dilation-based convolution modules in the ConvBlock, as well as the subsequent concatenation layers, with a single convolution module having a filter size equal to the sum of the three convolution filters, along with similar kernel size and activation (HTCE-MAX-ND).

To reduce computational costs, we focused solely on using the HTCE-MAX architecture and the saccade dataset for further experiments. Additionally, since this variant reduced the training duration by approximatly a factor of 28%, we performed the remaining ablation analysis using this variant (HTCE-MAX without dilation).

Furthermore, we experimented with different numbers of segments: 1, 3, 5, 7, and 10. Finally, we considered exploring the importance of each feature by sequentially removing the confidence level time series, the Remobi time series, and finally the gaze derivatives time series.

Comparison with other classifiers

To evaluate our proposed method, we compared it with five different baselines. Initially, we replaced our architecture with three distinct deep learning time series architectures. Subsequently, we experimented with additional classifiers, including the Rocket time series classifier and a 2-dimensional CNN applied to image spectrograms. It is important to note that our time series data has a large temporal dimension of 10,240. Therefore, directly substituting our meta-architecture with different baseline models in our setting is not straightforward. Thus, we adopted the same multi-segment approach by replacing our HTCE encoder with the different baselines. This process corresponds to substituting the HTCE module in the left subfigure of Supplementary Figure [7] with each baseline except the Rocket baseline:

- **TC-MAX**: we substitute the HTCE encoder with the Temporal convolutional architecture used in[51,52]. For a fair comparison, we increase the number of filters per stage from (128, 128, 128, 128) to (128,256,512,512).

	Saccade visual task				Vergence visual task			
	Epoch	Global	Negative	Positive	Epoch	Global	Negative	Positive
Model	Duration (s)	F1	F1	F1	Duration (s)	F1	F1	F1
HTCE-MAX	318	69.1	89.1	49.1	392	68.0	**89.4**	46.7
HTCE-MEAN	313	69.1	**89.4**	48.8	388	67.8	89.2	46.4
HTCSE	201	**69.3**	88.2	**50.3**	236	**68.4**	88.0	**48.8**
TC-MAX	280	68.9	88.9	49.0	282	66.4	87.9	44.8
LSTM-MAX	523	68.6	88.7	48.4	591	65.3	87.1	43.6
GRU-MAX	523	69.2	89.0	49.3	589	66.3	87.8	44.8
CNN2D-MAX	171	62.0	84.8	39.3	183	60.9	84.3	37.5
ROCKET	–	62.3	86.8	37.8	–	62.1	86.7	37.4

Table 2. Presentation of The epoch duration, The Global, positif, and negatif F1 scores for each of the different trained algorithms when trained on the saccade and vergence visual task.

- **LSTM-MAX**: we substitute the HTCE with the LSTM based architecture proposed in[53]. For a fair comparison, we increased the number of LSTM stages from three to five and increased the number of units from 64 to 512.
- **GRU-MAX**: we replace in LSTM-MAX architecture the LSTM module with the GRU module.
- **CNN2D-MAX**: We substitute the HTCE with a two-dimensional CNN encoder, which processes the log-Mel spectrograms[54,55] of each segment. Initially, each eye movement time series segment undergoes processing using the Short-Time Fourier Transform (STFT) with a frame length of 1024 and a frame step of 32. Subsequently, the magnitudes of the STFT are computed and multiplied with a Mel-filter bank consisting of 160 channels within the frequency range of 10-100 Hz to generate a Mel-scaled spectrogram. Finally, the spectrogram is scaled using the decibel (dB) scale to ensure perceptual relevance. We employ EfficientNet[56] as the encoder, omitting contextual information. Combining spectral information with time series information from different time series is not straightforward, necessitating separate encoders.
- **Rocket**[57]: we explore a time series classifier employing non-learnable random kernels to compute a feature map, which serves as a set of features for a ridge regression classifier. This algorithm has gained recognition for achieving state-of-the-art results compared to other time series classification algorithms[57–59] due to its superior accuracy-computation cost trade-off. For each pathological class, we trained a binary Ridge classifier using features computed via the Rocket algorithm[57]. We utilized the implementation provided in[60] and set the number of kernels to 1000 for each class, resulting in a total of 8000 kernels. For the ridge classifier, we employed the Scikit-Learn implementation[61]. We experimented with 10 regularization alphas ranging from -3 to 3 in the logarithmic space. For the remaining parameters, we utilized the default settings of each library.

To assess the statistical significance of our method compared to the baseline, we employ the non-parametric Wilcoxon signed rank test. for each method, We construct lists of validation scores by concatening the validation score of each batch and on each fold, then we use the Wilcoxon test to evaluate their significance of each two methods pairs, we use the scipy implementation[62].

Results

Analysis of the global performance

The F1 scores for each of the different trained algorithms are presented in Table [2] for both the saccade and the vergence visual tasks. In addition, we provide in Supplementary Table [19] the standard deviation of the different global score, across the three folds. Finally, we present in Supplementary Tables [15] and [16] the results of the Statistical Wilcoxon Test, used to assess the statistical significance of the comparison of Global F1 scores as presented in Table [2].

Overall, the various HTCE variants outperform the baseline methods. When considering the Global F1 score, the Hybrid HTCSE architecture achieves the highest score, followed by the two HTCE variants architecture. Furthermore, HTCE-MEAN achieves consistently the lowest variability (0.57 and 0.23) on both the saccade and the vergence tasks, respectively. Similarly, the highest variability is observed in the hybrid architecture.

Additionally, when assessing the statistical significance of the Global F1 score-based comparison on the vergence dataset, all the score differences are statistically significant except for the pair comparisons, namely TC-MAX vs GRU-MAX, HTCE-MAX vs HTCE-MEAN, and HTCE-MAX vs HTCSE. On the other hand, on the saccade dataset, all the Global F1 score-based comparisons are statistically significant except when comparing the two HTCE CNN variants, namely HTCE-MAX and HTCE-MEAN, and when comparing the three models TC-MAX, GRU-MAX, and HTCSE together.

Furthermore, when examining positive and negative classification performance separately, there is a slight difference as the CNN-based architecture performs better at classifying normal populations, while the hybrid-based architecture excels at detecting pathologies. Finally, when considering variability criteria, HTCE-MEAN and GRU-MAX achieve the lowest variability, in terms of the positive F1 score (0.77 and 0.42) on the saccade

Architecture	Number of Params.	FLOP (10^9)	Max required memory
HTCE-Baseline	7.37×10^6	465.88	1.53 Gb
HTCE-Max	7.19×10^6	410.56	1.40 Gb
HTCSE	9.01×10^6	138.29	1.12 Gb

Table 3. Evaluation of parameter count, computational cost (FLOP), and maximum required memory for the proposed architectures (HTCE-Max and HTCSE) in comparison to our proposed backbone, HTCE-Baseline.

and the vergence datasets, respectively. On the other hand, GRU-MAX and HTCSE achieve the lowest variability (0.05 and 0.15), in terms of the negative F1 score, on the saccade and the vergence datasets, respectively.

Analysis of the per-class performance

In addition to the global, negative, and positive F1 scores, we also evaluated the macro F1 score per sample. The results are presented in Supplementary Tables [8] and [9] for each of the different algorithms applied to both the saccade and vergence datasets. Finally, we provide in Supplementary Tables [20] and [21], the standard deviation of each metrics score across the three folds.

When considering the saccade dataset, all models except the CNN2D-MAX and the ROCKET classifier achieve higher overall macro F1 scores. The best models differ for each class: GRU-MAX achieves the best macro F1 score for classes 2, 5, and 6, while HTCE-MEAN achieves the best F1 score for classes 0 and 1. Finally, the Hybrid HTCE and the ROCKET achieve the best scores for classes 3 and 7, respectively.

On the other hand, when considering the vergence dataset, our three proposed architectures consistently achieve the best performance across all different groups of pathologies, except for classes 3 and 7, where the best performance is achieved by the GRUP-MAX and the ROCKET algorithms, respectively. Additionally, the Hybrid architecture achieves the best score for classes 0, 2, and 3, while the two CNN HTCE-based variants achieve the best score for classes 1, 4, 5, and 6. Please refer to Table [1], which provides an overview of the pathology group associated with each class.

Analysis of the dyslexia screening performance

Finally, to explore more in detail the model's performance on class 0 (dyslexia), we present the sensitivity, specificity, and F1 scores for each class and model, in both the saccade and vergence datasets in Supplementary Tables [17] and [18]. The Hybrid architecture demonstrates a higher sensitivity compared to the second-best model, exhibiting an increase of 3.8 and 5.7 points on the saccade and vergence datasets, respectively. Conversely, the TC-MAX and the two recurrent architectures (LSTM-MAX and GRU-MAX) achieve higher specificity on the saccade dataset, while in the HTCE, two variants (HTCE-MAX and HTCE-MEAN) attain the highest specificity concurrently on both datasets.

Ablation studies

Incorporating dilation rate

Supplementary Tables [10] present the global F1 score when training the HTCE-MAX, HTCE-MEAN, and HTCSE with and without the dilation rate. When considering the saccade dataset, replacing multiple small layers with different dilation rates with a bigger convolution increases the performance. On the other hand, when considering the vergence dataset, preventing the model from using dilation consistently decreases the performance.

Impact of the number of used segments

Additionally, Supplementary Tables [11] and [12] offer insights into the Global F1 and per-class macro F1 validation scores derived from experiments with varying numbers of segments using the HTCE-MAX model. Upon analyzing the Global F1 score, a clear correlation emerges between the model's performance and the number of segments employed, with peak performance observed at 10 segments. Moreover, the highest Global F1 score is achieved with 10 segments, particularly notable for the most prevalent classes in the dataset, namely classes 0, 1, and 4. It is noteworthy that our models are primarily optimized for the first two classes. For instance, our primary focus is on employing our architecture in screening scholar disorders, we opt to utilize 10 segments despite the overall suboptimal performance associated with this specific segmentation approach.

Removing contextual information

Finally, we present in Supplementary Tables [13] and [14] the Global F1 as well as the per-class macro F1 validation scores when removing the different auxiliary time series. The order of importance of the different time series, in terms of the global F1 loss relative to the HTCE-MAX when trained using all the time series, is the Remobi time series followed by the Gaze Derivative, and then the confidence level time series. When considering the per-class F1 score, while the absence of the Remobi time series shows a consistent decrease across the different classes, removing the confidence level time series yields different results for each class.

Comparison of the CNN and the Hybrid Architectures

Table [3] presents an evaluation of the computational and memory costs for the initial HTCE-Baseline, which incorporates the unreviewed HTCE encoder, our reviewed HTCE-MAX, and the Hybrid HTCSE.

The HTCE-Baseline uses a multi-segment approach through its implementation of the ConvBlock module discussed in[7], while the HTCE-MAX utilizes the same architecture with some modifications resulting in a 12% decrease in computation cost and an 8.5% reduction in memory usage compared to previous designs.

Furthermore, our proposed hybrid architecture has more parameters than HTCE-MAX but requires 70% less memory for training purposes and exhibits reduced computation cost by approximately 27%, when compared to HTCE-Baseline.

Supplementary Figure [8] depicts the progression of computation and memory costs as the number of parameters increases for three different architectures: HTCE, HTCE-MEAN, and HTCSE. It should be noted that all parameter counts are plotted in logarithmic scale to enhance visualization.

Upon analyzing the computational complexity in the logarithmic scale, we can observe that there is a direct relationship between the CNN architecture and the Hybrid architecture. This implies that for any given value of x, the computational load of the first architecture is consistently 3.36 times higher than that of the second architecture. This scaling factor serves as a fundamental criterion to compare their performance and resource demands across various parameter values.

Discussion

We highlight the unexplored field of pathology screening using gaze data and deep learning approaches. We summarize our results by comparing the training performances of three proposed architectures: HTCE-MAX, HTCE-MEAN, and HTCSE, against different baselines. To address challenges, we explore incorporating contextual information, developing a multi-segment approach, taking advantage of the high expressivity of the Transformers encoder by building a Hybrid architecture, as well as leveraging the implicit bias of the CNN such as weight sharing. Emphasizing the importance of structured data, we prioritize stimuli-driven tasks like Remobi saccade and vergence tests for enhanced analysis. The examination of attention maps contributes to understanding neural network processing in eye movement trajectory analysis. Finally, future directions include dataset integration, physiological data augmentation, decision explainability, and applying trained architectures in assistant AI technology. Each point is discussed below.

Deep learning application eye movement data

There have been multiple studies that utilize supervised learning to train deep learning architectures for tasks such as age and gender classification[51,52], pathology screening such as Autism Spectrum Disorder[63–67], Dyslexia[4,6,68], Alzheimer[69,70], and Parkinson[71], as well as other classification and regression tasks[72–75]. Furthermore, the methods of processing can be categorized into three groups: mapping each time series to a sequence of letters through natural language processing[63]; mapping each time series to a scan path[6,64,65,67,68,70,74,75]; or estimating the eye position coordinate, then mapping each time series data by dividing them into segments using sliding windows in the temporal domain[51,52,71,72] or in the spectral domain[4]. Furthermore, when it comes to architecture design, some studies employ CNNs[4,6,51,52,63,65,67,68,70,73,75], while others opt for recurrent neural networks, including RNN[72], LSTM[63], and GRU[69,74], or random convolution using the Rocket classifier[71]. Regarding dataset sizes, they ranged from 15 to 215 subjects with an average size of approximately 59 examples.

An analysis of the overall model performance

In our study, we intentionally selected Transformers for their expressive power and CNNs for their inherent bias, aiming to strike a balance between model expressiveness and lightweight interpretability, several studies incroporate interpretable layers to gain in intepretability[76,77,77,78] , however this came with the cost of sacrifying the model high expressivity.

For instance, when considering the incorporation of the dilated convolution module, we observe that while saccade eye movements exhibit stereotypical behavior with less variability, vergence eye movements demonstrate higher variability. This variance allows the model to leverage dilated convolution to address data variability effectively. In the saccade dataset, all models achieve relatively high scores in terms of the Global F1 score, summarizing overall model performance across different classes, assessed by their global macro F1 score.

Conversely, in the vergence dataset, which poses more significant challenges due to its high variability, differences between our proposed architecture and the baseline models become more pronounced. Notably, there is a 2-point difference in the global F1 score between the Hybrid architecture and the best model among the baselines. Finally, we observe exceptional performance of the Rocket algorithm in handling the last group. We hypothesize that this may be attributed to random kernel initialization and the high variability within this class, comprising multiple pathologies with limited available data for separate consideration.

The advantage of using well-structured data

The majority of studies in this field focus on more general visual tasks, such as scene exploration, image exploration, and reading. However, our study takes a different approach by using stimuli-driven visual tasks like the Remobi saccade test and the Remobi vergence test. Our primary objective when constructing our dataset was to simplify the visual task. By only including elementary eye movements such as saccades and vergences, we aimed to detect irregularities more effectively. These specific visual tasks are well-suited for evaluating eye movement trajectories. When performing stimuli-driven tasks, it is easier to compare each eye's trajectory with the optimal trajectory based on the position graph of the stimuli. Additionally, comparing the geometric structure of two

eye movement graphs is straightforward due to their high correlation resulting from shared stimulus positions. As a result, using such a task to design our dataset. leading to better generalization ability.

Hence, utilizing this type of task for dataset design can simplify analysis and enhance generalization abilities by reducing the complexity of eye movement trajectory analysis. For instance, removing the Remobi time series result in a loss in the per-class macro F1 score between 0.7 and 3.2 points, and a loss in the global Positive F1 of 3.2, which characterizes the overall model ability to screen the different pathologies.

The limitation in using context-free exploration task

On the other hand, contextual-free exploration tasks and reading tasks increase the complexity of the analysis. When examining the context of free exploration tasks, it is evident that two observers can have completely different and unrelated eye movement trajectories when observing the same image. This poses a challenge in constructing an accurate ground truth for optimal eye movement trajectories during image and scene exploration. Additionally, reading tasks involve not only visual factors but also cognitive abilities, which introduces biases in trajectory analysis. For instance, there may be differences in reading speed and fixation duration between native language comprehension and second language comprehension. Despite the richness of both context-free exploration and reading tasks, simplifying the data structure by using stimuli-driven visual tasks, reduces the complexity in our problem-solving approach. This allows models to achieve better results while utilizing less data.

Visualizing the attention map

In order to provide insight into the findings of our study, we utilized a technique called visualizing attention maps. This method allows us to understand how neural networks process different segments, specifically those with attention mechanisms, by visually depicting which parts of the input data receive focus and the level of attention given to each segment. To construct this global attention map, we followed a procedure similar to previous studies using similarity tensors in the encoder. Afterward, we standardized the resulting attention maps using the min-max scaler algorithm. Finally, these standardized scores were used to assign color values for each subsegment sized 256 through linear interpolation between green and red colors. The objective is to examine how much attention is being paid and which specific areas are being focused on by analyzing these maps.

Supplementary Figure [9] illustrates this visualization algorithm when applied specifically to positional left and right eye movement signals along with their conjugate signals. In addtion to the visualisation method proposed several method exist in the literature , we provie bellow a concise summary of the state of the art[79] on the method of interpretability of the CNN as well as the Transformer layers.

When concidering the interpretability of Convolutional Neural Network (CNN) architectures, various methods, such as Gradient-based, perturbation-based, and Class Activation Maps (CAM)-based techniques, come into play. Gradient-based methods[80–82] elucidate model decisions by tracing the gradient path from the output node to the input nodes, yielding a saliency map. Conversely, perturbation methods [16, 36] discern relevant input zones by analyzing the model's reactions to perturbations in different regions. Class Activation Maps-based methods construct saliency maps utilizing activation from the convolutional layer.[83–87].

Future directions

The results are promising. In the future, there are various potential avenues for further exploration of deep learning applied to eye movement gaze data collected from clinical centers. For instance, one direction could involve combining the two datasets used in our study instead of considering them separately as we did. Additionally, it is worth noting that transformer-based architectures typically require a larger amount of data compared to CNN-based architectures. Therefore, in our next step, we will consider developing a physiological data augmentation method to improve model training regularization and enhance generalization ability. Another important aspect is decision explainability. While we have produced figures to illustrate the attention weights used by the model, there is still a need to understand how each segment contributes to the model's decision-making process.

Noticeably, annotation is even more difficult clinically, particularly for cases like dyslexia and learning disorders. Therefore, the algorithms presented here could also be useful as a research tool for identifying eye movement-based appreciations.

Moreover, a second potential application of this research is integrating trained architectures into assistant AI technology. By incorporating eye-tracking technology within an assistant AI framework, doctors may be able to access accurate and real-time information quickly by utilizing eye gaze data.

Towards deep learning eye movement analysis

Traditional eye movement analysis[37,38] typically involves using a velocity threshold criterion to detect the start and end points of each movement. Parameters such as amplitude, velocity, and duration can then be calculated from these identified points to gain insights into patterns and characteristics of eye movements, which aid in understanding visual attention.

Another approach focuses on fixations rather than saccadic movements[88–90]. This is done by employing clustering techniques to extract fixations from the data. The resulting clusters can then be analyzed or visualized further using feature engineering methods for more detailed analysis. The next step involves training a machine learning algorithm using the feature extraction obtained through deterministic methods[1–6,91]. This approach has demonstrated promising outcomes in predicting specific pathologies, and is particularly effective when applied to research datasets, generating significant interest among scientists who are interested in utilizing supervised learning-based prediction for eye movement disorder studies.

Deep learning is a powerful and robust approach that is well-suited for analyzing clinical data, especially time series data. Our previous study[7] compared the effectiveness of machine learning and deep learning on different

datasets with varying input and label variability. The results indicated that while machine learning performed well on biased datasets, its ability to generalize decreased in more realistic scenarios with higher signal-to-noise ratios and greater input and label variability. In contrast, our findings demonstrated that deep learning outperformed machine leaning methods in detecting learning disorders in real clinical settings. This present study represents a significant advancement in the field of eye movement analysis.

Conclusions

This paper presents a novel deep learning framework that effectively identifies various pathologies using eye movement data obtained during Remobi saccade and vergence tests. The framework integrates convolutional neural networks to analyze gaze data, enabling the incorporation of important contextual information such as stimulus latency (the delay between target onset and eye movement onset), confidence level (of the eye movement recording itself), and spatial coordinates, achieving a mean macro F1 score of up to 78.9%. Additionally, we introduce a hybrid architecture that significantly reduces computation costs by a factor of 3.36 times while maintaining strong generalization capabilities. Our findings demonstrate the potential usefulness of this method for predicting diverse pathologies in clinical settings.

Data availibility

The datasets generated during and/or analyzed during the current study are not publicly available due to the contract between Orasis and its clients, which enables the use of data solely for internal research conducted by Orasis, aiming to further improve software analysis through research efforts. However, the datasets are available from the corresponding author on reasonable request.

Received: 20 November 2023; Accepted: 19 July 2024
Published online: 13 September 2024

References

1. El Hmimdi, A. E., Ward, L. M., Palpanas, T. & Kapoula, Z. Predicting dyslexia and reading speed in adolescents from eye movements in reading and non-reading tasks: A machine learning approach. *Brain Sci.* **11**, 1337 (2021).
2. El Hmimdi, A. E., Ward, L. M., Palpanas, T., Garnot, S. F. & Kapoula, V. Z. Predicting dyslexia in adolescents from eye movements during free painting viewing. *Brain Sci.* **12**, 1031 (2022).
3. Jothi Prabha, A. & Bhargavi, R. Prediction of dyslexia from eye movements using machine learning. *IETE J. Res.* **68**, 814–823 (2022).
4. Nerušil, B., Polec, J., Škunda, J. & Kačur, J. Eye tracking based dyslexia detection using a holistic approach. *Sci. Rep.* **11**, 15687 (2021).
5. Nilsson Benfatto, M. *et al.* Screening for dyslexia using eye tracking during reading. *PLoS ONE* **11**, e0165508 (2016).
6. Vajs, I. A., Kvaščev, G. S., Papić, T. M. & Janković, M. M. Eye-tracking image encoding: Autoencoders for the crossing of language boundaries in developmental dyslexia detection. *IEEE Access* **11**, 3024–3033 (2023).
7. El Hmimdi, A. E., Kapoula, Z. & Garnot, S. F. Deep learning-based detection of learning disorders on a large scale dataset of eye movement records. *BioMedInformatics* **4**, 519–541. https://doi.org/10.3390/biomedinformatics4010029 (2024).
8. LeCun, Y. *et al.* Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems***2** (1989).
9. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems***25** (2012).
10. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).
11. Szegedy, C. *et al.* Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9 (2015).
12. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (2016).
13. Vaswani, A. *et al.* Attention is all you need. *Advances in neural information processing systems***30** (2017).
14. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. *et al.* Improving language understanding by generative pre-training. *OpenAI* (2018).
15. Radford, A. *et al.* Language models are unsupervised multitask learners. *OpenAI blog* **1**, 9 (2019).
16. Song, K., Tan, X., Qin, T., Lu, J. & Liu, T.-Y. Mass: Masked sequence to sequence pre-training for language generation. arXiv preprint arXiv:1905.02450 (2019).
17. Chen, M. *et al.* Generative pretraining from pixels. In *International conference on machine learning*, 1691–1703 (PMLR, 2020).
18. Bao, H., Dong, L., Piao, S. & Wei, F. Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254 (2021).
19. Xie, Z. *et al.* Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9653–9663 (2022).
20. He, K. *et al.* Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16000–16009 (2022).
21. Dai, Z., Liu, H., Le, Q. V. & Tan, M. Coatnet: Marrying convolution and attention for all data sizes. *Adv. Neural. Inf. Process. Syst.* **34**, 3965–3977 (2021).
22. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520 (2018).
23. Tu, Z. *et al.* Maxvit: Multi-axis vision transformer. In *European conference on computer vision*, 459–479 (Springer, 2022).
24. Zhang, J. *et al.* Xformer: Hybrid x-shaped transformer for image denoising. arXiv preprint arXiv:2303.06440 (2023).
25. Mekruksavanich, S. & Jitpattanakul, A. A hybrid convolution neural network with channel attention mechanism for sensor-based human activity recognition. *Sci. Rep.* **13**(1), 12067 (2023).
26. Li, G., Fang, Q., Zha, L., Gao, X. & Zheng, N. Ham: Hybrid attention module in deep convolutional neural networks for image classification. *Pattern Recogn.* **129**, 108785 (2022).
27. Hatamizadeh, A. *et al.* Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In *International MICCAI Brainlesion Workshop*, 272–284 (Springer, 2021).
28. Philippi, D., Rothaus, K. & Castelli, M. A vision transformer architecture for the automated segmentation of retinal lesions in spectral domain optical coherence tomography images. *Sci. Rep.* **13**, 517 (2023).

29. Graham, B. *et al.* Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF international conference on computer vision*, 12259–12269 (2021).
30. Pupil labs - eye tracking hardware and software solutions. https://pupil-labs.com/ [Accessed: (2024-02-22)].
31. Leigh, R. J. & Zee, D. S. *The neurology of eye movements* (Oxford University Press, USA, 2015).
32. Vernet, M. *Coordination des mouvements oculaires dans l'espace 3D chez l'homme: substrat cortical étudié par TMS*. Ph.D. thesis, Université Pierre et Marie Curie-Paris VI (2009).
33. Yang, Q., Bucci, M. P. & Kapoula, Z. The latency of saccades, vergence, and combined eye movements in children and in adults. *Invest. Ophthalmol. Vis. Sci.* **43**, 2939–2949 (2002).
34. Yang, Q. & Kapoula, Z. Saccade-vergence dynamics and interaction in children and in adults. *Exp. Brain Res.* **156**, 212–223 (2004).
35. Bucci, M. P. *et al.* Normal speed and accuracy of saccade and vergence eye movements in dyslexic reader children. *J. Ophthalmol.* **2009**(1), 32514 (2009).
36. Ward, L. M. & Kapoula, Z. Dyslexics' fragile oculomotor control is further destabilized by increased text difficulty. *Brain Sci.* **11**, 990 (2021).
37. Ward, L. M. & Kapoula, Z. Differential diagnosis of vergence and saccade disorders in dyslexia. *Sci. Rep.* **10**, 22116 (2020).
38. Ward, L. M. & Kapoula, Z. Creativity, eye-movement abnormalities, and aesthetic appreciation of magritte's paintings. *Brain Sci.* **12**, 1028 (2022).
39. Kapoula, Z. *et al.* Objective evaluation of vergence disorders and a research-based novel method for vergence rehabilitation. *Transl. Vis. Sci. Technol.* **5**, 8–8 (2016).
40. Nan, M., Trăscău, M., Florea, A. M. & Iacob, C. C. Comparison between recurrent networks and temporal convolutional networks approaches for skeleton-based action recognition. *Sensors* **21**, 2051 (2021).
41. Catling, F. J. & Wolff, A. H. Temporal convolutional networks allow early prediction of events in critical care. *J. Am. Med. Inform. Assoc.* **27**, 355–365 (2020).
42. Bednarski, B. P. *et al.* Temporal convolutional networks and data rebalancing for clinical length of stay and mortality prediction. *Sci. Rep.* **12**, 21247 (2022).
43. Bai, S., Kolter, J. Z. & Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018).
44. Hendrycks, D. & Gimpel, K. Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016).
45. Wang, Q. *et al.* Learning deep transformer models for machine translation. arXiv preprint arXiv:1906.01787 (2019).
46. Dosovitskiy, A. *et al.* An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020).
47. Baevski, A. & Auli, M. Adaptive input representations for neural language modeling. arXiv preprint arXiv:1809.10853 (2018).
48. iterative stratification. https://scikit.ml/api/skmultilearn.model_selection.iterative_stratification.html [Accessed: (2024-02-22)].
49. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017).
50. Opitz, J. & Burst, S. Macro f1 and macro f1. arXiv preprint arXiv:1911.03347 (2019).
51. Bautista, L. G. C. & Naval, P. C. Clrgaze: Contrastive learning of representations for eye movement signals. In *2021 29th European Signal Processing Conference (EUSIPCO)*, 1241–1245 (IEEE, 2021).
52. Bautista, L. G. C. & Naval, P. C. Gazemae: general representations of eye movements using a micro-macro autoencoder. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 7004–7011 (IEEE, 2021).
53. Singh, S., Pandey, S. K., Pawar, U. & Janghel, R. R. Classification of ECG arrhythmia using recurrent neural networks. *Proc. Comput. Sci.* **132**, 1290–1297 (2018).
54. Ruffini, G. *et al.* Deep learning with EEG spectrograms in rapid eye movement behavior disorder. *Front. Neurol.* **10**, 806 (2019).
55. Gao, D., Tang, X., Wan, M., Huang, G. & Zhang, Y. Eeg driving fatigue detection based on log-mel spectrogram and convolutional recurrent neural networks. *Front. Neurosci.* **17**, 1136609 (2023).
56. Tan, M. & Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, 6105–6114 (PMLR, 2019).
57. Dempster, A., Petitjean, F. & Webb, G. I. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Min. Knowl. Disc.* **34**, 1454–1495 (2020).
58. Ruiz, A. P., Flynn, M., Large, J., Middlehurst, M. & Bagnall, A. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Disc.* **35**, 401–449 (2021).
59. Faouzi, J. Time series classification: A review of algorithms and implementations. *Machine Learning (Emerging Trends and Applications)* (2022).
60. Rocket implementation. https://github.com/angus924/rocket [Accessed: (2024-02-22)].
61. sklearn library. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeClassifierCV.html [Accessed: (2024-02-22)].
62. Wilcoxon signed-rank test. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wilcoxon.html [Accessed: (2024-03-29)].
63. Elbattah, M., Guérin, J.-L., Carette, R., Cilia, F. & Dequen, G. Nlp-based approach to detect autism spectrum disorder in saccadic eye movement. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1581–1587 (IEEE, 2020).
64. Chen, S. & Zhao, Q. Attention-based autism spectrum disorder screening with privileged modality. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1181–1190 (2019).
65. Jiang, M. & Zhao, Q. Learning visual attention to identify people with autism spectrum disorder. In *Proceedings of the IEEE international conference on computer vision*, 3267–3276 (2017).
66. Ahmed, I. A. *et al.* Eye tracking-based diagnosis and early detection of autism spectrum disorder using machine learning and deep learning techniques. *Electronics* **11**, 530 (2022).
67. Tao, Y. & Shyu, M.-L. Sp-asdnet: Cnn-lstm based asd classification model using observer scanpaths. In *2019 IEEE International conference on multimedia & expo workshops (ICMEW)*, 641–646 (IEEE, 2019).
68. Vajs, I., Ković, V., Papić, T., Savić, A. M. & Janković, M. M. Dyslexia detection in children using eye tracking data based on vgg16 network. In *2022 30th European Signal Processing Conference (EUSIPCO)*, 1601–1605 (IEEE, 2022).
69. Harisinghani, A. *et al.* Classification of alzheimer's using deep-learning methods on webcam-based gaze data. *Proceedings of the ACM on Human-Computer Interaction* **7**, 1–17 (2023).
70. Sun, J., Liu, Y., Wu, H., Jing, P. & Ji, Y. A novel deep learning approach for diagnosing Alzheimer's disease based on eye-tracking data. *Front. Hum. Neurosci.* **16**, 972773 (2022).
71. Uribarri, G., von Huth, S. E., Waldthaler, J., Svenningsson, P. & Fransén, E. Deep learning for time series classification of parkinson's disease eye tracking data. arXiv preprint arXiv:2311.16381 (2023).
72. Zemblys, R., Niehorster, D. C. & Holmqvist, K. gazenet: End-to-end eye-movement event detection with deep neural networks. *Behav. Res. Methods* **51**, 840–864 (2019).
73. Lee, S.-W. *et al.* Detection of abnormal behavior with self-supervised gaze estimation. arXiv preprint arXiv:2107.06530 (2021).
74. Uppal, K., Kim, J. & Singh, S. Decoding attention from gaze: A benchmark dataset and end-to-end models. In *Annual Conference on Neural Information Processing Systems*, 219–240 (PMLR, 2023).
75. Cole, Z. J., Kuntzelman, K. M., Dodd, M. D. & Johnson, M. R. Convolutional neural networks can decode eye movement data: A black box approach to predicting task from eye movements. *J. Vis.* **21**, 9–9 (2021).

76. Zhao, D., Tang, F., Si, B. & Feng, X. Learning joint space-time-frequency features for eeg decoding on small labeled data. *Neural Netw.* **114**, 67–77 (2019).
77. Borra, D., Mondini, V., Magosso, E. & Müller-Putz, G. R. Decoding movement kinematics from eeg using an interpretable convolutional neural network. *Comput. Biol. Med.* **165**, 107323 (2023).
78. Borra, D., Magosso, E., Castelo-Branco, M. & Simões, M. A Bayesian-optimized design for an interpretable convolutional neural network to decode and analyze the p300 response in autism. *J. Neural Eng.* **19**, 046010 (2022).
79. Englebert, A. *et al.* Explaining through transformer input sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 806–815 (2023).
80. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprintarXiv:1312.6034 (2013).
81. Smilkov, D., Thorat, N., Kim, B., Viégas, F. & Wattenberg, M. Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017).
82. Sundararajan, M., Taly, A. & Yan, Q. Axiomatic attribution for deep networks. In *International conference on machine learning*, 3319–3328 (PMLR, 2017).
83. Selvaraju, R. R. *et al.* Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626 (2017).
84. Wang, H. *et al.* Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 24–25 (2020).
85. Jiang, P.-T., Zhang, C.-B., Hou, Q., Cheng, M.-M. & Wei, Y. Layercam: Exploring hierarchical class activation maps for localization. *IEEE Trans. Image Process.* **30**, 5875–5888 (2021).
86. Petsiuk, V., Das, A. & Saenko, K. Rise: Randomized input sampling for explanation of black-box models. arXiv preprint arXiv:1806.07421 (2018).
87. Zeiler, M. D. & Fergus, R. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, 818–833 (Springer, 2014).
88. Negi, S. & Mitra, R. Fixation duration and the learning process: An eye tracking study with subtitled videos. *J. Eye Movem. Res.*https://doi.org/10.16910/jemr.13.6.1 *(2020).*
89. Bylinskii, Z., Borkin, M. A., Kim, N. W., Pfister, H. & Oliva, A. Eye fixation metrics for large scale evaluation and comparison of information visualizations. In *Eye Tracking and Visualization: Foundations, Techniques, and Applications. ETVIS 2015 1*, 235–255 (Springer, 2017).
90. Wegner-Clemens, K., Rennig, J., Magnotti, J. F. & Beauchamp, M. S. Using principal component analysis to characterize eye movement fixation patterns during face viewing. *J. Vis.* **19**, 2–2 (2019).
91. Asvestopoulou, T. *et al.* Dyslexml: Screening tool for dyslexia using machine learning. arXiv preprint arXiv:1903.06274 (2019).

Acknowledgements

The authors thank Vivien Sainte Fare Garnot for providing substantial comments, as well as for sharing his AI expertise in other fields. This work was granted access to the HPC resources of IDRIS under the allocation 2024-AD011014231 made by GENCI.

Author contributions

A.E.E.H. design the architecture, conducted the experiments and co-wrote the manuscript. T.P. supervised the study and co-wrote the manuscript. Z.K. supervised the study and and co-wrote the manuscript.

Competing interests

Zoï Kapoula is the founder of Orasis-EAR and the inventor of the REMOBI technology. Alae eddine El Hmimdi PhD work has been funded by Orasis-EAR and ANRT.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-024-68056-9.

Correspondence and requests for materials should be addressed to Z.K.