Regular Article

EyeTrackDL: A Robust Deep Learning Framework for Saccade Detection via Simulated Data Augmentation

Ha Ngoc Khoan[1,3], Tran Van Nghia[2], Le Ky Bien[3]

[1] 108 Central Military Hospital, 1 Tran Hung Dao street, Hanoi Vietnam
[2] Air Force - Air Defense Technical Institute, 166 Hoang Van Thai Street, Hanoi, Vietnam
[3] Academy of Military Science and Technology, 17 Hoang Sam street, Hanoi, Vietnam

Correspondence: Tran Van Nghia, nghiamosmipt@gmail.com
Communication: received 20 June 2025, revised 06 August 2025, accepted 23 August 2025
Online publication: 13 September 2025, Digital Object Identifier: 10.21553/rev-jec.409

Abstract– **Saccade detection is a fundamental task in visual behavior analysis and vestibular diagnostics. However, video Head Impulse Test (vHIT) recordings are often noisy, heterogeneous, and affected by class imbalance, particularly for covert saccades. In this paper, we propose EyeTrackDL, a lightweight yet effective deep learning framework based on a multilayer perceptron (MLP) architecture for classifying three types of eye movements: non-saccades, overt saccades, and covert saccades. Input signals are preprocessed using a fourth-order Butterworth filter, and two high-level features (onset time and duration) are extracted per saccade. To address data scarcity and imbalance, we apply SMOTE resampling and incorporate synthetic data generated from a kinematic vestibulo-ocular reflex (VOR) model. The model is evaluated using K-fold cross-validation (K = 2 to 10) on both real and simulated datasets. Results show that EyeTrackDL achieves an average accuracy of up to 96.5% on simulated data and approximately 83% in the real data, with significant improvements in the sensitivity of the covert saccades. Our findings demonstrate the potential of integrating simulation-based augmentation and class balancing for robust saccade detection in clinical environments.**

Keywords– **Saccade detection, video Head Impulse Test (vHIT), multilayer perceptron (MLP), simulated vestibular data, SMOTE, classification of eye movements, class imbalance, deep learning in healthcare.**

1 Introduction

1.1 Overview

Saccadic eye movements are rapid, ballistic shifts in gaze that enable the fovea to fixate on new targets in the visual field. They constitute a critical mechanism in the human oculomotor system, ensuring efficient visual exploration and stable perception during active gaze shifts [1]. Given their diagnostic relevance, accurate saccade detection has become increasingly important in clinical and cognitive neuroscience [2, 3].

In clinical contexts, tools such as the *video Head Impulse Test (vHIT)* are commonly used to assess vestibulo-ocular reflex (VOR) by capturing coordination between head and eye movements. The ability to distinguish between *overt* and *covert* saccades in vHIT recordings is particularly crucial, as covert saccades often reflect subtle compensatory mechanisms in patients with vestibular dysfunction [4]. However, the detection and classification of these saccades is a nontrivial task, owing to various factors such as sensor noise, patient variability, and the brief and low-amplitude characteristics of covert saccades.

Traditional algorithms for saccade detection have relied on velocity thresholds [5], dispersion metrics, or rule-based heuristics [6]. While these methods can be effective under controlled conditions, they often exhibit sensitivity to measurement artifacts and require careful parameter tuning, limiting their applicability in clinical practice. Furthermore, the rise of mobile and embedded eye-tracking systems introduces additional challenges regarding data quality and consistency.

Recent advances in deep learning have demonstrated promising results in automating saccade detection, achieving significant improvements in accuracy, generalizability, and noise robustness [7, 8]. Convolutional and recurrent neural networks have been increasingly successful in processing continuous gaze signals. However, existing models still face key limitations: (i) reliance on large annotated datasets, which are scarce in clinical settings; (ii) low sensitivity to rare saccade types, particularly covert saccades; and (iii) limited performance when applied to data with high inter-patient variability.

In this work, we present **EyeTrackDL**, a compact and robust deep learning framework for saccade classification in vHIT recordings. The framework combines real clinical data with simulated vestibular signals generated from a kinematic VOR model and employs SMOTE-based data balancing to address class imbalance. By leveraging high-level, interpretable features and a multilayer perceptron (MLP) architecture, EyeTrackDL achieves strong performance under data-limited and noise-prone conditions, making it suitable for real-world clinical deployment.

1.2 Issue Statement

Despite the increasing adoption of eye-tracking technologies like the video Head Impulse Test (vHIT) in vestibular diagnostics, saccade detection still relies primarily on manual review or threshold-based heuristics. These conventional methods often perform poorly under the noisy, non-uniform conditions characteristic of real-world clinical data.

Most current deep learning models for saccade classification have been trained on experimental datasets from cognitive science or behavioral studies, limiting their generalizability to clinical recordings. This creates two critical problems:

1) **Lack of high-quality annotated data**, particularly for covert saccades that are subtle, brief, and frequently underrepresented in datasets.
2) **Severe class imbalance** that biases learning algorithms toward dominant classes (e.g., non-saccades or overt saccades).

Moreover, prior studies have rarely exploited *simulated data* to augment training sets, despite its potential to generate diverse, controlled examples. Consequently, there is a need for a lightweight yet robust deep learning framework that leverages both real and synthetic data to address the practical challenges of saccade classification in noisy clinical settings.

1.3 Our Contributions

To overcome these limitations, we introduce **EyeTrackDL**, an efficient deep learning framework for accurate saccade detection and classification in clinical vHIT recordings. The main innovations of our approach are:

1) We developed a preprocessing and labeling pipeline that extracts saccadic events from eye-head velocity data using physiologically informed thresholding and a fourth-order Butterworth filter, classifying events into non-saccades, covert saccades, and overt saccades.
2) We implemented a compact Multilayer Perceptron (MLP) architecture that operates on two key features–saccade onset time and duration–enabling real-time performance in clinical applications.
3) We created a hybrid training dataset combining real vHIT recordings with synthetic data generated from a kinematic VOR model, facilitating structured augmentation of underrepresented classes.
4) We employed *SMOTE*-based resampling to address class imbalance, significantly improving covert saccade detection while maintaining performance on dominant classes.
5) We conducted extensive evaluation using nine k-fold cross-validation configurations (K = 2 to 10), reporting per-class metrics (accuracy, sensitivity, specificity) and analyzing the impact of data source (real vs. synthetic) on performance.

Collectively, these contributions demonstrate that **EyeTrackDL** achieves both high performance (96.5% average accuracy) and clinical utility, offering a generalizable and scalable solution for vestibular diagnostics.

The remainder of this paper is structured as follows:

- **Section 2 – Related Work** reviews saccade detection methods from classical threshold-based approaches to recent deep learning techniques, identifying key gaps in existing research.
- **Section 3: Background** covers the physiology of saccadic eye movements, clinical measurement techniques, simulated data applications, and the theoretical basis of our model.
- **Section 4: Methodological Framework** details the EyeTrackDL pipeline, including signal preprocessing, saccade labeling, data augmentation, and model architecture.
- **Section 5: Experimental Setup** describes the datasets (real and simulated), evaluation metrics, and experimental environment.
- **Section 6: Results and Discussion** analyzes classification performance across folds and classes, examines SMOTE and simulation effects, and discusses findings with limitations.
- **Section 7: Conclusion** summarizes contributions and suggests future research directions.

2 Related Works

Saccade detection is crucial for both visual behavior analysis and clinical diagnostics. Early approaches relied on velocity-based thresholds, such as the Engbert and Mergenthaler algorithm [5], which identifies microsaccades using adaptive velocity thresholds. Although straightforward to implement, these methods are sensitive to measurement noise and require manual parameter tuning. Otero-Millan *et al*. [9] developed an unsupervised clustering approach that estimates saccade onset and offset, but this method still performs poorly in high-noise conditions.

To address these limitations, researchers have increasingly turned to deep learning approaches. U'n'Eye (Bellet *et al*., 2019) [7] proposed a CNN-based classifier that processes horizontal and vertical eye velocity signals, achieving human-level performance on multiple benchmark datasets. Their U-Net-inspired architecture demonstrates strong generalization across unseen subjects and experimental conditions while maintaining robustness in noisy environments.

Alternative deep learning approaches have shown promising results. Startsev *et al*. [8] combined 1D CNNs with bidirectional LSTMs to model temporal dependencies in gaze data. Zemblys *et al*. [10] employed a generative adversarial network (GAN) to augment limited labeled datasets. Mihali *et al*. [11] introduced a Bayesian generative model that provides both microsaccade predictions and uncertainty estimates. While probabilistic approaches like those of Daye and Optican [12] and threshold-based methods such as Pekkanen and Lappi [13] offer valuable baselines, they remain dependent on parametric tuning and perform best under specific experimental conditions.

Although deep neural networks achieve superior accuracy and generalizability compared to traditional methods, they still require substantial annotated training data. A particular challenge is class imbalance, which is especially prevalent in real-world saccade datasets and complicates classifier optimization. Our approach addresses these limitations through two complementary strategies: (1) simulated data augmentation and (2) SMOTE-based resampling, which together improve detection robustness in data-scarce scenarios.

However, many prior works have focused primarily on overt or microsaccades and have not explicitly targeted covert saccade detection, which is clinically more challenging due to its subtle kinematic profile and lower signal amplitude. Moreover, most deep learning models rely on high-dimensional raw signals (e.g., full eye velocity traces), requiring considerable preprocessing and compute resources. In contrast, our model demonstrates that using only two interpretable features can yield competitive performance, particularly when supported by domain-specific simulation and class balancing.

To contextualize our approach, we directly compare it against both traditional threshold-based and contemporary deep learning methods in Table I. While U'n'Eye and LSTM-based models capture fine-grained temporal dynamics, they typically lack explicit mechanisms for handling rare-event classification and require more training time and data. Our MLP-based approach trades off sequence modeling for simplicity and speed, while still maintaining robust detection of covert saccades—an area where most prior models struggle.

Table I summarizes representative saccade detection methods, from classical threshold-based approaches to contemporary deep learning models. While early techniques like those of Engbert and Mergenthaler [5] and Otero-Millan *et al*. [9] provide computational simplicity and real-time performance, they exhibit limited noise robustness and poor temporal resolution for event duration.

Deep learning approaches like those of Startsev *et al*. [8] and Bellet *et al*. [7] achieve superior accuracy and temporal modeling capabilities. However, these methods still require large amounts of precisely labeled training data and fail to explicitly address class imbalance, potentially compromising their sensitivity to rare saccadic events like covert saccades.

Our proposed framework advances the field in three key dimensions:

- Incorporates simulated vestibular data generated from a kinematic VOR model, enabling controlled augmentation of underrepresented classes
- Utilizes SMOTE-based class balancing to significantly enhance detection sensitivity for minority-class events
- Employs a computationally efficient MLP architecture that processes high-level features (onset and duration), eliminating the need for high-resolution temporal sequences or sensor-specific calibration
- Demonstrates competitive or superior covert saccade detection performance compared to existing methods, while maintaining low training time and model complexity—an advantage for deployment in time-sensitive or embedded clinical settings

Together, these strategies achieve performance comparable to or better than deeper architectures while demonstrating superior generalizability to noisy, imbalanced clinical datasets.

3 Background

3.1 Physiology of Saccadic Eye Movements and Clinical Measurement Techniques

Saccadic eye movements represent rapid, ballistic shifts in gaze position that realign the fovea with visual targets. With amplitudes typically ranging from 1° to 20° and durations between 20–80 ms (depending on target distance and urgency), these movements constitute a fundamental component of the human oculomotor system. They facilitate efficient visual scanning and enable the acquisition of behaviorally relevant sensory information.

Neurophysiologically, saccades are controlled by a distributed network comprising the vestibular nuclei, superior colliculus, cerebellum, frontal eye fields, and associated cortical-subcortical circuits. This network integrates vestibular and visual sensory inputs to generate coordinated motor commands for the extraocular muscles.

During normal head movements, retinal image displacement is counteracted by the VOR, which generates compensatory eye movements opposite to head motion to maintain visual stability. In vestibular hypofunction, however, the VOR's compensatory capacity becomes impaired, leading to blurred vision during head movements. To restore gaze stability, the central nervous system produces *compensatory saccades* that realign the fovea with the intended visual target.

These compensatory saccades are generally classified as follows:

- **Overt saccades**: Occur after head movement cessation and are readily detectable by video recording systems.
- **Covert saccades**: Occur during active head motion, appearing more subtle due to overlap with the VOR response and consequently being more challenging to detect.

In clinical practice, the *vHIT* serves as the gold standard for vestibular assessment. The test delivers brief, unpredictable head impulses while simultaneously measuring head and eye angular velocities via infrared video-oculography and inertial sensors integrated into specialized goggles. Clinicians analyze the resulting eye-head velocity profiles to quantify VOR gain and detect compensatory saccades.

Nevertheless, several factors limit the accuracy of vHIT recordings:

- Signal artifacts from improper goggle fit, eyelid interference, or ambient light contamination
- Calibration errors causing misalignment between eye and head velocity baselines

Table I
Comparison of Related Methods in Saccade Detection

Study	Method Type	Label Type	Strengths	Limitations
Engbert & Mergenthaler (2006) [5]	Unsupervised	Microsaccade only	Simple, adaptive thresholds	Poor under noise, only onset detection
Otero-Millan *et al.* (2014) [9]	Unsupervised	Onset + Offset	Detects full saccade duration	Fails in high-noise recordings
Daye & Optican (2014) [12]	Probabilistic	Onset + Offset	Particle filter modeling	Requires parameter tuning
Pekkanen & Lappi (2017) [13]	Supervised	Onset + Offset	Handles smooth pursuit scenarios	Requires dual-threshold tuning
Zemblys *et al.* (2018) [10]	Deep Learning + GAN	Full labels	Generates training data effectively	Limited benchmarking
Startsev *et al.* (2018) [8]	CNN + BiLSTM	Full labels	Models temporal patterns well	Lower performance near boundaries
Mihali *et al.* (2017) [11]	Bayesian Model	Onset probabilities	Captures uncertainty	Requires probabilistic tuning
Bellet *et al.* (2019) [7]	CNN (U-Net)	Full labels	Human-level accuracy, robust	Class imbalance

- Challenges in detecting covert saccades due to their temporal overlap with head impulses and smaller amplitudes

These limitations underscore the need for robust automated saccade detection algorithms capable of reliably distinguishing between covert and overt saccades, particularly in noisy clinical recordings with ambiguous signals.

3.2 Simulated Vestibular Data and Its Role in Biomedical Machine Learning

Neuro-vestibular research faces significant data collection challenges, including reliance on specialized equipment, high inter-patient variability, and labeling inconsistencies from subjective annotations. Furthermore, available datasets typically exhibit severe class imbalance, with clinically important categories like covert saccades being both underrepresented and technically challenging to capture. These constraints have established simulated data as a strategic solution for augmenting machine learning pipelines in vestibular applications.

The vestibulo-ocular reflex (VOR) is a critical physiological mechanism that stabilizes gaze during head movements by generating compensatory eye movements. Modeling this reflex provides a principled foundation for generating synthetic eye movement signals.

Through computational modeling of vestibulo-ocular reflex (VOR) dynamics, synthetic datasets can be systematically generated with controlled parameters. Key simulation variables include head angular velocity, saccadic amplitude, event timing, and onset duration.

We implemented a kinematic VOR model, which uses predefined mathematical relationships between head motion and resulting eye trajectories. This approach enables the creation of physiologically plausible saccade patterns by varying motion profiles and timing parameters under controlled noise conditions.

This methodology enables precise generation of both covert and overt saccades while ensuring controlled variability and full reproducibility.

Simulated vestibular data additionally functions as an effective augmentation strategy, expanding the training distribution and improving model generalizability across diverse inputs. For our MLP classifier, we combined artificially generated sequences with real vHIT recordings during training, ultimately enhancing both classification accuracy and robustness in differentiating saccade types.

3.3 Deep Learning Foundations for Multiclass Classification

Deep learning, a machine learning subfield, utilizes artificial neural networks (ANNs) with multiple processing layers to model complex nonlinear data relationships [14]. The MLP represents one of the most fundamental architectures, demonstrating particular effectiveness for structured input data and discrete classification tasks.

In the context of multiclass classification, the objective is to learn a function $f : \mathbb{R}^d \to \{1, 2, ..., C\}$ that maps each input vector $\mathbf{x} \in \mathbb{R}^d$ to one of C discrete classes. An MLP achieves this by learning intermediate representations through stacked layers of neurons, each performing a linear transformation followed by a nonlinear activation.

The forward pass of an MLP with L hidden layers can be mathematically expressed as

$$\begin{aligned}
\mathbf{h}^{(1)} &= \phi\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right) \\
\mathbf{h}^{(2)} &= \phi\left(\mathbf{W}^{(2)}\mathbf{h}^{(1)} + \mathbf{b}^{(2)}\right) \\
&\vdots \\
\mathbf{h}^{(L)} &= \phi\left(\mathbf{W}^{(L)}\mathbf{h}^{(L-1)} + \mathbf{b}^{(L)}\right) \\
\hat{\mathbf{y}} &= \text{Softmax}\left(\mathbf{W}^{(L+1)}\mathbf{h}^{(L)} + \mathbf{b}^{(L+1)}\right),
\end{aligned} \tag{1}$$

here, $\mathbf{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$ and $\mathbf{b}^{(l)} \in \mathbb{R}^{d_l}$ are the weight matrix and bias vector of layer l, $\phi(\cdot)$ denotes the activation function—commonly the **Rectified Linear Unit (ReLU)** defined as $\phi(z) = \max(0, z)$—and $\hat{\mathbf{y}} \in \mathbb{R}^C$ represents the model's predicted class probabilities.

The Softmax function ensures that the output values are interpretable as probabilities

$$\hat{y}_i = \frac{\exp(z_i)}{\sum_{j=1}^{C} \exp(z_j)}, \quad \text{for } i = 1, \ldots, C. \tag{2}$$

The learning objective is to minimize a loss function that quantifies the discrepancy between the predicted and true class labels. For multiclass problems, the **categorical cross-entropy loss** is commonly used

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i), \tag{3}$$

where $y_i \in \{0, 1\}$ is the one-hot encoded ground truth label, and $\hat{y}_i \in [0, 1]$ is the predicted probability for class i.

The model parameters $\Theta = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^{L+1}$ are optimized by computing gradients of the loss function with respect to each parameter using **backpropagation** and updating them via an optimization algorithm such as Stochastic Gradient Descent (SGD) or Adam [15].

One significant challenge in real-world classification is the **class imbalance problem**, where certain classes are underrepresented in the training data. This leads to biased decision boundaries and poor recall on minority classes. In such cases, algorithm-level techniques like cost-sensitive learning, or data-level approaches such as resampling are used. Among the most effective data-level methods is the **Synthetic Minority Over-sampling Technique (SMOTE)** [16], which generates synthetic examples of minority class instances by interpolating between neighboring samples in the feature space.

In multiclass classification, SMOTE can be extended to address multiple minority classes by independently resampling each underrepresented category to achieve near-balanced class distributions. This approach provides more uniform learning signals across all classes, thereby enhancing the model's ability to recognize rare patterns during inference.

4 Methodological Framework

4.1 Overview of the EyeTrackDL Workflow

The proposed EyeTrackDL framework is structured into four major components:

1) Data Preparation
2) Saccade Detection and Labeling
3) Data Engineering
4) Model Training and Validation

Each processing stage converts raw eye-head impulse signals into structured classification inputs. The pipeline begins by applying a fourth-order Butterworth low-pass filter to raw vHIT recordings, followed by differentiation to compute both eye and head velocities.

Saccadic events are identified when eye velocity surpasses a 30°/s threshold and duration falls within the 10–80 ms range. These detected segments are then classified according to their head velocity at onset and encoded as feature vectors.

To mitigate class imbalance and enhance model training, we implement a streamlined SMOTE approach on the real dataset while supplementing it with synthetic signals generated from a parameterized VOR model. The resulting combined dataset trains the proposed MLP classifier, with performance assessed via K-fold cross-validation. An overview of the entire pipeline is provided in Figure 1.

4.2 Data Preparation

This study utilized two distinct datasets: (1) a clinical dataset comprising 760 eye-head movement recordings acquired from healthy subjects using the ICS Impulse system, and (2) a synthetic dataset containing 34,000 signals generated from a kinematic VOR model.

Each data file included three time-series channels: timestamp, eye position, and head position. We processed 599 real signal files through a standardized preprocessing pipeline. To attenuate high-frequency noise, all signals underwent fourth-order Butterworth low-pass filtering with a 20 Hz cutoff frequency.

Following filtration, we computed eye and head velocities using numerical differentiation based on the mean sampling interval:

$$v(t) = \frac{x(t + \Delta t) - x(t)}{\Delta t}. \tag{4}$$

This preprocessing stage was essential for maintaining key kinematic features (e.g., peak velocity and temporal resolution) crucial for saccade detection while effectively suppressing measurement noise.

4.3 Saccade Detection and Labeling

Following preprocessing, we analyzed eye and head velocity profiles to identify saccadic events using velocity thresholding. Detection criteria required eye velocity to exceed 30°/s for durations between 10-80 ms. Each detected event was then classified based on its concurrent head velocity:

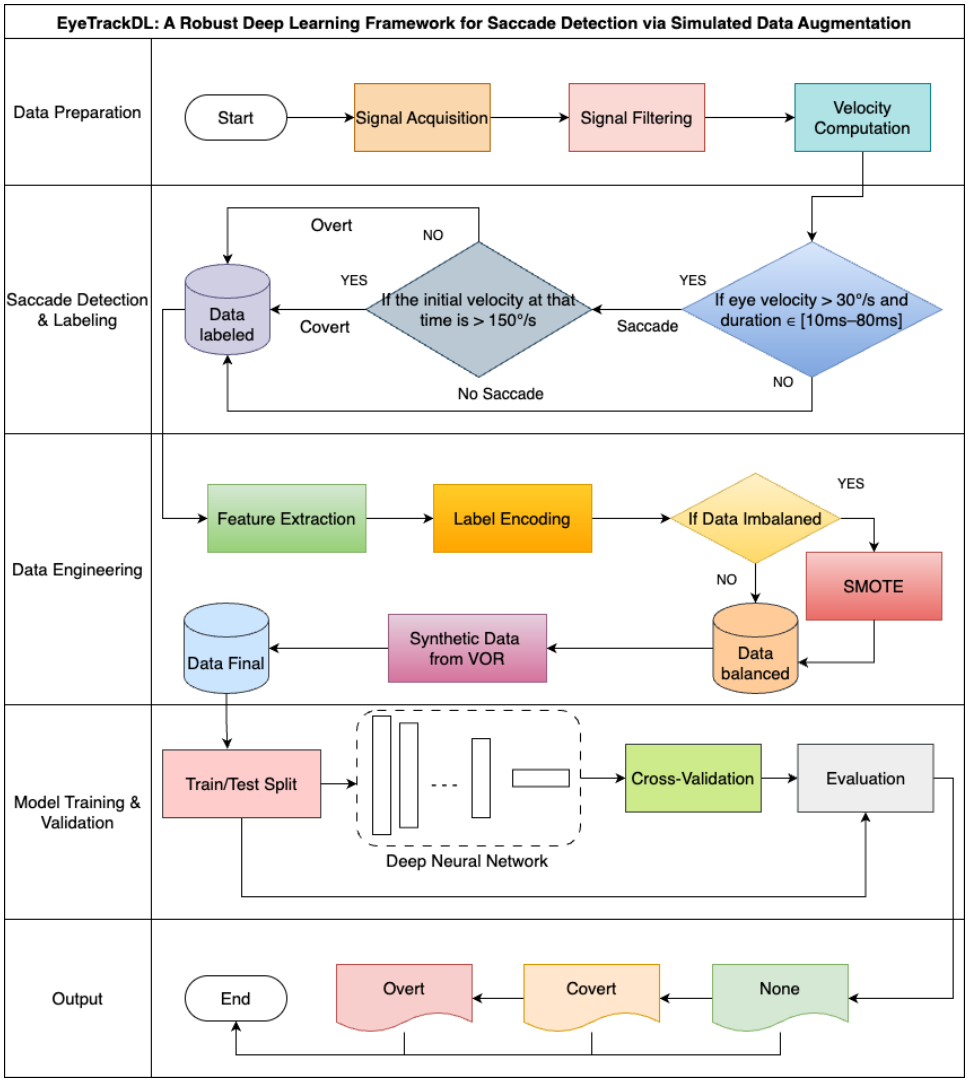

Figure 1. The proposed EyeTrackDL framework automates saccade classification through a four-stage workflow: data preparation, saccade detection and labeling, data engineering, and model training and validation. Both real and simulated vHIT signals are processed to detect saccadic events, extract key temporal features, and train a multilayer perceptron (MLP) classifier with enhanced generalizability.

- If the head velocity at the end of the event exceeded 150 deg/s, it was labeled as a *covert saccade*.
- If the head velocity was less than or equal to 150 deg/s, the event was labeled as an *overt saccade*.
- If no saccade was detected, the signal was labeled as *no saccade*.

Each detected event was encoded as a feature vector comprising saccade onset time and duration. These extracted features served as inputs for model training and evaluation. The complete detection and classification pipeline is formally described in Algorithm 1.

4.4 Data Engineering

In this section, each identified event was converted into a fixed-length feature vector. For each event (including non-saccade), the following two features were extracted:

- x_1: Onset time of the saccade (in seconds)
- x_2: Duration of the saccade (in seconds)

These vectors formed the input dataset used in model training. However, a significant class imbalance was observed among the three classes (non-saccade, covert, overt). To mitigate this issue, the **Synthetic Minority Oversampling Technique (SMOTE)** was applied.

In SMOTE, new synthetic samples are generated for minority classes by interpolating between existing samples and their neighbors. Given a minority class sample x_i and one of its nearest neighbors x_{nn}, a new point is generated as follows

$$x_{\text{new}} = x_i + \delta \cdot (x_{\text{nn}} - x_i), \tag{5}$$

where $\delta \sim \mathcal{U}(0,1)$ is a random number sampled from the uniform distribution.

This interpolation is repeated until the number of synthetic samples matches the majority class, resulting in a balanced dataset suitable for training. The balanced dataset was subsequently used in K-Fold cross-validation during model evaluation.

4.5 Model Architecture and Training Procedure

The classification model employed in this study is a fully connected MLP. Its structure consists of:

Algorithm 1 Saccade Detection and Labeling Algorithm

Require: Eye velocity $V_{eye}[t]$, Head velocity $V_{head}[t]$, Time vector $T[t]$
Require: Thresholds: $v_{eye} = 30$ deg/s, $v_{head} = 150$ deg/s, $d_{min} = 10$ ms, $d_{max} = 80$ ms
Ensure: Set of saccades with labels and features

Initialize empty list $Saccades \leftarrow []$
$i \leftarrow 1$
while $i \leq \text{length}(V_{eye})$ **do**
  if $|V_{eye}[i]| > v_{eye}$ **then**
    $start\_time \leftarrow T[i]$
    $duration \leftarrow 0$
    while $|V_{eye}[i]| > v_{eye}$ and $i \leq \text{length}(V_{eye})$ **do**
      $duration \leftarrow duration + \Delta t$
      $i \leftarrow i + 1$
    end while
    if $d_{min} \leq duration \leq d_{max}$ **then**
      if $|V_{head}[i-1]| > v_{head}$ **then**
        $label \leftarrow$ Covert
      else
        $label \leftarrow$ Overt
      end if
      Append $(start\_time, duration, label)$ to $Saccades$
    end if
  else
    $i \leftarrow i + 1$
  end if
end while
if $Saccades$ is empty **then**
  Append $(0, 0, NoSaccade)$ to $Saccades$
end if
return $Saccades$

- An input layer with 2 neurons (representing onset time and duration)
- Two hidden layers with 20 and 10 neurons respectively, each using the ReLU activation function
- A softmax output layer with 3 neurons (for the classes: non-saccade, covert, overt)

This intentionally lightweight architecture was selected to prioritize low latency, ease of deployment in clinical settings, and interpretability. While modern architectures such as CNNs or LSTMs may offer higher representational capacity, our objective was to evaluate whether a simple model could achieve clinically meaningful performance using minimal, robust features.

In low-data regimes, complex architectures are also prone to overfitting, especially when training data is noisy or imbalanced. Thus, we chose to begin with a compact MLP baseline before introducing additional model complexity in future work.

We systematically evaluated cross-validation configurations using K-fold cross-validation with K ranging from 2 to 10. For each K value, the dataset was partitioned into K equal folds, with the model trained on $K-1$ folds and validated on the remaining fold.

The choice of varying K from 2 to 10 allows us to assess performance stability across different data splits,

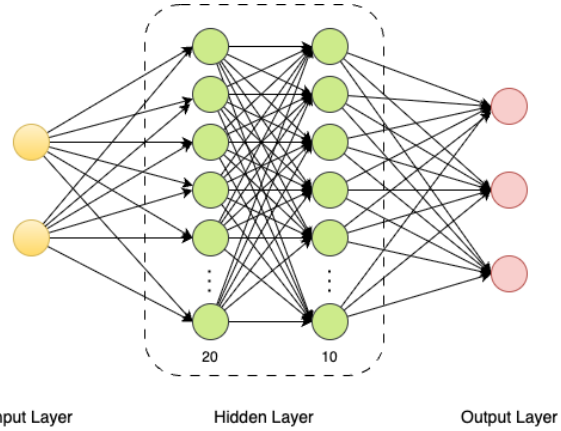

Figure 2. Architecture of the proposed MLP classifier: two input features, two hidden layers (20 and 10 neurons), and a softmax output layer for three-class prediction.

from minimal training data (e.g., $K = 2$) to finer-grained validation ($K = 10$). This approach also helps identify a trade-off point where generalization and training cost are optimally balanced.

Classification metrics (accuracy, sensitivity, and specificity) were computed per fold and averaged across all folds. The optimal K^* was determined by balancing classification performance against computational efficiency. The model architecture is presented in Figure 2.

5 Experimental Setup

5.1 Experimental Environment

All experiments were performed in MATLAB R2020b, utilizing the built-in `patternnet` function for model training. The complete processing pipeline—including signal preprocessing, model training, and performance evaluation—was implemented natively in MATLAB. Detailed hardware and software specifications are provided in Table II.

5.2 Dataset Description

Two datasets were utilized in this study:

- **Clinical dataset:** Consisted of 760 eye-head movement recordings obtained from healthy subjects using the ICS Impulse system. Each recording contained three time-series variables (timestamp, eye position, and head position) stored in plain-text format.
- **Synthetic dataset:** Comprised 34,000 simulated signals generated from a kinematic VOR model. The model adjusted key parameters (amplitude, velocity, and latency) to reproduce diverse saccadic dynamics.

All signals underwent temporal and amplitude normalization. The synthetic data served dual purposes: addressing class imbalance through dataset balancing and enhancing model generalizability via structured augmentation. Each saccadic event was encoded as a 2D feature vector containing

Table II
Hardware configuration of the experimental system

Component	Specification
Processor	AMD Ryzen 9 9900X, 12 cores / 24 threads, 4.4–5.6 GHz
RAM	32 GB DDR5, 5600 MHz
Storage	1 TB SSD, PCIe Gen 4 x8
Software	MATLAB R2020b

$$X = \left[x_1, x_2\right], \tag{6}$$

where x_1 is the onset time and x_2 is the duration of the saccade.

All clinical recordings were fully anonymized prior to analysis, with no personally identifiable information retained. Data collection followed institutional ethical guidelines and complied with applicable data privacy regulations. No patient intervention or identifying metadata was involved in this retrospective analysis.

5.3 Evaluation Metrics

Model performance was assessed using three standard classification metrics: accuracy, sensitivity (recall), and specificity. These metrics were calculated per class and averaged across all cross-validation folds.

- **Accuracy**: The proportion of correctly classified samples relative to the total sample size
- **Sensitivity (Recall)**: The true positive rate for a given class
- **Specificity**: The true negative rate for a given class

Mathematically, the metrics are defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{7}$$

$$\text{Sensitivity}_i = \frac{TP_i}{TP_i + FN_i}, \tag{8}$$

$$\text{Specificity}_i = \frac{TN_i}{TN_i + FP_i}, \tag{9}$$

where TP, TN, FP, and FN denote the number of true positives, true negatives, false positives, and false negatives respectively for class i.

6 Results and Discussion

6.1 Impact of Data Augmentation via SMOTE

Class imbalance significantly degrades classifier performance. We addressed this by implementing the SMOTE, which creates synthetic minority-class samples through linear interpolation between nearest neighbors in feature space.

To assess SMOTE's effectiveness, we compared two training regimes: (1) using real data only, and (2) using real data augmented with SMOTE-generated samples. Table III presents the resulting classification accuracies across fold configurations $K \in \{2, ..., 10\}$.

The baseline model (trained exclusively on real data) achieved 80.34% average accuracy, whereas the SMOTE-augmented version attained 96.54% - a 16.2 percentage point improvement. This performance gain remained consistent across all fold configurations, demonstrating that SMOTE-based augmentation effectively expands the training distribution and substantially improves model generalizability, particularly for class-imbalanced and data-scarce scenarios.

However, it is important to note the significant performance gap between synthetic (96.5%) and real clinical data (83%), which may indicate potential overfitting to the synthetic distribution. This discrepancy suggests that while SMOTE effectively increases the training pool, the synthetic samples may not fully capture the variability and noise characteristics of real-world data. To mitigate this, future work could explore domain adaptation techniques, adversarial training strategies, or hybrid datasets that blend simulated and empirically collected examples to better align feature distributions.

6.2 Performance Analysis Across Classes and Configurations

We evaluated model performance both at the class level and across all cross-validation configurations. Table IV provides complete sensitivity, specificity, and training time metrics for both real and synthetic datasets across K-fold values (K = 2–10).

The model showed particularly robust performance for non-saccade classification across all configurations. While overt saccade detection maintained consistently high accuracy, sensitivity for covert saccades improved substantially with synthetic data augmentation. Computational time increased linearly with fold count K, as expected.

We further observed that the model was trained using only two input features (onset time and duration), which, while computationally efficient, may constrain its ability to distinguish subtle differences between saccade types. Incorporating additional temporal or kinematic features—such as amplitude, peak velocity, or curvature—may improve discriminative capacity, particularly for borderline or ambiguous cases. This represents a promising direction for future work.

6.3 Error Analysis and Observations

Although the model achieved strong overall performance, error analysis identified several important patterns:

- **Type II errors** predominantly affected covert saccades, which showed the lowest sensitivity (62.6%) in the real dataset. These errors likely stem

Table III
Classification Accuracy (%) Before and After SMOTE Augmentation

K-Fold	Accuracy (Real Data)	Accuracy (SMOTE-Augmented)
2	62.20	87.63
3	81.94	97.64
4	82.75	97.66
5	82.79	97.65
6	82.93	97.58
7	82.60	97.72
8	82.40	97.72
9	83.01	97.65
10	82.46	97.58
Mean	**80.34**	**96.54**

Table IV
Detailed Performance Metrics per Class and Configuration (%)

Source	K	Accuracy	Sens (Non)	Spec (Non)	Sens (Covert)	Spec (Covert)	Sens (Overt)	Spec (Overt)	Time (s)
Real	2	62.20	100.00	100.00	64.30	71.80	61.70	75.20	160.0
	3	81.94	100.00	100.00	65.56	91.68	83.23	82.78	182.3
	4	82.75	100.00	100.00	66.29	90.99	81.99	83.14	203.2
	5	82.79	100.00	100.00	66.83	91.32	82.72	83.41	218.6
	6	82.93	100.00	99.93	66.66	91.09	82.20	83.39	238.4
	7	82.60	100.00	100.00	64.63	91.58	83.17	82.34	287.4
	8	82.40	100.00	100.00	64.76	91.25	82.50	82.37	294.6
	9	83.01	100.00	100.00	66.35	91.33	82.66	83.19	311.5
	10	82.46	100.00	100.00	64.34	91.54	83.06	82.16	321.8
Synthetic	2	87.63	100.00	100.00	88.65	87.12	84.25	89.32	160.4
	3	97.64	100.00	100.00	98.88	97.02	94.04	99.44	182.0
	4	97.66	100.00	100.00	99.01	96.99	93.99	99.05	204.1
	5	97.65	100.00	100.00	99.01	96.98	93.96	99.50	217.6
	6	97.58	100.00	100.00	98.80	96.68	93.96	99.40	237.4
	7	97.72	100.00	100.00	98.99	97.09	94.18	99.50	288.1
	8	97.72	100.00	100.00	98.96	97.09	94.19	99.48	292.4
	9	97.65	100.00	100.00	98.86	97.05	94.11	99.43	310.5
	10	97.58	100.00	100.00	98.88	96.92	93.85	99.44	320.0

from: (1) low-amplitude signals, (2) sensor artifacts (e.g., goggle slippage), and (3) labeling ambiguity near the 150°/s head velocity threshold.

- **Type I errors** occurred most frequently in overt saccade classification, typically when high-velocity events failed to meet duration thresholds.
- Confusion matrix analysis revealed that most misclassifications involved covert-overt confusion, indicating the need for additional discriminative features beyond onset time and duration.
- We also examined a subset of misclassified covert saccade instances. In many cases, the signal was embedded in noise or exhibited partial suppression, complicating detection. A visual inspection of representative failure cases confirmed that these signals lack sharp transitions, making them difficult to identify even by human annotators. A deeper integration of signal morphology or frequency-based analysis may help mitigate such errors.

Despite the simplicity of the MLP architecture, its training speed and convergence stability were consistent across folds. Compared to more complex architectures like CNNs or LSTMs, the MLP's low computational cost makes it attractive for real-time or embedded applications. Nonetheless, future work should include direct benchmarking against these models to contextualize trade-offs in accuracy versus efficiency.

Performance stability peaked within the $K \in [5,7]$ fold range, representing an optimal trade-off between computational cost and generalization capability.

7 Conclusion

This study introduces an MLP-based deep learning framework for saccade detection and classification in vHIT recordings. The proposed system's key innovation lies in its dual-mode operation, processing both clinical recordings and synthetic signals generated from a kinematic VOR model. This simulation approach effectively mitigates class imbalance while expanding the training distribution.

Through comprehensive K-fold cross-validation ($K \in [2,10]$), the augmented model achieved 96.54% mean accuracy with consistently high sensitivity and specificity across all classes. These findings establish simulated data as a clinically viable alternative to real recordings, especially valuable for resource-limited settings.

While demonstrating strong overall performance, error analysis revealed persistent challenges in covert saccade detection due to their low-amplitude signatures. Future work will explore temporal feature integration and hybrid CNN-LSTM architectures to better model the spatiotemporal dynamics of vestibulo-ocular responses.

References

[1] R. J. Leigh and D. S. Zee, *The Neurology of Eye Movements*. Oxford University Press, 2015.

[2] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. van de Weijer, "Eye tracking: A comprehensive guide to methods and measures," *Oxford University Press*, 2011.

[3] S. Martinez-Conde, S. L. Macknik, and D. H. Hubel, "Toward a saccade-based biomarker for parkinson disease," *Frontiers in Neurology*, vol. 4, p. 178, 2013.

[4] H. G. MacDougall, K. P. Weber, L. A. McGarvie, G. M. Halmagyi, and I. S. Curthoys, "Covert saccades during head impulse testing: a marker of vestibular compensation?" *Neurology*, vol. 72, no. 7, pp. 588–593, 2009.

[5] R. Engbert and K. Mergenthaler, "Microsaccades are triggered by low retinal image slip," *Proceedings of the National Academy of Sciences*, vol. 103, no. 33, pp. 12 349–12 353, 2006.

[6] M. Nyström and K. Holmqvist, "Practical robust fixation detection," *Behavior Research Methods*, vol. 42, no. 1, pp. 372–384, 2010.

[7] M. E. Bellet, J. Bellet, N. Guyader, M. Boucart, J. Grainger, and F. Vitu, "Human-level saccade detection performance using deep neural networks," *Scientific Reports*, vol. 9, no. 1, pp. 1–12, 2019.

[8] M. Startsev, I. Agtzidis, and M. Dorr, "Lstm-based saccade detection," in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*. ACM, 2018, pp. 1–9.

[9] J. Otero-Millan, X. M. Troncoso, S. L. Macknik, I. Serrano-Pedraza, and S. Martinez-Conde, "Oculomotor strategies for fixating and following moving targets," *Vision Research*, vol. 49, no. 5, pp. 705–726, 2014.

[10] R. Zemblys, D. C. Niehorster, and K. Holmqvist, "Gazenet: End-to-end eye-movement event detection with deep neural networks," *Behavior Research Methods*, vol. 51, no. 2, pp. 840–864, 2018.

[11] A. Mihali and R. C. Muresan, "A bayesian generative model for microsaccade detection," *Frontiers in Neuroscience*, vol. 11, pp. 1–15, 2017.

[12] P. M. Daye and L. M. Optican, "A bayesian framework for saccade generation: Eye movements in pursuit tracking and interception," *Journal of Vision*, vol. 14, no. 12, pp. 1–25, 2014.

[13] J. Pekkanen and O. Lappi, "A new threshold-free algorithm for detecting fixations and saccades in eye-tracking data," *Behavior Research Methods*, vol. 49, no. 2, pp. 1234–1250, 2017.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

Ha Ngoc Khoan graduated from Le Quy Don Technical University in 2008 and received a Master's degree from the Le Quy Don Technical University in 2014. Currently, he is a PhD. student at the Academy of Military Science and Technology and works at the Cyclotron Centre, 108 Military Central Hospital.
Main research areas: Biomedical engineering, electronics and communication engineering, radiolabeling, medical and biomedical image processing, nuclear medicine instrumentation, medical image analysis, image reconstruction, image post-processing, interventional radiology.

Tran Van Nghia graduated from Le Quy Don Technical University in 2009 and received a PhD's degree from the Moscow Institute of Physics and Technology in 2018. Currently, he works at the Air Force - Air Defense Technical Institute.
Main research areas: Signal processing, next generation radio telecommunications and television systems, algorithms for crest factor reduction, digital pre-distortion, machine learning, radar signal processing, electronic warfare, and FPGA design.

Le Ky Bien graduated from the Naval Academy in Baku, Azerbaijan, in 1986, and received his Ph.D. from the Baltic State Technical University in Saint Petersburg, Russia, in 2004. He is currently working at the Institute of Military Science and Technology.
His main research interests include system analysis, control and signal processing, circuit design, machine learning, radar signal processing, electronic warfare, and biomedical engineering.