Two-Stream Vision Swin Transformer for Video-based Eye Movement Detection

Ziheng Wang[1,2], **Xiaowei Bai**[1,2,✉], **Xiaodong Wang**[1,2], **Zhenyu Fang**[3,4], **Liang Xie**[1,2], **Ye Yan**[1,2], **Erwei Yin**[1,2]

[1] Defense Innovation Institute, Academy of Military Sciences, Beijing, China
[2] China Intelligent Game and Decision Laboratory, Beijing, China
[3] School of Software, Northwestern Polytechnical University, Xian, China
[4] Tianjin Artificial Intelligence Innovation Center, Tianjin, China
{w1296524574,yy_taiic}@163.com, {xielnudt,yinerwei1985}@gmail.com,
xwbai@ipp.ac.cn, 2201112081@stu.pku.edu.cn, zhenyu.fang@nwpu.edu.cn

Abstract

Eye movement detection plays a crucial role in various fields, including eye tracking applications and understanding human perception and cognitive states. Existing detection methods typically rely on gaze positions predicted by gaze estimation algorithms, which may introduce cumulative errors. While certain video-based methods, directly classifying behaviours from videos, have been introduced to address this issue, they often have limitations as they primarily focus on detecting blinks. In this paper, we propose a video-based two-stream framework designed to detect four eye movement behaviours—fixations, saccades, smooth pursuits, and blinks—from infrared near-eye videos. To explicitly capture motion information, we introduce optical flow as the input for one stream. Additionally, we propose a spatio-temporal feature fusion module to combine information from the two streams. The framework is evaluated on a large-scale eye movement dataset and performs excellent results.

Keywords: eye movement detection; action recognition; neural networks;

Introduction

Eye movement detection is a critical area of research within the field of eye tracking studies, alongside gaze estimation and eye-controlled human-computer interaction. The goal is to accurately predict the start time, end time, and semantic classification of various eye movement behaviours, including fixations, saccades, smooth pursuits, blinks, and additional categories, in a given video or signal sequence obtained from eye-tracking devices. Eye movement detection has widely served for various downstream tasks, such as human-computer interaction (Harezlak, Duliban, & Kasprowski, 2021; Niu et al., 2023) and understanding human perception and cognitive states (Leigh & Zee, 2015; Gale & Findlay, 2021).

Most previous eye movement detection approaches have complicated pipelines, involving pre-processing, classification methods and post-processing, as shown in Figure 1. These approaches, called signal-based, first regress gaze positions from eye images and subsequently classify eye movement behaviours using different classification methods, such as traditional handcrafted feature-based methods (Larsson, Nyström, Andersson, & Stridh, 2015), machine learning methods (Zemblys, Niehorster, Komogortsev, & Holmqvist, 2018), and deep learning methods (Zemblys, Niehorster, & Holmqvist, 2019), followed by a post-processing procedure, which is employed to merge individual samples into coherent events. Signal-based approaches rely on accurate gaze position predictions from upstream gaze estimation algorithms. Decrease in the precision of predicted results leads to cumulative errors that impact the classification performance of eye movement detection. Compared to signal-based approaches, video-based algorithms (Nousias et al., 2022; Zeng et al., 2023) directly classify behaviours in videos without the need for intermediate gaze position acquisition. This reduces dependency on upstream eye tracking algorithms and eliminating cumulative errors. However, these approaches primarily focus on detecting blinks and there is limited work on directly classifying other eye movement behaviours from videos, such as saccades and smooth pursuits, which also play crucial roles in understanding human perception and cognitive states.

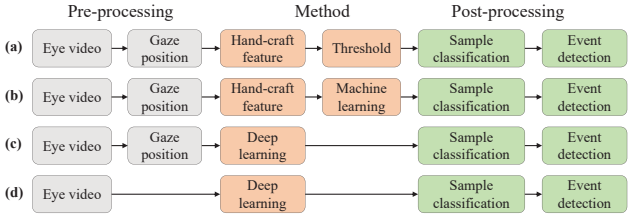

Figure 1: Comparison of different pipelines of eye movement detection. (a) Handcrafted feature-based methods. (b) Machine learning methods. (c) Deep learning methods. (d) Video-based methods in this paper.

In this paper, we propose a video-based eye movement detection framework capable of frame-by-frame classifying four types of eye movement behaviours: fixations, saccades, smooth pursuits, and blinks, from near-eye infrared videos. Considering the characteristics of rapid occurrence and short duration of eye movement behaviours, optical flow is introduced to capturing subtle motion information. Inspired by SlowFast (Feichtenhofer, Fan, Malik, & He, 2019), we design a two-stream architecture to detect eye movement behaviours, where one slow stream utilizes a small number of infrared frames, obtained through downsampling from near-eye infrared videos, as input to focus on spatial features extraction, and another fast stream utilizes a larger number of optical flow frames calculated from videos as input to capture temporal features. To better capture high-dimensional spatio-

In L. K. Samuelson, S. L. Frank, M. Toneva, A. Mackey, & E. Hazeltine (Eds.), *Proceedings of the 46th Annual Conference of the Cognitive Science Society.*

temporal features, we employ the Transformer-based video action recognition model, Video Swin Transformer(VST) (Liu et al., 2022), as the primary feature extraction backbone in each stream. In addition, we propose a spatio-temporal fusion module, employing lateral connections to merge spatial and temporal information and facilitate interaction between the two streams for a better eye movement detection performance.

In summary, the contributions of this paper are as follows:

- We propose a video-based eye movement detection framework capable of classifying four types of eye movement behaviours from near-eye infrared videos. To the best of our knowledge, this is the first method to achieve multi-class classification of eye movement behaviours based on near-eye infrared videos.
- We design a two-stream architecture focusing on extracting spatial and temporal features independently. Additionally, we introduce optical flow as an input modality into one stream to capture the subtle motion information inherent in eye movement behaviours. The Transformer-based action recognition model Video Swin Transformer is used as our feature extraction backbone in each stream.
- We propose a spatio-temporal feature fusion module to merge spatial and temporal information, enabling the framework to have a better performance on the frame-level classification task.

Related Work

Eye Movement Detection Methods

Previous methods for eye movement detection can be roughly categorized into groups based on the pipelines as shown in Figure 1. (a) **Handcrafted feature-based methods** are developed in early eye movement detection research, which relying on complex features(i.e. Velocity (Dorr, Martinetz, Gegenfurtner, & Barth, 2010), PCA-based dispersion thresholding (Berg, Boehnke, Marino, Munoz, & Itti, 2009), etc.) that may lead to a large number of tunable parameters and thresholds, resulting in increased computational costs. (b) **Machine learning methods** allow thresholding and classification to be learned from handcrafted features and performed automatically by using algorithms, such as k-nearest-neighbors (Vidal, Bulling, & Gellersen, 2012), support vector machines (Anantrasirichai, Gilchrist, & Bull, 2016), random forests (Zemblys et al., 2018) and others, but they are still necessary to perform the step of obtaining handcrafted features. (c) **Deep learning methods** automatically learn all features and appropriate thresholds from gaze positions using deep neural networks (Zemblys et al., 2019). However, methods mentioned above are susceptible to upstream gaze estimation tasks and suffer from cumulative errors. (d) **Video-based methods** have been explored to solve this problem by directly classifying eye movement behaviors from videos without any intermediate step. (de la Cruz, Lira, Luaces, & Remeseiro, 2022) combines a convolutional neural network for feature extraction with a bidirectional recurrent neural network that performs sequence learning and classifies the blinks in RGB near-eye videos. (Nousias et al., 2022) performs temporal filtering and adaptive thresholding on the data obtained from iris and eyelid segmentation to classify blinks in near-infrared high-resolution image sequences. Existing video-based methods only detect blinks and have not been applied to the detection of other eye movement behaviours. This paper expands video-based detection methods in the categories of eye movement behaviors.

Action Recognition Methods

Video-based eye movement detection can be regarded as an action recognition task and therefore action recognition models can be used to assist in the classification of eye movement behaviours. Current deep neural networks for action recognition can primarily be categorized into two groups: CNN-based and Transformer-based networks. (Carreira & Zisserman, 2017) inflates the 2D convolutions to 3D convolutions and extracts spatio-temporal features using a two-stream structure. Inspired by the retinal ganglia of primates, (Feichtenhofer et al., 2019) proposes SlowFast, a CNN-based network, using a slow, high-resolution channel to analyse static content in the video, and a fast, low-resolution channel to analyse dynamic content. With the development of vision transformer, Transformer-based networks have achieved excellent performance in various action recognition tasks. (Bertasius, Wang, & Torresani, 2021) studies five different variants of space-time attention and suggests a factorized space-time attention for its strong speed-accuracy tradeoff. (Liu et al., 2022) advocates Video Swin Transformer, which uses an inductive bias of locality in video Transformers instead of self-attention globally and achieves state-of-the-art accuracy on a broad range of video recognition benchmarks. We use Video Swin Transformer as our backbone to extract spatial and temporal features in near-eye infrared videos.

Method

Problem Definition

Given a near-eye infrared video clip $X = \{x_1, x_2, \cdots, x_T\} \in \mathbb{R}^{T \times c \times h \times w}$, we have T infrared frames and corresponding optical flow frames, where c, h and w denote the channel, height and width of each frame, respectively. Ground truth of the video clip is defined as $Y = \{s_k, e_k, c_k\}_{k=1}^{K}$. Here K denotes the number of eye movement behaviour instances in the video clip X. s_k, e_k and c_k denote the start time, end time and semantic classification of each instance. Our goal is to train a model to predict the start time, end time and semantic classification of each instance in the video clip X with high precision.

Method Overview

We propose a video-based two-stream framework for eye movement detection, directly classifying eye movement behaviours from near-eye infrared videos, without the intermediate step of obtaining gaze positions. Inspired by SlowFast

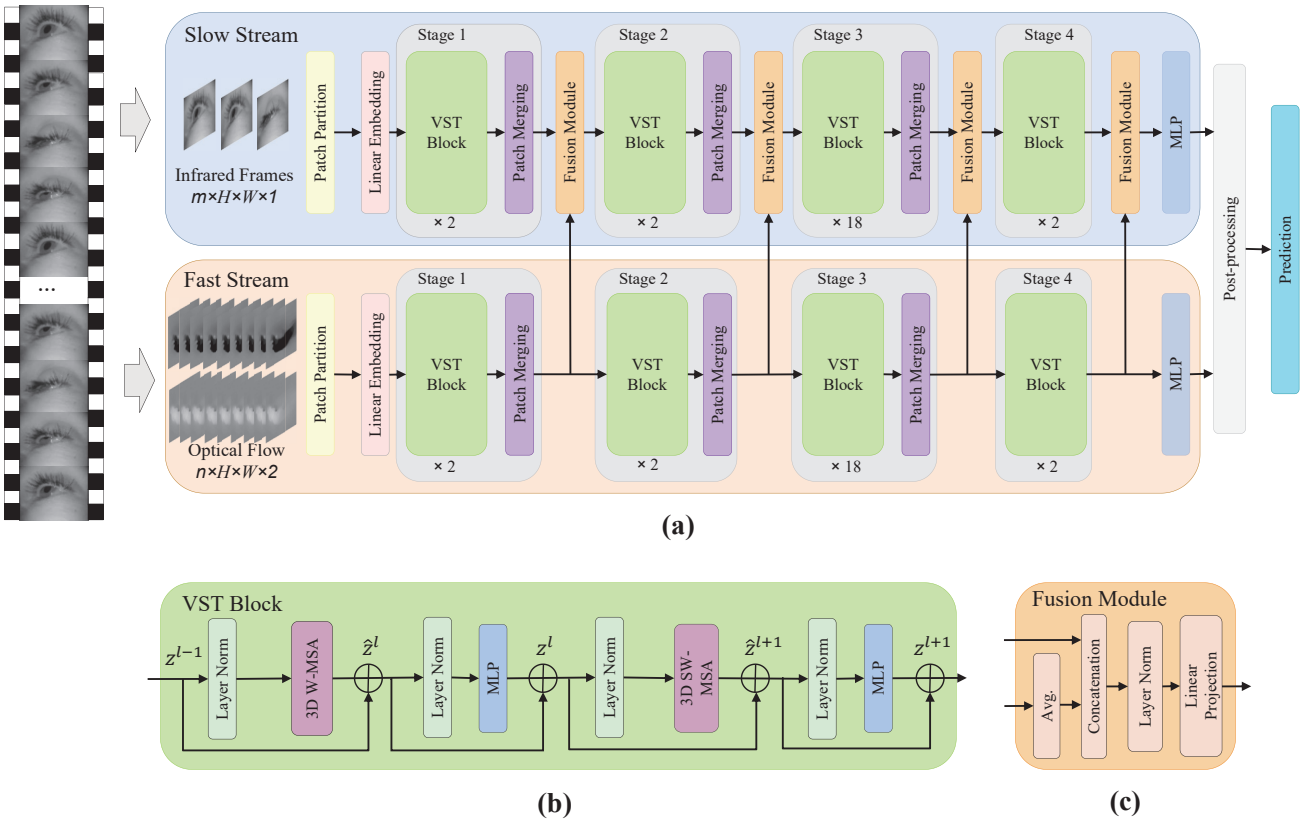

Figure 2: (a) Overview of our proposed framework. (b) Structure of two successive Video Swin Transformer blocks. (c) Structure of our proposed spatio-temporal feature fusion module.

(Feichtenhofer et al., 2019), our two-stream framework consists of a slow stream, emphasizing the extraction of spatial features at a lower frame rate, and a fast stream, focusing on capturing temporal and motion features at a higher frame rate. Considering that optical flow inherently reflects object motion characteristics and overlooks many spatial details, we use optical flow as the input for the fast stream. Concurrently, infrared frames act as the input for the slow stream to extract spatial information with a downsampling operation. As illustrated in the Figure 2(a), the overall process of our framework can be summarized as follows: first, horizontal and vertical components of optical flow are obtained for each frame from the given clip, and sparse infrared frame sequences are obtained through downsampling, serving as inputs for the fast stream and slow stream, respectively. Then, each stream utilizes Vision Swin Transformer (Liu et al., 2022) as the feature extraction backbone, extracting high-dimensional temporal and spatial features at frame level through four stages, followed by a Multilayer Perceptron (MLP) for frame-level classification. The frame-level representations obtained from the fast stream are fused into the slow stream by a spatio-temporal feature fusion module after each stage. Finally, a post-processing procedure is employed to average the outputs of both streams and obtain the final predictions.

Fast Stream

Optical Flow Representation Considering that eye movement behaviours are rapid and short-duration processes, we introduce optical flow to capture the motion characteristics of eye movement behaviours. For an infrared frame, there are horizontal and vertical components of optical flow, we concatenate them in the channel dimension as an optical flow frame. We select n optical flow frames corresponding to the n infrared frames and take them as the input for the fast stream for temporal feature extraction. The optical flow frames can be represented as:

$$\text{Input}_{fast} = \left\{ (\mathrm{I}_i^x \oplus \mathrm{I}_i^y) \right\}_{i=1}^{n} \in \mathbb{R}^{n \times h \times w \times 2} \tag{1}$$

where I_i^x and I_i^y denote horizontal and vertical components of optical flow corresponding to the i-th infrared frame. $\oplus$ is the concatenating operation at channel level.

Patch Partition and Linear Embedding Considering that optical flow has already provided motion information, unlike the approach used in (Liu et al., 2022), we treat each 2D patch, instead of a 3D patch, with dimensions $1 \times 4 \times 4 \times 2$, as an individual token for subsequent frame-level feature learning. In this way, the patch partitioning layer obtains $n \times \frac{H}{4} \times \frac{W}{4} \times 1$ tokens, with each token contains a 32-dimensional fea-

ture. The linear embedding layer maps each token to a C-dimensional feature vector.

Frame-level Feature Learning The feature vectors acquired from linear embedding pass through four stages to learn deeper frame-level features. After each stage, the temporal dimension remains unchanged, the spatial dimension is halved, and the channel doubles. Each stage includes several VST blocks, as shown in Figure 2(b), each utilizing the shifted window mechanism in 3D W-MSA and 3D SW-MSA with a local inductive bias for a better speed-accuracy trade-off. With the shifted window mechanism, two consecutive VST blocks are computed as:

$$\begin{aligned} \hat{z}^l &= \text{3DW}-\text{MSA}(\text{LN}(z^{l-1}))+z^{l-1}, \\ z^l &= \text{MLP}(\text{LN}(\hat{z}^l))+\hat{z}^l, \\ \hat{z}^{l+1} &= \text{3DSW}-\text{MSA}(\text{LN}(z^l))+z^l, \\ z^{l+1} &= \text{MLP}(\text{LN}(\hat{z}^{l+1}))+\hat{z}^{l+1} \end{aligned} \tag{2}$$

where $\hat{z}^l$ and z^l denote the output features of the 3D(S)W-MSA module and the MLP for block l, respectively; 3DW-MSA and 3DSW-MSA denote 3D window based multi-head self-attention using regular and shifted window partitioning configurations, respectively.

Except for stage 4, patch merging is employed for $2\times$ spatial downsampling, followed by concatenation of adjacent 2×2 neighboring patches and a linear layer for channel dimension halving.

MLP Fast stream has acquired temporal feature information from the clip through four stages. Subsequently, a MLP head is used for classifying behaviour types of each frame. To predict frame-level labels for the input clip, the output dimension of the MLP is set to $c\times n$, representing the predicted behaviour probabilities for each frame, where c and n denote the number of behaviour types and input frames for the clip.

Slow Stream

Given that adjacent near-eye infrared frames are nearly identical, we downsample the input video clip to reduce redundancy with a downsampling rate of r, sending the low frame rate infrared frame sequence with length of $m=\lceil \frac{n}{r} \rceil$ to the slow stream for learning frame-level spatial features such as appearance and shape. Unlike optical flow frames, near-eye infrared frames are input as a single-channel, where the patch size in the channel dimension changes from 2 to 1. The patch partitioning layer obtains $m\times\frac{H}{4}\times\frac{W}{4}\times 1$ tokens, and each token contains a 16-dimensional feature. The mapping dimension of the linear embedding layer remains consistent with the fast stream.

Inspired by SlowFast (Feichtenhofer et al., 2019), we employ lateral connections for two-stream feature fusion. A Fusion Module is added after each stage during frame-level feature learning in the slow stream. The features learned from the fast stream at each stage are incorporated into the slow stream through lateral connections followed by the feature fusion module to fuse temporal and spatial information.

Despite downsampling resulting in the loss of spatial information, the output size of the MLP is also set to $c\times n$ for n-frame prediction in the slow stream. Table 1 shows variations in data size after each step in each stream.

Table 1: Variations in data size after each step in each stream.

	Slow Stream	Fast Stream
Input size	$m\times H\times W\times 1$	$n\times H\times W\times 2$
Partition	$m\times\frac{H}{4}\times\frac{W}{4}\times 16$	$n\times\frac{H}{4}\times\frac{W}{4}\times 32$
Embedding	$m\times\frac{H}{4}\times\frac{W}{4}\times C$	$n\times\frac{H}{4}\times\frac{W}{4}\times C$
Stage 1	$m\times\frac{H}{8}\times\frac{W}{8}\times 2C$	$n\times\frac{H}{8}\times\frac{W}{8}\times 2C$
Stage 2	$m\times\frac{H}{16}\times\frac{W}{16}\times 4C$	$n\times\frac{H}{16}\times\frac{W}{16}\times 4C$
Stage 3	$m\times\frac{H}{32}\times\frac{W}{32}\times 8C$	$n\times\frac{H}{32}\times\frac{W}{32}\times 8C$
Stage 4	$m\times\frac{H}{32}\times\frac{W}{32}\times 8C$	$n\times\frac{H}{32}\times\frac{W}{32}\times 8C$
MLP	$c\times n$	$c\times n$

Spatio-temporal Feature Fusion Module During frame-level feature learning of the fast stream, frame-level classification tokens are obtained through each stage. As shown in Figure 3, each frame is represented as a frame-level classification token $f_i, i=1,\ldots,n$, where n denotes the number of optical flow frames in the fast stream. Similarly, the frame-level classification token in the slow stream can be described as $g_j, j=1,\ldots,m$, where m represents the number of infrared frames in the slow stream. Assuming $r=\lceil \frac{n}{m} \rceil$, we average r tokens to one token in the fast stream, obtaining the classification tokens $f'_i, i=1,\ldots,m$. Subsequently, based on the frame index, averaged tokens from the fast stream are concatenated to the slow stream through lateral connections and then sent to the next stage before the operation of layer normalization and linear projection to half the dimension of channel. The detailed fusion structure can be observed in the Figure 2(c).

Loss Function

Each stream utilizes the cross-entropy loss $\mathcal{L}_{cls}$ to calculate the loss between the output and the ground truth.

$$\begin{aligned} \mathcal{L}_{cls} = -\frac{1}{n}\sum_{i=1}^{n} Y(i)\odot log(\hat{Y}(i) \\ +(1-Y(i))\odot log(1-\hat{Y}(i)) \end{aligned} \tag{3}$$

where $Y(i)$ and $\hat{Y}(i)$ represent the label vector and predicted vector of i-th frame, and $\odot$ denotes the Hadamard product. The losses are then weighted, yielding final loss function. The final loss function $\mathcal{L}$ is defined as:

$$\mathcal{L}=\lambda\mathcal{L}_{fast}+(1-\lambda)\mathcal{L}_{slow} \tag{4}$$

Table 2: Sample-level F1 scores and Event-level F1 scores on Gaze-in-the-Wild dataset compared to CNN-based and Transformer-based action recognition baselines. "SP" stands for "Smooth Pursuit".

Method	Sample-F1					Event-F1				
	Fixation	SP	Saccade	Blink	Average	Fixation	SP	Saccade	Blink	Average
Eye-LRCN(de la Cruz et al., 2022)	65.79	50.32	45.72	67.47	57.33	50.53	36.12	34.69	44.07	41.35
C3D(Tran, Bourdev, Fergus, Torresani, & Paluri, 2015)	67.47	66.70	65.86	83.28	70.83	65.26	43.02	57.71	82.77	62.19
I3D(Carreira & Zisserman, 2017)	71.95	68.29	67.85	84.10	73.05	67.43	46.57	62.02	85.24	65.32
Slow-Fast(Feichtenhofer et al., 2019)	76.58	69.44	63.12	87.94	74.27	69.80	55.34	61.29	86.56	68.25
ViViT-B(Arnab et al., 2021)	73.39	63.97	59.67	88.71	71.44	68.91	53.31	57.33	86.36	66.48
MViTv2-S(Li et al., 2022)	77.96	69.70	68.56	89.34	76.39	70.87	54.04	69.92	87.17	70.50
TimeSformer(T+S)(Bertasius et al., 2021)	78.32	70.96	72.47	90.56	78.08	71.13	55.37	69.36	88.09	70.99
ours(Slow stream)	68.49	55.86	54.46	87.44	66.56	68.96	50.23	53.92	85.91	64.76
ours(Fast stream)	72.61	64.97	71.28	89.66	74.63	71.63	53.14	68.11	87.46	70.09
ours(Two-Stream)	**79.92**	**71.85**	**73.60**	**91.47**	**79.21**	**72.39**	**60.32**	**72.52**	**88.38**	**73.40**

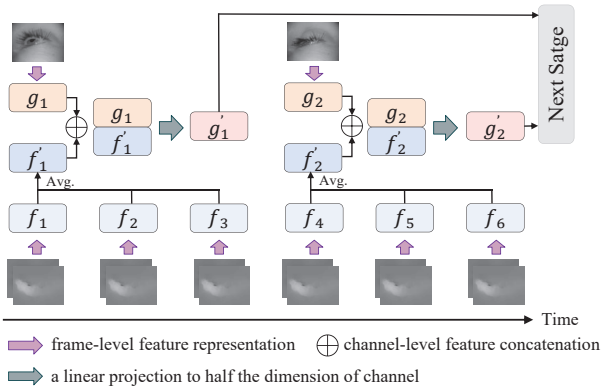

Figure 3: Illustration of our proposed feature fusion module. (an example of n=6, m=2, r=3)

where $\mathcal{L}_{fast}$ is the cross-entropy loss $\mathcal{L}_{cls}$ of the fast stream, $\mathcal{L}_{slow}$ is the cross-entropy loss $\mathcal{L}_{cls}$ of the slow stream and λ is balanced weight.

Post-processing Module

The goal of eye movement detection is to detect the start, end times and semantic classification of each eye movement behaviour instance. To achieve this, a post-processing step is necessary. The predicted results from the fast stream and slow stream are averaged. For each frame, the class with the highest predicted probability is then selected as the predicted label for eye movement behaviour. Subsequently, an operation Φ that adjacent frames with the same predicted label are merged into one event is used to determine the semantic classification and start and end time of each instance. The final prediction $\hat{Y}$ is defined as:

$$\hat{Y}(X) = \Phi(\{max(\hat{Y}(i))\}_{i=1}^{n}) = \{\hat{s}_i, \hat{e}_i, \hat{c}_i\}_{i=1}^{K'} \quad (5)$$

where $\hat{s}_i$, $\hat{e}_i$ and $\hat{c}_i$ denote the start time, end time and classification of i-th predicted eye movement event. K' is the total number of predicted events for the input X.

Experiments

Dataset and Evaluation Metrics

We evaluated our proposed framework on Gaze-in-the-Wild dataset (Kothari et al., 2020), a large-scale public infrared near-eye dataset, which contains over 140 minutes of hand-labelled eye movement events collected from 19 participants in four different environments and approximately 20,000 fixations, 18,000 saccades, 1,200 smooth pursuits, and 4,000 blinks. Our framework is evaluated using leave-one-out cross validation by testing on a single subject's data, training on remaining subjects, and reporting average performance. In order to ensure the balanced distribution of data, undersampling is adopted for fixations and saccades.

We used sample-level metrics sample-F1 score (Sokolova & Lapalme, 2009) and event-level metrics event-F1 score (Hooge, Niehorster, Nyström, Andersson, & Hessels, 2018) to assess the performance of our method.

Baselines

In addition to the blink detection model Eye-LRCN (de la Cruz et al., 2022), we choose the mainstream CNN-based backbones including C3D (Tran et al., 2015), I3D (Carreira & Zisserman, 2017), Slow-Fast (Feichtenhofer et al., 2019) and Transformer-based backbones including MViTv2-S (Li et al., 2022), TimeSformer(T+S) (Bertasius et al., 2021), ViViT-B (Arnab et al., 2021)as the feature extractor, followed by a MLP for classification, to compare with our method.

Implementation Details

We computed optical flow by using implementation of (Brox, Bruhn, Papenberg, & Weickert, 2004) from the OpenCV toolbox before training. Swin-S ($C = 96$, layer numbers$=\{2,2,18,2\}$) is used as our feature extraction backbone. During the training phase, the input of each stream is resized from the original resolution of 640x480 to 256x256 and subjected to a random crop operation, resulting in a size of 224x224. Data augment techniques including rotation and flipping are also used to improve the robustness of the model and reduce the sensitivity to frames. All compared models are trained for 30 epochs, utilizing a batch size of 16 and a cosine learning

rate decay scheduler on the NVIDIA A100 GPU. For optimization, we employ AdamW (Kingma & Ba, 2014) to minimize the weighted cross-entropy loss function, starting with an initial learning rate of 5e-5 and a weight decay of 0.02. The balanced weight λ is set to 0.8. We set the input size of fast stream and slow stream to $n = 16$ and $m = 8$ ($r = 2$) for the framework, respectively. During testing, we use a no-lapping sliding window method to detect the whole video. The sliding window size is set to 16.

Results

Main Experimental Results

Table 2 shows the experimental results of our method and baselines on Gaze-in-the-Wild dataset. As expected, our proposed method demonstrates superior performance, outperforming TimeSformer (Bertasius et al., 2021), the most effective network among the baselines, with an average improvement of 1.13% and 2.41% for sample-level and event-level evaluation metrics, respectively. Notably, our method excels in blink detection, achieving sample-F1 and event-F1 of 91.47% and 88.38%, respectively, followed by fixations, saccades, and smooth pursuits. Although event-F1 for smooth pursuits is 60.32%, it signifies a significant improvement of 4.95% compared to TimeSformer (Bertasius et al., 2021).

Ablation Studies

Analysis of Two-stream Architecture To evaluate the effectiveness of the two-stream architecture, we conducted separate assessments of the single-stream model on Gaze-in-the-Wild dataset. The results in Table 2 illustrate that the two-stream architecture, coupled with a feature fusion module, surpasses the performance of using a single-stream strategy alone, showcasing improvements across all evaluation metrics. Additionally, we observe differential enhancement levels in recognizing all four eye movement behaviours when utilizing optical flow as the input for the single-stream structure compared to using the raw infrared video. Particularly, there is a notable enhancement for saccades, with improvements of 16.82% and 14.19% in sample-level and event-level metrics, respectively. This enhancement can be attributed to the rapider nature of saccades compared to other eye movement behaviours. The utilization of optical flow effectively captures and represents the distinctive characteristics of these swift movements.

Analysis of Downsampling Rate To study the effects of downsampling rate on performance, we set the input size of the fast stream to n=16 and adjusted the downsampling rate r to 1, 2, 4, and 8, respectively, on Gaze-in-the-Wild dataset. The larger the downsampling rate, the fewer input frames for the slow stream. As shown in Figure 4, with the increase of the downsampling rate, our framework exhibits a trend of initially improving and subsequently declining classification performance for fixations, saccades, and smooth pursuits, achieving the best performance when the downsampling rate is 2. We conjecture this is due to the fact that the redundancy of data in neighbouring frames makes the framework underperform when downsampling is not used ($r = 1$), and then with the decrease of the number of input frames for the slow stream ($r > 2$), resulting in the reduction of spatial feature information, the performance decreases. Interestingly, we observe that when the downsampling rate is 2, there is a significant decrease in the classification performance for blinks. We attribute this to the distinct generation mode of blinks compared to the other three eye movement behaviours, as some details about eyelid movements are forgotten when the framework focuses more on learning eyeball movement information.

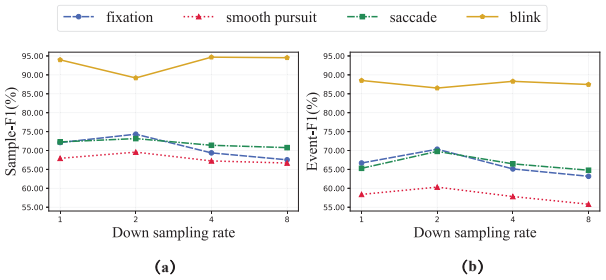

Figure 4: Results for Sample-F1 (a) and Event-F1 (b) on Gaze-in-the-Wild dataset with varying downsampling rate parameters.

CONCLUSION

In this paper, we present a novel two-stream vision swin transformer framework for eye movement detection, designed to classify four eye movement behaviours in infrared near-eye videos. To the best of our knowledge, this is the first video-based multi-class eye movement detection method. One stream of our framework introduces optical flow as input to capture temporal and motion information of eye movement behaviours, while the other stream focuses on extracting spatial features such as appearance and shape. To fuse spatial and temporal information, we propose a feature fusion module with lateral connections to combine the frame-level feature representations learned from two streams. Our framework is evaluated on Gaze-in-the-Wild dataset, demonstrating superior performance compared to other baselines in both sample-level and event-level metrics. In the future, we aim to explore alternative approaches dedicated to enhancing the recognition performance of smooth pursuits.

Acknowledgments

This work was supported in part by the grants from the National Natural Science Foundation of China under Grant 62332019 and 62076250, the National Key Research and Development Program of China (2023YFF1203900, 2023YFF1203903).

References

Anantrasirichai, N., Gilchrist, I. D., & Bull, D. R. (2016). Fixation identification for low-sample-rate mobile eye trackers. In *2016 ieee international conference on image processing (icip).*

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., & Schmid, C. (2021). Vivit: A video vision transformer. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 6836–6846).

Berg, D. J., Boehnke, S. E., Marino, R. A., Munoz, D. P., & Itti, L. (2009). Free viewing of dynamic stimuli by humans and monkeys. *J Vis*, *9*(5), 19.1.

Bertasius, G., Wang, H., & Torresani, L. (2021). Is space-time attention all you need for video understanding? In *Icml* (Vol. 2, p. 4).

Brox, T., Bruhn, A., Papenberg, N., & Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *Computer vision-eccv 2004: 8th european conference on computer vision, prague, czech republic, may 11-14, 2004. proceedings, part iv 8* (pp. 25–36).

Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *Computer vision and pattern recognition.*

de la Cruz, G., Lira, M., Luaces, O., & Remeseiro, B. (2022). Eye-lrcn: A long-term recurrent convolutional network for eye blink completeness detection. *IEEE Transactions on Neural Networks and Learning Systems*.

Dorr, M., Martinetz, T., Gegenfurtner, K. R., & Barth, E. (2010, 08). Variability of eye movements when viewing dynamic natural scenes. *Journal of Vision*, *10*(10), 28-28. Retrieved from https://doi.org/10.1167/10.10.28 doi: 10.1167/10.10.28

Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2019). Slowfast networks for video recognition. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 6202–6211).

Gale, A. G., & Findlay, J. M. (2021). Eye movement patterns in viewing ambiguous figures. In *Eye movements and psychological functions* (pp. 145–168). Routledge.

Harezlak, K., Duliban, A., & Kasprowski, P. (2021). Eye movement-based methods for human-system interaction. a comparison of different approaches. *Procedia Computer Science*, *192*, 3099–3108.

Hooge, I. T., Niehorster, D. C., Nyström, M., Andersson, R., & Hessels, R. S. (2018). Is human classification by experienced untrained observers a gold standard in fixation detection? *Behavior Research Methods*, *50*, 1864–1881.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kothari, R., Yang, Z., Kanan, C., Bailey, R., Pelz, J. B., & Diaz, G. J. (2020). Gaze-in-wild: A dataset for studying eye and head coordination in everyday activities. *Scientific reports*, *10*(1), 2539.

Larsson, L., Nyström, M., Andersson, R., & Stridh, M. (2015). Detection of fixations and smooth pursuit movements in high-speed eye-tracking data. *Biomedical Signal Processing and Control*, *18*, 145–152.

Leigh, R. J., & Zee, D. S. (2015). *The neurology of eye movements*. Contemporary Neurology.

Li, Y., Wu, C.-Y., Fan, H., Mangalam, K., Xiong, B., Malik, J., & Feichtenhofer, C. (2022). Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 4804–4814).

Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., & Hu, H. (2022). Video swin transformer. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 3202–3211).

Niu, Y., Li, X., Yang, W., Xue, C., Peng, N., & Jin, T. (2023). Smooth pursuit study on an eye-control system for continuous variable adjustment tasks. *International Journal of Human–Computer Interaction*, *39*(1), 23–33.

Nousias, G., Panagiotopoulou, E.-K., Delibasis, K., Chaliasou, A.-M., Tzounakou, A.-M., & Labiris, G. (2022). Video-based eye blink identification and classification. *IEEE Journal of Biomedical and Health Informatics*, *26*(7), 3284–3293.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information processing & management*, *45*(4), 427–437.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the ieee international conference on computer vision* (pp. 4489–4497).

Vidal, M., Bulling, A., & Gellersen, H. (2012). Detection of smooth pursuits using eye movement shape features. In *Proceedings of the symposium on eye tracking research and applications* (pp. 177–180).

Zemblys, R., Niehorster, D. C., & Holmqvist, K. (2019). gazenet: End-to-end eye-movement event detection with deep neural networks. *Behavior research methods*, *51*, 840–864.

Zemblys, R., Niehorster, D. C., Komogortsev, O., & Holmqvist, K. (2018). Using machine learning to detect events in eye-tracking data. *Behavior research methods*, *50*, 160–181.

Zeng, W., Xiao, Y., Wei, S., Gan, J., Zhang, X., Cao, Z., ... Zhou, J. T. (2023). Real-time multi-person eyeblink detection in the wild for untrimmed video. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 13854–13863).