SynBlink and BlinkFormer: A Synthetic Dataset and Transformer-Based Method for Video Blink Detection

Bo Liu[1]
bliu03@buaa.edu.cn
Yang Xu[1]
yangxu_22@buaa.edu.cn
Feng Lu[1,2] ✉
lufeng@buaa.edu.cn

[1] State Key Laboratory of VR
Technology and Systems,
Beihang University,
Beijing, China

[2] Peng Cheng Laboratory,
Shenzhen, China

Abstract

Accurate blink detection algorithms have significant implications in numerous fields, including human-computer interaction, driving safety, cognitive science, and medical diagnostics. Despite considerable efforts, the dataset volume for blink detection remains relatively small due to the cost of data collection and annotation, and there is still room for improvement in the accuracy of current algorithms. In this paper, we introduce a workflow for synthesizing video data in Blender. Fully-rigged 3D human models are programmatically controlled, with variations in head movement, blinking, camera angles, background types, and lighting intensities. We used this workflow to create the SynBlink dataset, which includes 50,000 video clips and their corresponding annotations. Additionally, we present BlinkFormer, an innovative blink detection algorithm based on Transformer architecture that fully exploits temporal information from video clips. The model not only detects blinks for the entire input video but also estimates blink strength for each frame individually. Experimental results reveal that the BlinkFormer outperforms other state-of-the-art blink detection methods, achieving the highest F1-score on HUST-LEBW dataset. This accomplishment highlights the effectiveness of our approach in accurately detecting blinks and its potential for real-world applications. Our code and data are publicly available at https://github.com/desti-nation/BlinkFormer.

1 Introduction

Blinking is a physical act that involves briefly shutting and reopening one's eyes. It can serve as an indicator of both an individual's physical condition and their mental state. Significantly, there is a noticeable alteration in blink patterns when individuals experience fatigue, anxiety, or nervousness [29, 34]. Therefore, blink detection has significant value in many fields, including driving safety [11, 28], human-computer interaction [24], cognitive science [10], and medical diagnosis [31].

✉ Corresponding Author

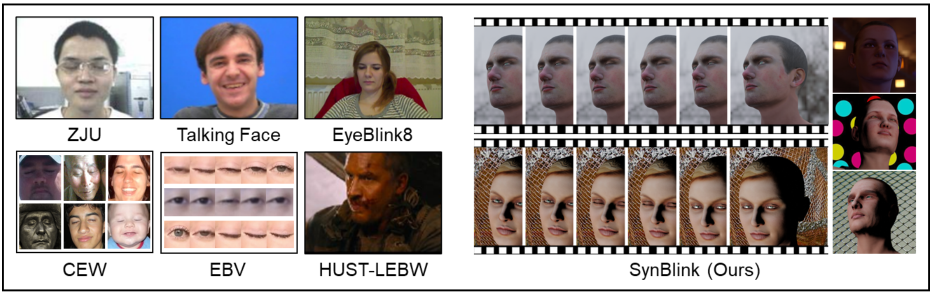

Figure 1: Blink Detection Datasets Overview

In a broader sense, blink detection methods can be categorized into two groups: one based on still images and the other based on video sequences. Still image-based blink detection is also known as eye state detection (closed or open). The methods can be classified into three main categories, including shape based [25], template based [16, 37], and learning based [17, 33]. However, image-based methods cannot capture the dynamic process of blinking, making it difficult to apply in real-world settings. Our paper primarily focuses on blink detection in videos.

Recently, there has been a growing interest in blink detection on video sequences or clips. As shown in Figure 1, several datasets have been proposed, including EyeBlink8 [13], Talking Face [7], Researcher's Night [15], EBV [26], and HUST-LEBW [20]. As a result of the expensive collection and annotation process, the number of videos in these datasets is relatively small, typically consisting of several hundred clips. These datasets are predominantly composed of frontal images captured under favorable lighting conditions, making them prone to overfitting. To date, several methods have been proposed for blink detection, including geometric-based methods such as EAR [4], motion-based methods such as Motion Vector [13], deep learning-based methods using CNN [3, 8] and LSTM [20, 26]. However, these methods fail to fully leverage the temporal information present in blink sequences, indicating potential for further improvement. Therefore, large-scale video blink datasets and algorithms that better exploit temporal information need to be developed urgently.

To tackle this challenge, we develop a workflow for generating synthetic blink videos using 3D human models. This approach allows us to create large amounts of complex and precise-labeled data more efficiently than manual collection and annotation. Moreover, we introduce SynBlink, a challenging dataset consisting of 50,000 video clips, which is currently the largest known video blink dataset. The dataset includes labels indicating whether a sequence contains blinks, the intensity of each blink frame-by-frame, and keypoints. As depicted in Figure 1, SynBlink is diverse and covers individuals of different ethnicities, angles, eye sizes, and lighting conditions.

In addition, we present BlinkFormer, a novel blink detection algorithm that utilizes the Transformer architecture to overcome the challenges associated with temporal feature extraction and parallelism in RNN-based methods. Moreover, we extend BlinkFormer with an additional head that produces the strength of each blink in every frame. This enables more precise learning of the critical features of blinking, particularly given the characteristics of the SynBlink dataset.

The main contributions of our paper are threefold: **1) The Synthetic Data Workflow:** A controllable and flexible workflow for synthesizing human video clips with blinks, allowing for the unlimited generation of data with precise labeling. **2) SynBlink:** The largest video

Dataset	Volume	People	Annotation	Source	Background	Age*	Ethnicity	Region	Angle	Light
ZJU [30]	80 Videos (10876 Frames)	20	Open/Close (Image)	Lab	Indoor	Y M	Asian	Face	Front	Good Stable
CEW [33]	2423 Images	-	Open/Close (Image)	Internet & LFW [21]	Indoor Outdoor	Y M E	-	Face	Varied	Good
RT-BENE [8]	243714 Images	-	Open/Close (Image)	RT-GENE [14]	Indoor	-	-	Eye	Front	Good
Eyeblink8 [13]	8 Videos (71748 Frames)	4	Open/Close (Frame)	Home	Indoor	Y M	European	Face	Front	Good
Talking Face [7]	4 Videos (5000 Frames)	1	Blink Strength (Frame)	-	Indoor	M	European	Face	Front	Good Stable
Researcher's Night [15]	107 Videos (223000 Frames)	-	Blink Interval Face Bouding Box Eye Landmarks (Frame)	Event	Indoor	Y M	European	Face	Front	Varied
HUST-LEBW [20]	673 Video Clips (8749 Frames)	172	Blink State (Video Clip)	Film	Indoor Outdoor	Y M E	Varied	Face	Varied	Varied
EBV [26]	11373 Video Clips (142280 Frames)	50	Open/Close (Frame)	Internet	Indoor	-	-	Eye	Front	Good
SynBlink	50000 Video Clips (650000 Frames)	100	Blink State(Video Clip) Eye Landmarks & Blink Strength(Frame)	Synthesis	Indoor Outdoor Textures HDRI	Y M E	Varied	Face	Varied	Varied

*Note: In the Age column, Y stands for Young, M stands for Middle-aged, and E stands for Elderly.

Table 1: Comparison of SynBlink Dataset with Existing Blink Datasets

blink dataset available as of now. **3) BlinkFormer:** The first Transformer-based method that performs both blink detection for video clips and estimates the blink strength in frame level.

2 Related Work

Blink Detection Datasets. Table 1 provides a summary of the most widely used eye blink detection databases. Despite the existence of some blinking datasets, they still remain inadequate. Static image datasets such as CEW [33] and RT-BENE [8], as well as those containing temporal information such as ZJU [30], EyeBlink8 [13], Talking Face [7], Researcher's Night [15], and EBV [26], have mostly been captured indoors with front view and lighting. As of now, the performance of existing methods has been saturated on these datasets. For instance, the two-stream CNN-based approach [32] achieved an F1-score exceeding 98% on ZJU, Talking Face, and Eyeblink8 datasets. HSUT-LEBW [20] is a challenging dataset consisting of video clips extracted from movies. However, due to the time-consuming and labor-intensive annotation process, the dataset only contains 673 video clips, which makes it relatively small. To accelerate the progress and implementation of blinking detection algorithms, a larger dataset is urgently needed. As shown in Table 1, the synthetic dataset we propose, SynBlink, is the largest dataset containing the most number of video clips and is more diverse in terms of scene, age, ethnicity, angle, and lighting. Therefore, our dataset brings challenges and new opportunities to this field.

Blink Detection Algorithms. Still image-based methods [8, 22, 32] are commonly used to determine whether an eye is open or closed. However, they cannot capture the dynamic process of blinking, which results in a lack of temporal features and poor generalization. Therefore, our paper primarily focuses on blink detection algorithms based on video data.

Considering temporal information, various methods have been proposed. Hu *et al.* [20] utilized uniform LBP as an appearance feature and the LBP difference between consecutive frames as a motion feature for blink characterization. Torricelli *et al.* [35] proposed a detection method using frame differencing and anthropometric properties of the eyes. Dru-

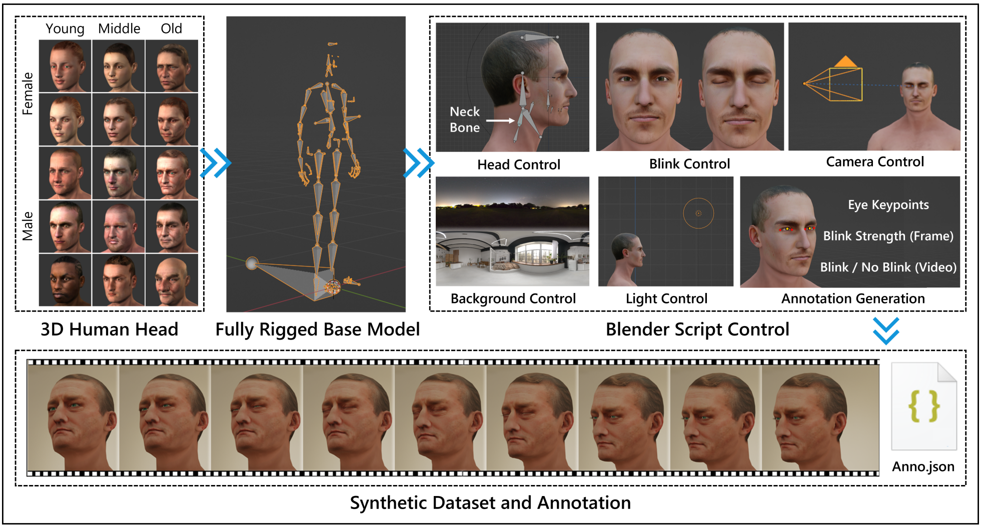

Figure 2: Generation Workflow of SynBlink Dataset

tarovsky *et al.* [13] used a KLT tracker in the eye region to extract blink motion information. Li *et al.* [26] utilized a CNN to transform input eye regions into discriminative features, which were then fed into an LSTM to estimate eye states for each frame. Additionally, Hu *et al.* [20] proposed a 2D Pyramidal Bottleneck Block Network (PBBN) for verifying potential eye blinks in motion, as well as a 3D PBBN model consisting of multiple 3D blocks and global average pooling layers. Recently, self-attention based Transformer has delivered impressive results in image and video analysis [1, 12]. Building upon this inspiration, we introduce the pure Transformer-based approach for blink detection.

3 SynBlink: A Synthetic Dataset for Blink Detection

Based on the aforementioned introduction, the existing blink datasets are relatively small in scale, particularly in terms of video datasets. This has introduced limitations on the applicability of novel deep learning techniques, such as Transformer-based methods [1, 12, 36]. In addition, annotating real-world blink data is both labor-intensive and prone to errors and uncertainties. Therefore, we propose a workflow for generating large-scale data by controlling the blinking of 3D virtual humans in Blender. The specific generation workflow is described below.

3.1 Synthetic Data Generation Workflow

Figure 2 illustrates the entire generation workflow of the SynBlink dataset, which consists of the folowing key parts:

1) 3D Model Preparation. Following [2], we utilize the Realistic Human 100 Pack from Reallusion Inc., which includes 100 3D heads. This pack has diversity among different ethnicities, age ranges, and variations, obtained from real face scans. To control the blink and head movement, we use two base body models which are fully rigged in Character Creator (one male and one female). As depicted in Figure 2, the skeletal system in base models

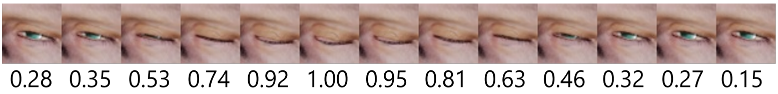

Figure 3: Blink Strength Example

includes more than 70 bones that cover the entire body, including fingers and the face. We replace each 3D head with the base model's original head in Character Creator separately. A total of 100 fully-rigged models are exported and saved as Fbx files.

2) Blender Script Control. Subsequently, we import the Fbx files into Blender, which is a free and open-source 3D computer graphics software that supports modeling, rendering, animation, and script control. We use randomized scripts to control various scenarios below, simulating complex conditions in the real world.

Head Control: By rotating the neck bone in the base model, we can turn the entire head structure connected to this bone simultaneously. Setting the initial and final frames' neck bone angles can automatically generate the intermediate head postures through interpolation in Blender.

Blink Control: To simulate blinking, we use shape keys in Blender to alter the blink strength (the degree of openness) of the eyes. First, we set the blink strength at the starting and ending frames. Then, we select any frame in the video clip and set its blink strength to 1, representing fully closed eyes. The remaining frames are generated automatically through interpolation in Blender. The blink strength of each frame is also recorded in the annotation. Figure 3 is an example.

Camera Control: To center the camera on the head, constraints are added by including a spherical empty object with the head at its center. This ensures that the camera can always capture the person's head. The position of the camera is randomized within a certain range to provide a greater variety of shooting angles.

Background Control: We increase data diversity and simulate various scenes by adding three types of backgrounds: 118,000 images from the training set of the COCO dataset [27], Describable Textures Dataset (DTD) [6] containing 5,640 texture images in the wild, and High Dynamic Range Imaging (HDRI) images downloaded from the Poly Haven website [19], including indoor/outdoor scenes, different lighting, times of day, seasons, etc.

Light Control: Point light sources are used for lighting, with their positions randomly distributed within a certain range. The light intensity is also randomly distributed in three levels: low, medium, and high.

3) Annotation Generation. We annotate the key points of the eyes (left and right eye corner points and pupil center points). To convert a 3D vertex coordinate (X,Y,Z) to its corresponding 2D image point (u,v) in Blender, we use the camera projection matrix P. The conversion formula is $[u,v,w]^T = P[X,Y,Z,1]^T$. Here, w is the scaling factor, which is usually equal to 1. Therefore, the final 2D point coordinates are $(u/w, v/w)$.

Finally, rendering is done using the Eevee engine [18] in Blender.

3.2 Dataset Details and Strengths

The SynBlink dataset consists of 50,000 video clips, each comprising 13 frames. The size of each frame is 350 × 350 pixels. The size of the dataset is 126GB. The dataset includes 25085 blinking video clips and 24915 non-blinking ones. As shown in Table 1, SynBlink offers the following advantages:

1) Large Data Volume: SynBlink is currently the largest video blink dataset available, providing feasibility and convenience for deep learning methods that require a massive amount of data. **2) Complex Scenarios**: Different head postures, head movements, eye sizes, age, ethnicity, gender, light intensity, camera angles, background types, etc., have all been taken into account. **3) Accurate Annotations**: Synthetic workflow provides the advantage of obtaining accurate data annotations, including whether the entire clip blinks or not, keypoints of the eyes in each frame, pupil center positions, and blinking strengths.

4 BlinkFormer: A Video Blink Detection Transformer

4.1 Network Architecture Overview

Inspired by the Vision Transformer [12], we propose BlinkFormer for video blink detection. Given a video clip, eye patches are extracted and embedded into linear features. After position encoding, transformer architecture will learn the temporal information through multi-head attention. A blink detection head is used to classify the whole sequence into blink detection result, while another head output the blink strength of each frame.

4.2 The BlinkFormer Model

1) Eye Embedding. Our approach is designed for video blink detection and deals with cropped eye clip $\mathbf{X} \in \mathbb{R}^{T \times C \times W \times H}$, where T represents the input clip length, C represents the number of channels, and (W,H) is the resolution of the frame. Taking inspiration from the ViT architecture, we consider each frame within a clip as a patch. These patches are flattened into a fixed-size embedding token denoted as $\mathbf{X}_{Emb} \in \mathbb{R}^{T \times D}$, where D represents the dimensionality of the patch embedding. The transformation can be formulated as follows:

$$\mathbf{X}_{Emb} = Linear(Re(X)). \tag{1}$$

Here, Re is a function that converts the shape (T,C,H,W) into $(T,C \times H \times W)$ by rearranging the pixels in each frame into a single vector. $Linear$ is a linear transformation that maps the input to the shape (T,D).

2) Position Embedding. To account for the temporal information between frames, we add learnable position embeddings to the input tensor. The position embeddings $\text{Pos}_{Emb} \in \mathbb{R}^{(T+1) \times D}$ are generated using a random normal distribution. We also add a learnable classification token $\text{Cls}_{Token} \in \mathbb{R}^{1 \times D}$:

$$\mathbf{X}'_{Emb} = (\text{Cls}_{Token} \oplus \mathbf{X}_{Emb}) + \text{Pos}_{Emb}. \tag{2}$$

Here, $\oplus$ denotes concatenation along the first dimension.

3) Transformer Encoder. We then apply a Transformer [36] model to $\mathbf{X}'_{Emb}$. The Transformer consists of multiple layers of multiheaded self-attention and linear blocks. Each layer models all pairwise interactions between all temporal tokens, and it thus models long-range interactions across the video from the first layer, resulting in a transformed sequence $\mathbf{Z} \in \mathbb{R}^{(T+1) \times D}$:

$$\mathbf{Z} = Transformer(\mathbf{X}'_{Emb}), \tag{3}$$

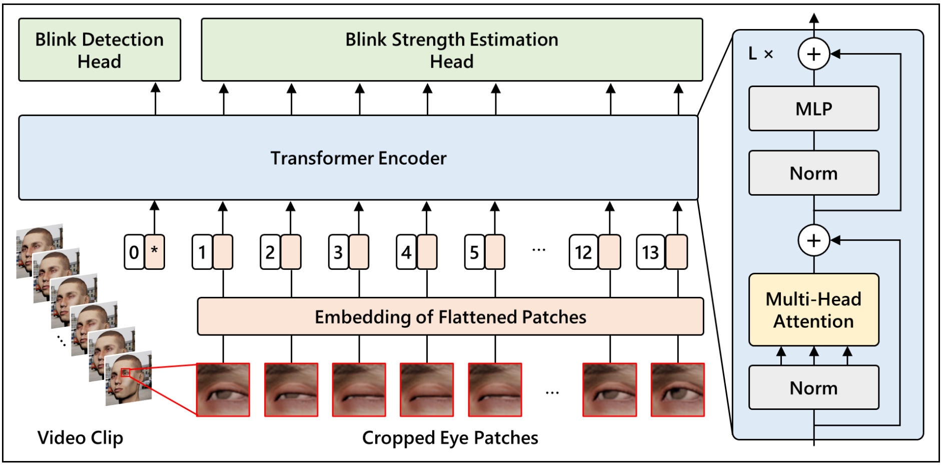

Figure 4: Architecture of BlinkFormer

where $Transformer$ denotes the transformer network with L layers.

4) Blink Detection and Strength Estimation Heads. Finally, we apply a blink detection head to obtain the binary classification output (Out_{cls}), and another head for blink strength estimation (Out_{reg}). The blink detection head consists of a layer normalization LN, followed by a linear transformation:

$$Out_{cls} = Linear(LN(\mathbf{Z^0})). \tag{4}$$

Meanwhile, $\mathbf{Z}^{1:T} \in \mathbb{R}^{T\times D}$ is then fed into a fully connected layer to regress blink strength Out$_{reg} \in \mathbb{R}^T$ of each frame. This parameter represents the intensity of eye blinking in each frame, with a value of 1 indicating a completely closed eye and a value of 0 indicating a fully open eye:

$$Out_{reg} = Softmax(Linear(LN(\mathbf{Z}^{1:T}))). \tag{5}$$

4.3 Loss Function

The total loss for BlinkFormer consists of two parts: cross-entropy (CE) loss for blink detection and mean squared error (MSE) loss for blink strength estimation.

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{CE}} + (1-\alpha) \cdot \mathcal{L}_{\text{MSE}}, \tag{6}$$

where α is typically chosen based on the relative importance of the two tasks. If the dataset does not have annotations for blink strength, then α is set to 1.

We define the task of video blink detection as a binary classification problem. Let p_i be the predicted probability of blink and y_i be the ground truth for video clip x_i. Then, the binary cross-entropy loss is defined as:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log p_i + (1-y_i)\log(1-p_i)), \tag{7}$$

where N is the batch size.

For blink strength estimation, we use mean squared error loss to measure the difference between the predicted strength $s_{i,t}$ and the ground truth $y_{i,t}$ for each frame:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N \times T} \sum_{i=1}^{N} \sum_{t=1}^{T} (s_{i,t} - y_{i,t})^2. \tag{8}$$

To ensure that the predicted strength probabilities for each frame remain within the range of 0 to 1, a sigmoid activation is applied to the predicted strengths s.

5 Experiments

In this section, we first introduce the experimental details. Next, we compare the performance of our proposed BlinkFormer with other blink detection methods. Finally, we conduct cross-dataset evaluation between SynBlink and HUST-LEBW.

5.1 Implementation Details

The BlinkFormer takes a sequence of 13 eye frames (each with a size of 48×48 pixels) as input. To prevent overfitting, we employed data augmentation techniques such as random rotation, color jitter, and horizontal flip. The training, validation, and testing ratio for the SynBlink dataset is 6:2:2, while the training and testing ratio for HUST-LEBW dataset follows the original paper [20]. We empirically set α to 0.01. The Adam optimizer [23] was used with an initial learning rate of 1e-3. The network was trained for 500 epochs with batch size 32 on 4 NVIDIA GeForce RTX 3090 graphics cards. Precision, Recall, and F1-score are used as evaluation metrics.

5.2 Performance Comparision of Blink Detection Methods

We conducted a comparative study of our proposed BlinkFormer against current state-of-the-art blink detection algorithms on the challenging HUST-LEBW [20] real-world dataset, as presented in Table 2. The results demonstrate that our BlinkFormer achieves the highest F1-score (84.31%) as compared to other methods. Although Eye Template [5] achieves the highest precision of 98.28%, it suffers from very low recall and is not suited for real-world applications. LRCN [26] demonstrated the highest recall of 89.92%, but its precision is relatively low. In contrast, our method effectively balances precision and recall and ranks in the top three for all three metrics. It is worth noting that the methods marked with an asterisk (*) in the table represent our reimplementation version since the original papers were not open-sourced. Moreover, the red, green, and yellow colors in the table correspond to the best, sub-optimal, and third-best results, respectively. Overall, our experimental findings strongly corroborate the superiority of BlinkFormer over existing blink detection methods, providing a robust foundation for developing more accurate and reliable eye-blink detection systems in the future.

5.3 Cross-dataset Evaluation

Table 3 shows the evaluation results of our proposed model on SynBlink and HUST datasets. We observe that the performance of the model trained on the cross-domain dataset is lower

Method	**F1**	**Precison**	**Recall**
Variance Map (ver.) [2]	51.58%	49.03%	54.41%
Variance Map (hor.) [2]	55.53%	52.25%	59.35%
Variance Map (flow.) [2]	47.10%	48.30%	46.02%
Eye Template [5]	33.28%	**98.28%**	20.12%
Motion Vector [13]	51.58%	49.03%	54.41%
EAR [4]	42.95%	61.15%	33.12%
LRCN [26]	**78.52%**	69.69%	**89.92%**
mEBAL [9]	75.42%	67.14%	**87.77%**
HUST [20]	78.18%	75.82%	80.69%
3D PBNN* [3]	**82.03%**	**78.36%**	86.07%
BlinkFormer (Ours)	**84.31%**	**82.06%**	**86.69%**

Table 2: Comparation of Various Methods on HUST-LEBW dataset

than that trained on the same-domain dataset. This is likely due to the domain distribution differences between the two datasets. What's more, we also observe that using blink strength estimation (BSE) head leads to better prediction results when the model is trained on SynBlink. Furthermore, integrating blink strength estimation into the blink detection task for videos can also enhance cross-domain performance and improve the model's generalization ability to some extent. This can be attributed to the fact that the blink strength estimation task offers a more stringent constraint, thereby facilitating the overall accuracy of blink detection in diverse video settings.

Model	Test Train	HUST-LEBW	SynBlink
BlinkFormer	HUST-LEBW	84.31%	67.73%
BlinkFormer	SynBlink	70.90%	93.71%
BlinkFormer (with BSE head)	SynBlink	71.35%	94.67%

Table 3: F1-score of Cross-dataset Evaluation

6 Conclusion

For video blink detectoin task, we propose a workflow to synthesize blinking videos that can replicate human blinks under different scenarios. Based on this process, we introduced a dataset named SynBlink, which currently holds the largest number of video clips. Concurrently, our proposed BlinkFormer is the first to utilize transformer structures for blink detection, achieving the best F1-score. In the future, further research can focus on investigating and developing methods that address the domain gap between simulated and real data, as well as exploring effective ways to facilitate domain transfer between these two types of data.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62372019.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[2] Shubhajit Basak, Peter Corcoran, Faisal Khan, Rachel Mcdonnell, and Michael Schukat. Learning 3d head pose from synthetic data: A semi-supervised approach. *IEEE Access*, 9:37557–37573, 2021.

[3] Salah Eddine Bekhouche, I Kajo, Y Ruichek, and Fadi Dornaika. Spatiotemporal cnn with pyramid bottleneck blocks: Application to eye blinking detection. *Neural Networks*, 152:150–159, 2022.

[4] Jan Cech and Tereza Soukupova. Real-time eye blink detection using facial landmarks. *Cent. Mach. Perception, Dep. Cybern. Fac. Electr. Eng. Czech Tech. Univ. Prague*, pages 1–8, 2016.

[5] Michael Chau and Margrit Betke. Real time eye tracking and blink detection with usb cameras. Technical report, Boston University Computer Science Department, 2005.

[6] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[7] Timothy F. Cootes. Talking face video dataset. https://personalpages.manchester.ac.uk/staff/timothy.f.cootes/data/talking_face/talking_face.html, accessed May 10, 2023.

[8] Kevin Cortacero, Tobias Fischer, and Yiannis Demiris. Rt-bene: A dataset and baselines for real-time blink estimation in natural environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[9] Roberto Daza, Aythami Morales, Julian Fierrez, and Ruben Tolosana. Mebal: A multimodal database for eye blink detection and attention level estimation. In *Companion Publication of the 2020 International Conference on Multimodal Interaction*, pages 32–36, 2020.

[10] Lagunes-Ramírez Derick, González-Serna Gabriel, Lopez-Sánchez Máximo, Fragoso-Díaz Olivia, Castro-Sánchez Noé, and Olivares-Rojas Juan. Study of the user's eye tracking to analyze the blinking behavior while playing a video game to identify cognitive load levels. In *2020 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, volume 4, pages 1–5. IEEE, 2020.

[11] Matjaz Divjak and Horst Bischof. Eye blink based fatigue detection for prevention of computer vision syndrome. In *MVA*, pages 350–353, 2009.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] Tomas Drutarovsky and Andrej Fogelton. Eye blink detection using variance of motion vectors. In *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part III*, pages 436–448. Springer, 2015.

[14] Tobias Fischer, Hyung Jin Chang, and Yiannis Demiris. Rt-gene: Real-time eye gaze estimation in natural environments. In *Proceedings of the European conference on computer vision (ECCV)*, pages 334–352, 2018.

[15] Andrej Fogelton and Wanda Benesova. Eye blink completeness detection. *Computer Vision and Image Understanding*, 176:78–85, 2018.

[16] D González-Ortega, FJ Díaz-Pernas, Míriam Antón-Rodríguez, Mario Martínez-Zarzuela, and JF Díez-Higuera. Real-time vision-based eye state detection for driver alertness monitoring. *Pattern Analysis and Applications*, 16:285–306, 2013.

[17] Chao Gou, Yue Wu, Kang Wang, Kunfeng Wang, Fei-Yue Wang, and Qiang Ji. A joint cascaded framework for simultaneous eye detection and eye state estimation. *Pattern Recognition*, 67:23–31, 2017.

[18] Ezra Thess Mendoza Guevarra. *Modeling and Animation Using Blender: Blender 2.80: The Rise of Eevee*. Apress, 2019.

[19] Poly Haven. Poly Haven • Poly Haven — polyhaven.com. https://polyhaven.com/. [Accessed 14-May-2023].

[20] Guilei Hu, Yang Xiao, Zhiguo Cao, Lubin Meng, Zhiwen Fang, Joey Tianyi Zhou, and Junsong Yuan. Towards real-time eyeblink detection in the wild: Dataset, theory and practices. *IEEE Transactions on Information Forensics and Security*, 15:2194–2208, 2019.

[21] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[22] Ki Wan Kim, Hyung Gil Hong, Gi Pyo Nam, and Kang Ryoung Park. A study of deep cnn-based classification of open and closed eyes using a visible light camera sensor. *Sensors*, 17(7):1534, 2017.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] Aleksandra Królak and Paweł Strumiłło. Eye-blink detection system for human–computer interaction. *Universal Access in the Information Society*, 11:409–419, 2012.

[25] Yuriy Kurylyak, Francesco Lamonaca, and Giovanni Mirabelli. Detection of the eye blinks for human's fatigue monitoring. In *2012 IEEE International Symposium on Medical Measurements and Applications Proceedings*, pages 1–4. IEEE, 2012.

[26] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *2018 IEEE international workshop on information forensics and security (WIFS)*, pages 1–7. IEEE, 2018.

[27] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

[28] Ang Liu, Zhichao Li, Lang Wang, and Yong Zhao. A practical driver fatigue detection algorithm based on eye state. In *2010 Asia Pacific conference on postgraduate research in microelectronics and electronics (PrimeAsia)*, pages 235–238. IEEE, 2010.

[29] Antonio Maffei and Alessandro Angrilli. Spontaneous blink rate as an index of attention and emotion during film clips viewing. *Physiology & Behavior*, 204:256–263, 2019.

[30] Gang Pan, Lin Sun, Zhaohui Wu, and Shihong Lao. Eyeblink-based anti-spoofing in face recognition from a generic webcamera. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. IEEE, 2007.

[31] Joan K Portello, Mark Rosenfield, and Christina A Chu. Blink rate, incomplete blinks and computer vision syndrome. *Optometry and vision science*, 90(5):482–487, 2013.

[32] Ritabrata Sanyal and Kunal Chakrabarty. Two stream deep convolutional neural network for eye state recognition and blink detection. In *2019 3rd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*, pages 1–8. IEEE, 2019.

[33] Fengyi Song, Xiaoyang Tan, Xue Liu, and Songcan Chen. Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients. *Pattern Recognition*, 47(9):2825–2838, 2014.

[34] John A Stern, Donna Boyer, and David Schroeder. Blink rate: a possible measure of fatigue. *Human factors*, 36(2):285–297, 1994.

[35] Diego Torricelli, Michela Goffredo, Silvia Conforto, and Maurizio Schmid. An adaptive blink detector to initialize and update a view-basedremote eye gaze tracking system in a natural scenario. *Pattern Recognition Letters*, 30(12):1144–1150, 2009.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[37] Feng Yutian, Hu Dexuan, and Ning Pingqiang. A combined eye states identification method for detection of driver fatigue. 2009.