electronics

MDPI

Article

Spiking Feature-Driven Event Simulation with Movement-Aware Polarity Integration

Jiwoong Oh, Byeongjun Kang, Hyungsik Shin and Dongwoo Kang *

School of Electronic and Electrical Engineering, Hongik University, Seoul 04066, Republic of Korea; jwoh3923@g.hongik.ac.kr (J.O.); qudwns2692@g.hongik.ac.kr (B.K.); hyungsik.shin@hongik.ac.kr (H.S.)
* Correspondence: dkang@hongik.ac.kr

Abstract

Event-based face detection has attracted significant interest due to the unique advantages of event cameras, including high temporal resolution, high dynamic range, and low power consumption. However, the lack of annotated public datasets remains a major challenge for training effective event-based face detection models. In this paper, we propose a spiking feature-driven synthetic event generation framework that utilizes a spiking neural network (SNN) in conjunction with a pretrained convolutional backbone to generate synthetic event representations from a single RGB image. To incorporate motion-induced ON/OFF polarity information, we introduce a movement-aware polarity integration (MPI) module that assumes four directional facial movements. An event-similarity score is further employed to select representations most consistent with real event data for training. Unlike conventional approaches relying on video-based simulators, our method enables efficient synthetic event dataset construction without requiring video inputs or additional simulation training. Experimental results on the N-Caltech101 dataset demonstrate a face detection accuracy of 99.91%, outperforming existing event-based face detection methods.

Keywords: event camera; synthetic event generation; spiking neural network; spiking feature; event polarity generation; face detection

1. Introduction

Neuromorphic cameras, also known as event cameras, are next-generation image sensors that operate asynchronously, unlike traditional frame-based RGB cameras [1,2]. Event cameras detect brightness changes at each pixel independently, generating event streams at extremely high speeds [1,2]. These event streams contain information on time, pixel location, and polarity of the brightness change. The key advantages of event cameras include low power consumption, as they only generate data when there is a change in brightness, and their ability to produce high-quality images in both dark and bright environments due to their sensitivity to brightness changes. Additionally, event cameras capture data at very short time intervals, making them ideal for accurately tracking fast-moving objects without motion blur [3,4]. Their ability to process data in real-time allows for low-latency response. As a result, event cameras have garnered significant attention in computer vision tasks related to mobility [5,6], robotics [7,8], and mixed reality [9–11].

Research on event-based face detection remains in its early stages compared to frame-based methods. While many object detection studies [12,13] have explored the advantages of event cameras, face detection has received relatively less attention. Most event-based studies have focused on tasks like blink detection [14,15], drowsiness detection [16], and

Academic Editor: George A. Tsihrintzis

Received: 24 February 2026
Revised: 26 March 2026
Accepted: 27 March 2026
Published: 29 March 2026

facial expression recognition [17]. For blink detection, ref. [14] applied Gaussian tracking on raw event streams, while ref. [15] converted event streams into event frame images and detected faces and eye regions, followed by pixel-based blink detection. Drowsiness detection [16] employed a support vector machine to analyze hand-crafted features, and ref. [17] generated a facial expression dataset.

Challenges in event-based face detection include the scarcity of public training datasets and the sparse nature of event images, which complicate manual annotation. For example, ref. [16] collected a real event dataset from 26 subjects in a vehicle, while refs. [15,17] used video-based simulators [18] to convert RGB videos into synthetic event streams. However, public face video datasets are limited, and annotating these datasets is labor-intensive. A style transfer-based method [9] generates synthetic event frames from a single image, but this approach doesn't fully capture the characteristics of real event data.

In this paper, to address the lack of annotated event training datasets, we propose a spiking feature-driven synthetic event generation framework using a spiking neural network (SNN) as shown in Figure 1. Given a static RGB image, feature maps are first extracted using a pretrained convolutional backbone and subsequently processed by a Leaky Integrate-and-Fire (LIF)-based spiking layer to obtain sparse binary feature activations that better align with the characteristics of real event streams. To further approximate motion-induced polarity variations from a single static image, we introduce a movement-aware polarity integration (MPI) module that assumes horizontal and vertical facial movements and integrates positive and negative spike responses accordingly. Finally, an event-similarity score is computed using a style similarity metric against a real DAVIS 346 anchor event image to select the most event-like spiking feature maps. The selected representations are then used as synthetic training samples for a RetinaFace-based face detector [19]. The main contributions of this work are as follows:

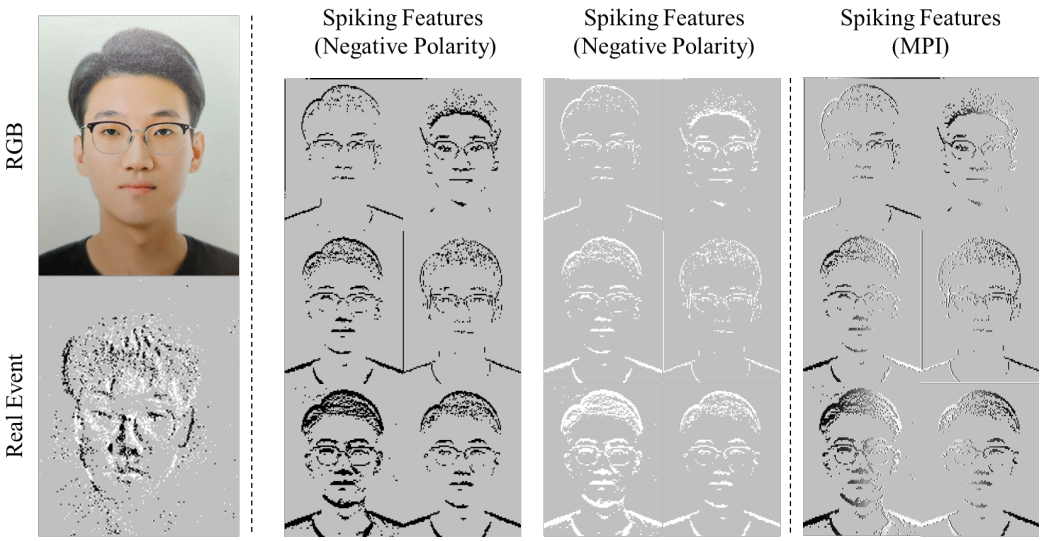

Figure 1. Examples of an RGB image and a real event image (left), along with synthetic event-like representations generated from the RGB image using spiking features (right). The generated spiking representations are utilized as synthetic training samples for event-based face detection.

- The primary contribution of this work is the introduction of a spiking feature-driven synthetic event generation framework that employs a Leaky Integrate-and-Fire (LIF)-based spiking neural layer to convert continuous deep convolutional features into sparse binary representations from a single static RGB image, without requiring video-based event simulators or additional event-simulation training while preserving RGB face annotations.

- We propose a movement-aware polarity integration (MPI) module that reconstructs motion-consistent event polarity by integrating positive and negative spike responses under assumed horizontal and vertical facial movements.
- We introduce an event-similarity score based on a style similarity metric against a real event anchor image to select event-like spiking feature maps and construct a large-scale synthetic training dataset.
- Extensive experiments on the N-Caltech101 Face dataset [20] validate the effectiveness of the proposed spiking synthetic event training pipeline.

2. Related Works

2.1. Event-Based Face Detection

Face detection refers to the task of estimating face bounding boxes without relying on predefined scale and position priors. As one of the most researched areas in computer vision, frame-based face detection has evolved from simple handcrafted feature-based methods to modern deep learning approaches. After early methods like Haar-like features with an AdaBoost classifier [21,22], the advent of deep learning networks led to significant performance improvements in both object detection [23–25] and face detection [19,26,27]. Face detection faces several challenges related to variations in scale, pose, occlusion, expression, makeup, illumination, and blur, among others. To address these challenges, extensive research has been conducted. Earlier two-stage methods [28,29], which separate the classification and box regression tasks, were commonly used in face detection. However, recent state-of-the-art methods have shifted towards one-stage designs [19,27,30], where face classification and box regression are handled in a single pass, resulting in faster performance. To further improve multi-scale face detection, many of these methods utilize feature pyramids [31], which extract features from different layers of the network and combine them to improve detection across multiple scales.

Compared to frame-based face detection, research on event-based face detection has been relatively limited. Most event-based detection studies have focused on static objects [12,13], while event-based face detection remains underexplored. One major reason is that event-based face detection presents unique challenges because the human face is a deformable object that undergoes continuous changes. Since event cameras only detect changes in pixel intensity, this challenge becomes even more pronounced. The dynamic nature of facial features, including eye movements, changes in facial expressions, occlusions, makeup, lighting, and pose, makes accurate face detection in event-based systems more difficult [9,11]. Most existing research has focused on blink detection [14,15] due to the advantage of event cameras in capturing rapid movements. One approach uses a Gaussian tracker to detect eye blinking from raw event streams, focusing solely on blinks without addressing other facial feature detection [14]. Another method [15] applies the GR-YOLO network for high-precision face and eye region detection, though this study relied on synthetic event data generated by the ESIM event simulator [18]. Drowsiness detection [16] has also been explored, where a support vector machine (SVM) classifier [32] was used for drowsiness detection, but accuracy for face or eye detection was not evaluated. Similarly, work in event-based facial expression recognition has not assessed the accuracy of face, eye, or pupil detection using actual event cameras [17].

2.2. Synthetic Event Generation

A widely adopted technique to address the challenges in event camera dataset generation is to project frame-based public datasets [33–35] onto an LCD monitor and then record the projected images using a real event camera [20,36]. This can be accomplished by either physically moving the event camera [20] or by manipulating the displayed images

on the monitor [36]. The latter approach, involving the movement of the on-screen image, offers the advantage of eliminating the need for additional equipment while still producing data that closely resembles well-established frame-based datasets. For example, using an event camera like the DVS 128 sensor [2], this method can create datasets well-suited for assessing performance in event-based object classification. When it comes to public face datasets, however, the availability is much more restricted. Moreover, the few datasets that do exist often lack detailed annotations [17,20], making it even more difficult to develop robust face detection models for event cameras.

To address the scarcity of event training datasets, several studies have focused on developing event simulators that convert RGB video datasets into synthetic event streams [18,37,38]. Simulators like ESIM [18], v2e [37], and Vid2E [38] generate events by detecting changes in pixel intensity between frames. These DVS simulators model event generation by calculating the relationship between pixel intensity variations and a triggering threshold. For instance, ESIM [18] and Vid2E [38] employ a Gaussian-distributed threshold to simulate spatially varying noise. Additionally, simulators like v2e [37] incorporate shot noise into the event model. However, these techniques require a large amount of RGB video datasets with face annotations. Public face video datasets are scarce, and most do not provide the necessary annotations. Furthermore, manually annotating video datasets is extremely labor-intensive. As a result, event camera-based face and eye detection algorithms often struggle with the lack of adequately annotated training databases.

To overcome the challenges of RGB video dataset-based synthetic event generation, some studies have proposed methods to translate a single RGB image into an event image using image-to-image translation [9]. One such method is an adaptive Styleflow algorithm [39], which generates synthetic event images from a single RGB frame [9], rather than relying on traditional video-based event simulators. This approach assigns the event style to the RGB image and transforms it into an event-style image, allowing the direct use of face annotations from RGB frame images, making it highly efficient for face and eye detection. However, this method does not fully capture the asynchronous characteristics of real event data. Additionally, using a style transfer algorithm [39] to generate the synthetic event dataset is a time-consuming process, as generating 70,000 synthetic event images took three days on an NVIDIA RTX 3090 GPU [9]. They primarily focus on visual texture rather than the inherent asynchronous, sparse properties of real event data.

Recently, SNNs have gained attention for their structural similarity to event cameras, as they process information using discrete, sparse spikes rather than continuous values. While SNNs are inherently suited for neuromorphic data, their application in generating synthetic events from static RGB images has been underexplored. In this work, we utilize an LIF-based SNN layer to convert continuous deep features into sparse binary representations that better align with real event characteristics. Combined with the MPI module, this approach enables efficient synthetic event generation from static RGB images without requiring video data or additional training.

2.3. Event-Similarity

Traditional perceptual metrics like PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index) [40] are typically used for tasks like denoising and super-resolution, where the original and processed images are expected to have nearly identical structures. These metrics work well for measuring pixel-wise similarity and structural integrity in tasks where high-frequency details are restored [40]. However, PSNR and SSIM fall short when comparing real event data with synthetic event images generated from RGB frames. In these cases, the structure of the event image can be significantly different from

the RGB image because the real event images were captured from actual subjects, while the RGB images were taken from public datasets.

Given this limitation, we turned to metrics [41,42] used in style transfer algorithms [39,43,44], which are designed to evaluate how well one image replicates the style of another, even when their structural content differs significantly. These metrics [41,42] focus on the perceptual similarity between images, including aspects like style and texture, rather than structural alignment. They are widely used in style transfer and image generation tasks to assess how well one image mimics the artistic style or texture of another, even if the content is different. For example, Learned Perceptual Image Patch Similarity (LPIPS) [42] is employed for filtering collections of images [42], while Fréchet Inception Distance (FID) [41] can detect overfitting in Generative Adversarial Networks (GANs) [45], or be used to compare the sensitivity of different GAN models to hyper-parameters [46]. By focusing on these style-similarity metrics, our proposed event-similarity measure quantitatively evaluates the similarity between deep features extracted from RGB images and real event data, even when their structures differ.

3. Method

3.1. Overall Framework

Our proposed event-based face detection framework involves synthetic event generation from a single RGB image, deep feature selection via our proposed event-similarity score, and event-based face detection training. First, a single RGB frame image is passed through the first convolutional layer of a pretrained deep neural network, VGG-16 [47]. These initial layers capture edges and textures, preserving the shape structure of the original image. However, continuous deep features fundamentally differ from the discrete, asynchronous nature of real events. To bridge this domain gap, the extracted continuous features are passed through an SNN equipped with LIF neurons, converting them into highly sparse, binary spiking representations. Subsequently, to simulate the dynamic properties of real event cameras, we introduce the movement-aware polarity integration module. The MPI module synthesizes positive and negative polarities under assumed multi-directional facial movements. Finally, the most realistic synthetic event images are selected using our proposed event-similarity score and utilized to train the RetinaFace [19] model from scratch. The overall process for generating the synthetic training dataset is illustrated in Figure 2.

3.2. Spiking Feature Generation via SNN

To approximate the sparse activation characteristics of real event cameras, we introduce an SNN layer following a pretrained convolutional backbone. Given an input RGB image I , deep features F are first extracted via the initial convolutional layer of a pretrained VGG-16 [47]. We employ the LIF neuron model to convert continuous deep feature maps into discrete spike responses. The continuous-valued deep features are accumulated as membrane potentials in LIF neurons, and when the potential exceeds a predefined threshold, a binary spike is generated, and the membrane potential is reset. The resulting binary spike map is denoted as $S \in \{0,1\}^{H\times W}$, where $S(x) = 1$ indicates a spike firing at spatial position x. Through this integrate-and-fire mechanism, dense convolutional features are converted into sparse binary spiking feature maps that better align with the spatial sparsity and discrete nature of real event data without requiring video-based simulation.

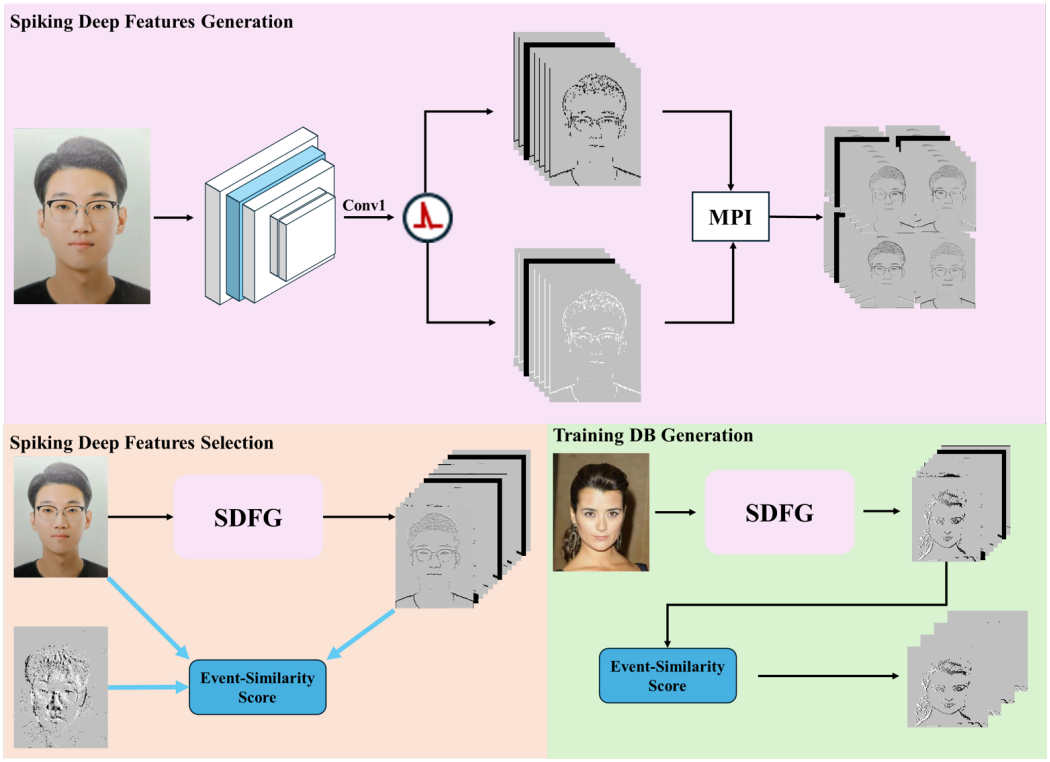

Figure 2. Overall process of SNN-based synthetic event dataset generation. Deep features extracted from a single RGB image are converted into binary spike responses using a LIF-based spiking layer, followed by polarity integration via the proposed MPI module. Synthetic training samples are then selected using the event-similarity score.

3.3. Movement-Aware Polarity Integration

While the SNN layer provides structural sparsity, real event cameras also generate distinct positive (ON) and negative (OFF) polarities depending on object motion and brightness gradients. Since our method extracts features from a single static RGB image, approximating motion-induced polarity variation is necessary to enhance the event-like characteristics of the synthetic representations. To achieve this, we propose the movement-aware polarity integration (MPI) module, as shown in Figure 3.

The MPI module takes this binary spiking feature map and assumes four primary directional facial movements: horizontal (left, right) and vertical (up, down). We define the set of displacement vectors as:

$$\mathcal{M} = \{(n,0),\ (-n,0),\ (0,n),\ (0,-n)\} \tag{1}$$

where n denotes the shift magnitude in pixels.

For each assumed movement direction $\mathbf{m} \in \mathcal{M}$, polarity-specific spike maps are spatially shifted to simulate motion-induced brightness changes. Specifically, the positive (ON) polarity map is generated by shifting the spike map along the movement direction:

$$S^{+}_{\mathbf{m}}(x) = S(x - \mathbf{m}) \tag{2}$$

while the negative (OFF) polarity map retains the original spike positions:

$$S^{-}(x) = S(x) \tag{3}$$

Note that S^{-} is defined independently of $\mathbf{m}$, as the negative polarity represents brightness decrease at the original position regardless of movement direction.

During this process, overlapping regions between positive and negative polarity responses may occur, i.e., pixels where $S^+_{\mathbf{m}}(x) = 1$ and $S^-(x) = 1$ simultaneously. To resolve polarity conflicts in these regions, we employ a sigmoid-based probabilistic selection. The selection probability is designed such that the shifted positive polarity dominates in regions along the assumed movement direction, while the original negative polarity is preserved on the opposite side. Specifically, the probability of selecting the positive polarity at pixel x is defined as:

$$P_{\text{pos}}(x) = \sigma(k \cdot d_{\mathbf{m}}(x)) \tag{4}$$

where σ denotes the sigmoid function, $d_{\mathbf{m}}(x)$ is the normalized spatial distance along the movement-aligned axis from the image center, ranging from -1 to 1, and k is the steepness parameter that controls the sharpness of the transition.

Using the above definitions, the synthetic event map $E_{\mathbf{m}}$ for a given direction $\mathbf{m}$ is determined as:

$$E_{\mathbf{m}}(x) = \begin{cases} +1 & \text{if } S^+_{\mathbf{m}}(x) = 1 \text{ and } S^-(x) = 0 \\ -1 & \text{if } S^+_{\mathbf{m}}(x) = 0 \text{ and } S^-(x) = 1 \\ +1 \text{ w.p. } P_{\text{pos}}(x),\ -1 \text{ otherwise} & \text{if } S^+_{\mathbf{m}}(x) = 1 \text{ and } S^-(x) = 1 \\ \pm 1 \text{ w.p. } r/2,\ 0 \text{ otherwise} & \text{if } S^+_{\mathbf{m}}(x) = 0 \text{ and } S^-(x) = 0 \end{cases} \tag{5}$$

where the last case introduces salt-and-pepper noise at ratio r in background regions to emulate intrinsic background activity observed in dynamic vision sensors.

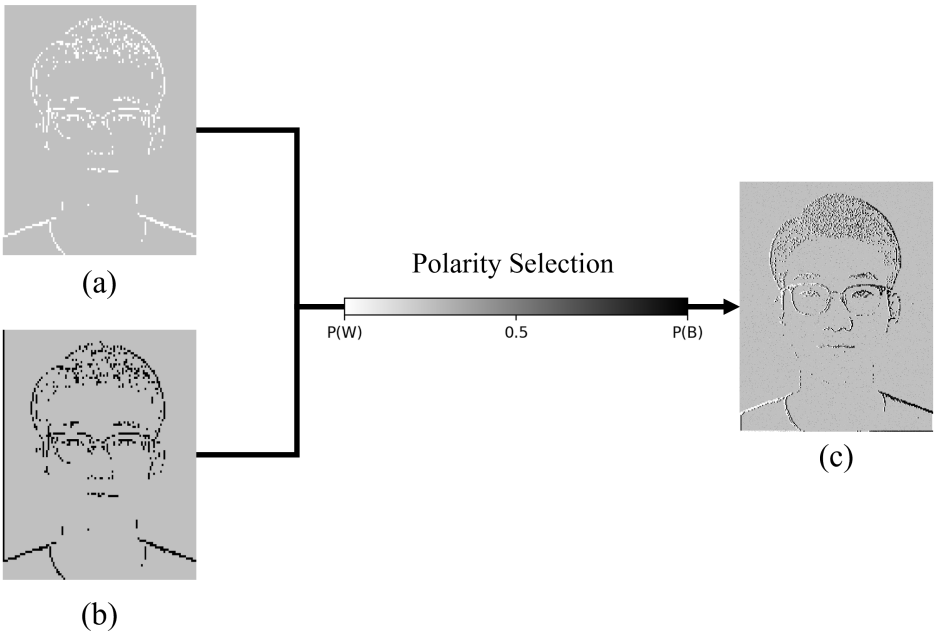

Figure 3. Movement-aware polarity integration. Positive polarity (**a**), which is shifted to the left relative to negative polarity (**b**), and the original negative polarity are combined via sigmoid-based polarity selection, producing the integrated result (**c**).

Based on these assumed movements, the MPI module assigns positive polarity (brightness increase) to the leading edges and negative polarity (brightness decrease) to the trailing edges. This process is performed independently for each of the four directions, yielding a set of four synthetic event maps $E_{syn} = \{E_{\mathbf{m}}\}_{\mathbf{m} \in \mathcal{M}}$ from a single input image. All four directional event maps are used as training samples, effectively augmenting the dataset by

a factor of four while preserving facial structures with ON/OFF polarity characteristics consistent with real event streams.

3.4. Event Similarity

To quantitatively measure the similarity between our synthesized event features E_{syn} and real event images X_e captured by event cameras, classical perceptual metrics like PSNR or SSIM [40] are generally inconsistent with human perception [48]. Furthermore, due to operational differences and slight variations in sensor positioning, capturing identical shape structures between RGB and event cameras is extremely difficult. Therefore, our primary objective is to select synthetic spiking features that best reflect the unique characteristics of event images based on appearance and texture style, rather than exact shape matching. We utilize similarity metrics commonly employed in neural style transfer: Fréchet Inception Distance (FID) [41] and Art Fréchet Inception Distance (ArtFID) [48], which incorporates Learned Perceptual Image Patch Similarity (LPIPS) [42]. The FID [41] measures style similarity and is formulated as follows:

$$\mathrm{FID}(X_e, E_{syn}) = ||\mu_e - \mu_{syn}||_2^2 + \mathrm{Tr}\left(\Sigma_e + \Sigma_{syn} - 2(\Sigma_e\Sigma_{syn})^{\frac{1}{2}}\right) \tag{6}$$

where (μ_e, Σ_e) and $(\mu_{syn}, \Sigma_{syn})$ represent the mean and covariance of the Inception features of the real event image X_e and the synthetic event image E_{syn}, respectively.

ArtFID combines style and content preservation to assess overall perceptual content. ArtFID is calculated as:

$$\mathrm{ArtFID}(E_{syn}, X_c, X_e) = 1 + \left(\frac{1}{N}\sum_{i=1}^{N} d(X_c^{(i)}, E_{syn}^{(i)})\right) \cdot (1 + \mathrm{FID}(X_e, E_{syn})) \tag{7}$$

where X_c represents the original RGB frame image, d denotes the perceptual distance in LPIPS [42], and N is the number of images used to compare similarity. Using these style-similarity metrics, we compute the event-similarity score to construct a highly realistic synthetic event training dataset.

3.5. Event-Based Face Detection

The synthetic event features selected via the event-similarity score are used as the training dataset for event-based face detection based on the practical single-stage face detector, RetinaFace [19]. RetinaFace is a joint learning-based algorithm that detects face bounding boxes and simultaneously identifies five key facial landmarks. Using the publicly available RGB dataset CelebA [49], which includes annotation information, we extracted spiking features, integrated polarities via MPI, and selected those with high similarity to a real event anchor image. Because our pipeline perfectly preserves spatial dimensions, no additional bounding box annotation is required. RetinaFace's network architecture comprises the backbone network (MobileNet [50]), the feature fusion pyramid, and the context module [19]. We trained the model based on this framework, performing training from scratch using our synthetically generated spiking event dataset, thereby validating the robustness and efficiency of the proposed generation pipeline.

4. Experiments

4.1. Experiment Setup

4.1.1. Implementation Details

For the convolutional backbone, we utilized pretrained network models—VGG-16 [47] and ResNet-50 [51], trained on ImageNet [52]. For spiking feature formation, the LIF neurons were configured with a membrane decay parameter of 2.0, a firing

threshold of 1.5, and time steps of 8, with the initial membrane potential set to zero. Other implementation details followed the default configuration of the LIF module in the snnTorch library. In the MPI module, the pixel shift magnitude n was set to 5, the steepness parameter k of the sigmoid-based probabilistic selection was set to 5.0, and the salt-and-pepper noise ratio r was set to 0.02. The event-similarity score metrics used were FID [41], LPIPS [42], and ArtFID [48]. Synthetic feature representations exhibiting the highest similarity to a real event image captured by a DAVIS 346 sensor (iniVation AG, Zurich, Switzerland) [53] were selected for training. Face detection training was conducted on a PC running Ubuntu 20.04.6 LTS, equipped with an Nvidia RTX 4090 GPU (24 GB). Learning from scratch training strategy was employed, with the input image size set to 640 × 640, batch size set to 160, and training conducted over 300 epochs. The other parameters were set as follows: an SGD optimizer with a learning rate of 0.001, momentum of 0.9, and a gamma value of 0.1.

4.1.2. Datasets

For synthetic event image generation from RGB inputs, we utilized 5000 images selected from the CelebA dataset [49], which contains approximately 200,000 face images from 10,000 individuals. These images were used for spiking feature generation and ranked based on their similarity to a real event image captured using a DAVIS 346 event camera [53]. The highest-ranked spiking feature representations were then processed by the MPI module, which assumes four primary motion directions (left, right, up, and down), resulting in the generation of 20,000 synthetic event images for face detection training. The generated synthetic event images were resized to 640 × 640 for consistency during training.

To validate the effectiveness of our synthetic event generation method for event-based face detection, we used the publicly available N-Caltech101 dataset [20] as a benchmark. We utilized 3458 face images from N-Caltech101, comprising 100 object classes, including the face category. The dataset was created by displaying frame-based RGB datasets [33] on an LCD monitor and capturing the data with a real event camera, Prophesee Gen3 ATIS [54], generating event data for each class. For the N-Caltech101 dataset, we processed the raw event data by accumulating event streams within a 30 ms time window to generate single-event frame images. Each pixel in these frames represents either a positive or negative event polarity, or no event. Since the dataset does not include face bounding box annotations, detection accuracy was determined by considering detections successful if they covered more than 70% of the face region.

4.2. Experimental Results

4.2.1. Synthetic Event Generation Evaluation

In Table 1, we present the event-similarity scores for the top three spiking feature-based synthetic event images, where VGG-16 was used as the pretrained CNN. The best result was obtained from the 28th channel of VGG-16's first convolution layer, after passing through the SNN (LIF) and MPI modules, achieving the lowest scores across all three metrics: ArtFID [48], FID [41], and LPIPS [42]. Lower values indicate higher similarity to real event images, and Table 1 displays the results in ascending order based on ArtFID. The top 3 spiking deep features exhibit noticeable similarities to real event images with varying levels of sparseness, reflecting the asynchronous nature of event data and making them suitable for face detection training. Figure 4 illustrates the synthetic event generation process, comparing the original RGB image, deep features from the first convolutional layer of VGG-16, spiking representations after the SNN LIF filter, and the final MPI-integrated output with real event images.

Table 1. Quantitative Comparison of Event-Similarity Scores on Synthetic Event Representations. The best results are highlighted in bold. The downward arrow indicates that lower values are better.

Selected Conv1 Channel	ArtFID ↓	FID ↓	LPIPS ↓
28	**38.69**	**22.57**	**0.641**
18	40.01	22.81	0.680
48	43.34	23.95	0.737

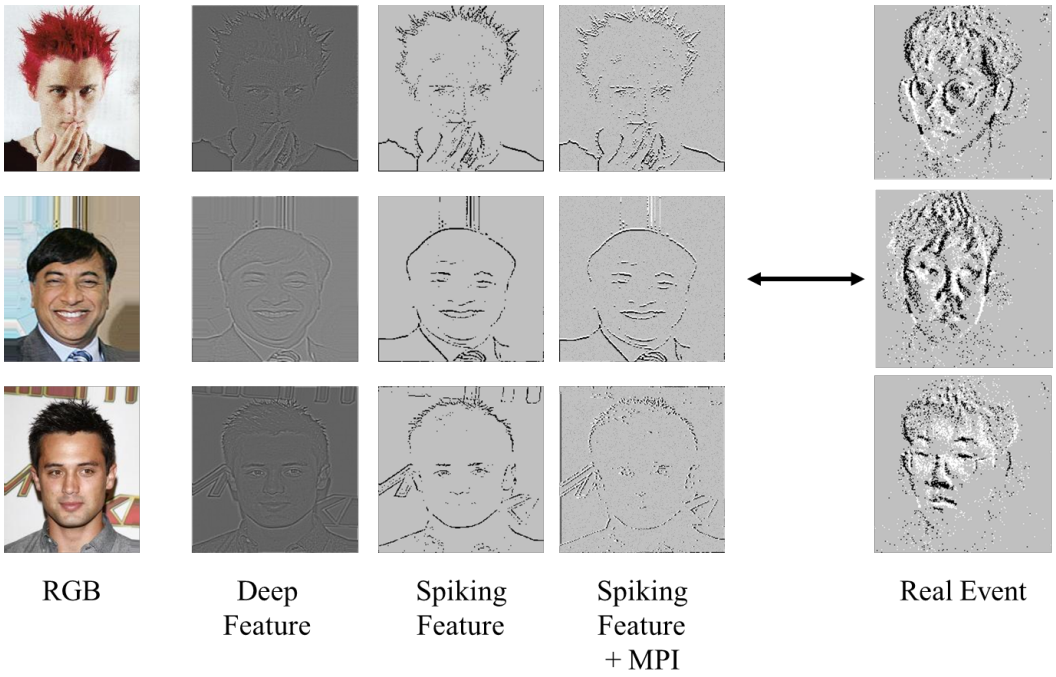

Figure 4. Visual comparison of each stage in the proposed pipeline and real event images.

4.2.2. Event-Based Face Detection Evaluation

To assess the effectiveness of the proposed synthetic event image generation based on spiking features as a training dataset, we evaluated our RetinaFace-based face detector [19], which was trained from scratch. The detector's performance was tested on real event camera datasets, the public N-Caltech101 dataset [20]. Out of 3458 face images from the N-Caltech101 dataset, our method successfully detected 3455 faces, yielding a detection accuracy of 99.91%, with only 24 missed detections. Figure 5 illustrates the detection results from N-Caltech101.

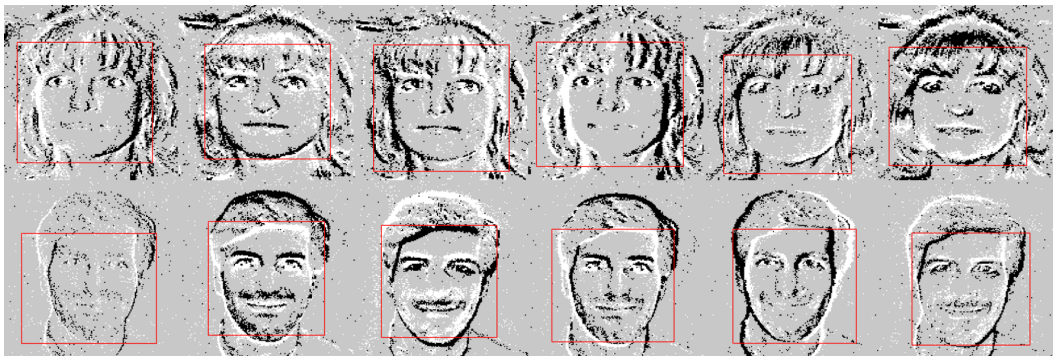

Figure 5. Face detection results on N-Caltech101 [20], with red boxes marking detected faces. Each row shows the same person while the event camera was moved to generate different event streams.

4.3. Ablation Study and Discussion

4.3.1. Ablation Study

Our proposed method integrates two key components: SNN-based spiking features and Movement-aware Polarity Integration (MPI). We conducted an ablation study on the N-Caltech101 dataset [20] to verify the effectiveness of each component as shown in Table 2. The baseline, a face detector trained with deep features from the first convolution layer of VGG-16, achieves 99.30% accuracy. When only the SNN (LIF) module is applied, the accuracy slightly decreases to 98.88%, suggesting that spiking activation alone introduces excessive sparsity without compensating polarity information. However, when the MPI module is additionally applied, the accuracy increases to 99.91%. This demonstrates that polarity information plays a critical role in event-based representations, and the proposed MPI module effectively leverages it to improve detection performance.

Table 2. Ablation Study on the Primary Components of the Proposed Method. The best results are highlighted in bold. A checkmark indicates that the corresponding component is included.

Model	SNN	MPI	Accuracy (%)
Baseline (CNN Deep Feature)			99.30
Single Polarity	✓		98.72
Ours	✓	✓	**99.91**

4.3.2. Comparison with Existing Methods

We conducted a comparison on the N-Caltech101 dataset [20], where our proposed method achieved an accuracy of 99.91% in event-based face detection. Lenz et al. [14] proposed a blink-based face detection method using the Prophesee Gen3 ATIS camera and reported an accuracy of 59%. Their approach relies on Gaussian tracking and a blink model, which is effective when blink events are present; however, motion in the surrounding scene may interfere with blink detection, potentially affecting face detection performance. Ryan et al. [15] achieved 90% face detection accuracy and 88.93% blink detection accuracy using a GR-YOLO network trained on synthetic event data generated from RGB video streams via the ESIM simulator [18]. However, the use of transformation-based video synthesis from static RGB images may limit motion diversity in the generated event data. Kang et al. [9] reported 99.4% accuracy by converting single RGB images into event-style representations using a style transfer approach. However, generating 70,000 synthetic event images required three days on an NVIDIA RTX 3090 GPU, and the evaluation was conducted on a limited test set of 1000 images from two subjects.

As an additional comparison, we evaluated previous SNN-based approaches [55–57] designed for object classification on the N-Caltech101 dataset. These methods reported classification accuracies below 80% across all object categories and did not provide separate results for the face class. In contrast, our approach specifically targets face detection and achieved an accuracy of 99.91% on the face category. For further evaluation, we also tested the event-style generation method proposed in [9] on N-Caltech101, obtaining detection accuracies of 56.15% with training from scratch and 86.23% with full fine-tuning. Detailed comparisons are provided in Table 3.

To further evaluate computational efficiency, we compare the proposed method with the approach in [9]. As shown in Table 4, the method in [9] requires approximately 72 h to generate 70,000 synthetic event images using an NVIDIA RTX 3090 GPU. In contrast, our method generates 80,000 synthetic event images in 21 min using only a CPU, achieving a significant speedup in image generation throughput. This demonstrates that

the proposed framework is highly efficient and practical for large-scale synthetic event dataset construction.

Table 3. Quantitative Comparisons of Face Detection Results. The best results are highlighted in bold.

Method	Dataset	Users	Detection	Classification	Other
Lenz et al. [14]	self-collected	10	59%	-	Blink Detection: 59%
Ryan et al. [15]		3	90%	-	Blink Detection: 88.93%
Kang et al. [9]		2	99.40%	-	Pupil Align: 97.20%
Zheng et al. [55]	N-Caltech101	385	-	79.47%	All Classes
Fang et al. [56]			-	78.20%	All Classes
Zhou et al. [57]			-	72.83%	All Classes
Kang et al. [9]			86.23%	-	Face Class
Ours	N-Caltech101	385	**99.91%**	**99.91%**	Face Class

Table 4. Computational efficiency comparison for synthetic event image generation. The proposed method achieves a ∼235× speedup over Kang et al. [9], operating entirely on CPU.

Method	Generated Synthetic Images	Total Time	Hardware
Kang et al. [9]	70,000	72 h	RTX 3090
Ours	80,000	21 min	CPU only

4.3.3. Discussion and Future Work

To further analyze the characteristics of the proposed synthetic event representations, we conduct a distribution-level analysis as shown in Figure 6. CNN-based deep features exhibit a continuous pixel value distribution, which fundamentally differs from real event camera outputs that only produce discrete ON (+1) and OFF (−1) events asynchronously, with the majority of pixels remaining inactive. In contrast, the proposed method naturally produces discrete bipolar outputs through SNN-based spiking activation, yielding a representation structurally consistent with real event data. Furthermore, the proposed method shows a polarity ratio (88.9% background, 5.7% ON, 5.5% OFF) closer to that of real N-Caltech101 events (78.0% background, 11.2% ON, 10.7% OFF), confirming that the proposed approach better reflects the statistical characteristics of real event streams. To further examine the robustness of the proposed event-similarity-based feature selection, we evaluate the effect of using multiple anchor images with varying event densities. Specifically, we collect 275 real event frames captured by a DAVIS 346 sensor and select five representative anchor images covering a wide range of event densities (2.30%, 3.28%, 5.04%, 5.36%, and 6.96%). For each anchor image, we independently perform ArtFID-based feature ranking. As shown in Table 5, the top-2 selected channels remain consistent across all anchor selections. These results demonstrate that the proposed feature selection strategy is robust to the choice of anchor image and is not significantly affected by variations in event density.

A limitation of the current MPI module is that it assumes simplified motion patterns limited to four directional movements, which may not fully capture complex real-world facial motion such as rotation, scaling, and expression changes. Future work will explore more advanced motion modeling approaches, such as optical flow-based methods, to better simulate realistic motion dynamics. Another limitation of this study is the use of a single evaluation metric based on detection accuracy, due to the lack of bounding box annotations in the N-Caltech101 dataset [20]. Standard object detection metrics such as precision,

recall, and mAP cannot be directly computed. Future work will focus on developing a high-quality event-based face detection dataset with reliable annotations to enable more comprehensive evaluation. Finally, evaluating temporal consistency and polarity accuracy is not directly applicable in our framework, as it operates on a single static RGB image and lacks paired RGB-event ground truth. Exploring such evaluation under controlled data acquisition remains future work.

Table 5. Robustness of ArtFID-based feature selection across anchor images with varying event densities. The top-2 selected channels remain consistent (28, 18) across all density levels.

Event Density	Selected Conv1 Channel Rank 1	Selected Conv1 Channel Rank 2
2.30%	28	18
3.28%	28	18
5.04%	28	18
5.36%	28	18
6.96%	28	18

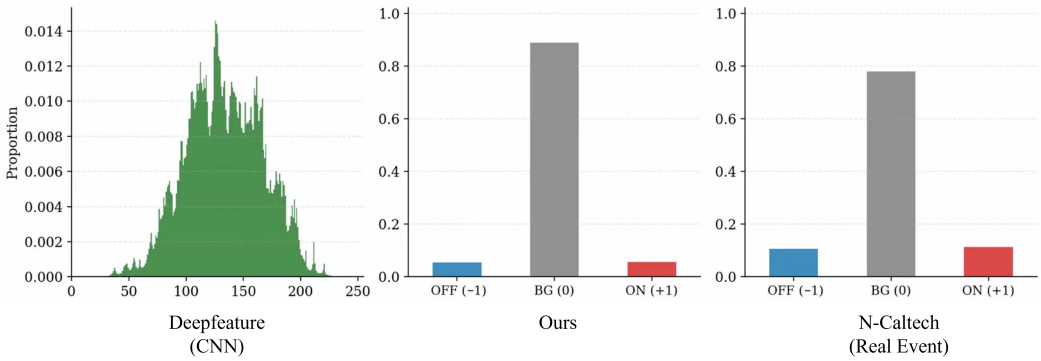

Figure 6. Pixel value distribution comparison. CNN deep features exhibit a continuous distribution (**left**), while the proposed spiking feature with MPI (**center**) and real event data from N-Caltech101 (**right**) show discrete bipolar (ON/OFF) distributions with dominant background proportion.

5. Conclusions

In this paper, we proposed a synthetic event-based face detection framework using SNN-based spiking feature formation and a movement-aware polarity integration module. By converting continuous deep features into polarity-consistent event-like representations from a single RGB image, the proposed method enables efficient synthetic event dataset construction without requiring video-based simulation. Experimental results on the N-Caltech101 dataset demonstrated a detection accuracy of 99.91%, outperforming existing event-based face detection approaches. Future work will focus on extending the proposed framework to fully event-only detection pipelines.

Limitation and Future Work

A limitation of our approach is that it has only been studied with conventional deep neural networks, not with newer models like Vision Transformers [58] or Diffusion models [59]. In the future, we plan to investigate these architectures to enhance event-like deep feature extraction.

Author Contributions: Conceptualization, J.O., B.K., H.S. and D.K.; methodology, J.O., B.K., H.S. and D.K.; software, J.O.; validation, J.O. and B.K.; formal analysis, J.O. and D.K.; investigation, J.O. and D.K.; resources, D.K.; data curation, J.O. and B.K.; writing—original draft preparation, J.O. and D.K.; writing—review and editing, J.O., H.S. and D.K.; visualization, J.O. and B.K.; supervision, D.K.; project administration, D.K.; funding acquisition, D.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00337012), in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2026-25487310), and in part by the 2026 Hongik University Innovation Support Program Fund.

Institutional Review Board Statement: Not applicable. This study did not involve human subjects research requiring institutional review. The study utilized publicly available datasets, and the human face images used for illustration purposes belong to the author.

Data Availability Statement: Publicly available datasets were analyzed in this study. The N-Caltech101 dataset can be found in [20], and the CelebA dataset can be found in [49].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Gallego, G.; Delbruck, T.; Orchard, G.; Bartolozzi, C.; Taba, B.; Censi, A.; Leutenegger, S.; Davison, A.J.; Conradt, J.; Daniilidis, K.; et al. Event-Based Vision: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 154–180. [CrossRef]
2. Lichtsteiner, P.; Posch, C.; Delbruck, T. A 128 × 128 120 dB 15 µs Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE J. Solid-State Circuits* **2008**, *43*, 566–576. [CrossRef]
3. Zhang, L.; Zhang, H.; Chen, J.; Wang, L. Hybrid Deblur Net: Deep Non-Uniform Deblurring with Event Camera. *IEEE Access* **2020**, *8*, 148075–148083. [CrossRef]
4. Zhang, S.; Sun, L.; Wang, K. A Multi-Scale Recurrent Framework for Motion Segmentation with Event Camera. *IEEE Access* **2023**, *11*, 80105–80114. [CrossRef]
5. Shariff, W.; Dilmaghani, M.S.; Kielty, P.; Moustafa, M.; Lemley, J.; Corcoran, P. Event Cameras in Automotive Sensing: A Review. *IEEE Access* **2024**, *12*, 51275–51306. [CrossRef]
6. Gelen, A.G.; Atasoy, A. An Artificial Neural SLAM Framework for Event-Based Vision. *IEEE Access* **2023**, *11*, 58436–58450. [CrossRef]
7. Shiba, S.; Aoki, Y.; Gallego, G. Fast Event-Based Optical Flow Estimation by Triplet Matching. *IEEE Signal Process. Lett.* **2022**, *29*, 2712–2716. [CrossRef]
8. Iaboni, C.; Patel, H.; Lobo, D.; Choi, J.W.; Abichandani, P. Event Camera Based Real-Time Detection and Tracking of Indoor Ground Robots. *IEEE Access* **2021**, *9*, 166588–166602. [CrossRef]
9. Kang, D.; Kang, D. Event Camera-Based Pupil Localization: Facilitating Training with Event-Style Translation of RGB Faces. *IEEE Access* **2023**, *11*, 142304–142316. [CrossRef]
10. Iddrisu, K.; Shariff, W.; Corcoran, P.; O'Connor, N.E.; Lemley, J.; Little, S. Event Camera-Based Eye Motion Analysis: A Survey. *IEEE Access* **2024**, *12*, 136783–136804. [CrossRef]
11. Kang, D.; Lee, Y.K.; Jeong, J. Exploring the Potential of Event Camera Imaging for Advancing Remote Pupil-Tracking Techniques. *Appl. Sci.* **2023**, *13*, 10357. [CrossRef]
12. Messikommer, N.; Gehrig, D.; Loquercio, A.; Scaramuzza, D. Event-Based Asynchronous Sparse Convolutional Networks. In *European Conference on Computer Vision—ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer: Cham, Switzerland, 2020; pp. 415–431.
13. Schaefer, S.; Gehrig, D.; Scaramuzza, D. AEGNN: Asynchronous Event-Based Graph Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; Computer Vision Foundation: New York, NY, USA , 2022; pp. 12371–12381.
14. Lenz, G.; Ieng, S.H.; Benosman, R. Event-Based Face Detection and Tracking Using the Dynamics of Eye Blinks. *Front. Neurosci.* **2020**, *14*, 587. [CrossRef] [PubMed]

15. Ryan, C.; O'Sullivan, B.; Elrasad, A.; Cahill, A.; Lemley, J.; Kielty, P.; Posch, C.; Perot, E. Real-time face & eye tracking and blink detection using event cameras. *Neural Netw.* **2021**, *141*, 87–97. [CrossRef] [PubMed]
16. Chen, G.; Hong, L.; Dong, J.; Liu, P.; Conradt, J.; Knoll, A. EDDD: Event-Based Drowsiness Driving Detection Through Facial Motion Analysis with Neuromorphic Vision Sensor. *IEEE Sens. J.* **2020**, *20*, 6170–6181. [CrossRef]
17. Berlincioni, L.; Cultrera, L.; Albisani, C.; Cresti, L.; Leonardo, A.; Picchioni, S.; Becattini, F.; Del Bimbo, A. Neuromorphic Event-Based Facial Expression Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*; Computer Vision Foundation: New York, NY, USA, 2023; pp. 4109–4119.
18. Rebecq, H.; Gehrig, D.; Scaramuzza, D. ESIM: An Open Event Camera Simulator. In *Proceedings of the 2nd Conference on Robot Learning*; Billard, A., Dragan, A., Peters, J., Morimoto, J., Eds.; Proceedings of Machine Learning Research: Cambridge, MA, USA, 2018; Volume 87, pp. 969–982.
19. Deng, J.; Guo, J.; Ververas, E.; Kotsia, I.; Zafeiriou, S. RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Computer Vision Foundation: New York, NY, USA, 2020.
20. Orchard, G.; Jayawant, A.; Cohen, G.K.; Thakor, N. Converting static image datasets to spiking neuromorphic datasets using saccades. *Front. Neurosci.* **2015**, *9*, 437. [CrossRef]
21. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*; IEEE: Piscataway, NJ, USA, 2001; Volume 1, p. I. [CrossRef]
22. Viola, P.; Jones, M.J. Robust real-time face detection. *Int. J. Comput. Vis.* **2004**, *57*, 137–154. [CrossRef]
23. Aziz, L.; Haji Salam, M.S.B.; Sheikh, U.U.; Ayub, S. Exploring Deep Learning-Based Architecture, Strategies, Applications and Current Trends in Generic Object Detection: A Comprehensive Review. *IEEE Access* **2020**, *8*, 170461–170495. [CrossRef]
24. Jiao, L.; Zhang, F.; Liu, F.; Yang, S.; Li, L.; Feng, Z.; Qu, R. A Survey of Deep Learning-Based Object Detection. *IEEE Access* **2019**, *7*, 128837–128868. [CrossRef]
25. Kang, J.; Tariq, S.; Oh, H.; Woo, S.S. A Survey of Deep Learning-Based Object Detection Methods and Datasets for Overhead Imagery. *IEEE Access* **2022**, *10*, 20118–20134. [CrossRef]
26. Lu, P.J.; Chuang, J.H. Fusion of Multi-Intensity Image for Deep Learning-Based Human and Face Detection. *IEEE Access* **2022**, *10*, 8816–8823. [CrossRef]
27. Ross, T.Y.; Dollár, G. Focal loss for dense object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; Computer Vision Foundation: New York, NY, USA, 2017; pp. 2980–2988.
28. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
29. Chi, C.; Zhang, S.; Xing, J.; Lei, Z.; Li, S.Z.; Zou, X. Selective refinement network for high performance face detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*; Association for the Advancement of Artificial Intelligence: Washington, DC, USA, 2019; Volume 33, pp. 8231–8238.
30. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In *Computer Vision—ECCV 2016: Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Proceedings, Part I ; Springer: Cham, Switzerland, 2016; pp. 21–37.
31. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; Computer Vision Foundation: New York, NY, USA, 2017; pp. 2117–2125.
32. Cortes, C. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
33. Li, F.-F.; Fergus, R.; Perona, P. Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop*; Computer Vision Foundation: New York, NY, USA, 2004; p. 178. [CrossRef]
34. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features From Tiny Images. 2009. Available online: https://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf (accessed on 8 February 2026).
35. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
36. Li, H.; Liu, H.; Ji, X.; Li, G.; Shi, L. Cifar10-dvs: An event-stream dataset for object classification. *Front. Neurosci.* **2017**, *11*, 309. [CrossRef]
37. Hu, Y.; Liu, S.C.; Delbruck, T. v2e: From Video Frames to Realistic DVS Events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*; Computer Vision Foundation: New York, NY, USA, 2021; pp. 1312–1321.

38. Gehrig, D.; Gehrig, M.; Hidalgo-Carrio, J.; Scaramuzza, D. Video to Events: Recycling Video Datasets for Event Cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; Computer Vision Foundation: New York, NY, USA, 2020.
39. Fan, W.; Chen, J.; Ma, J.; Hou, J.; Yi, S. StyleFlow For Content-Fixed Image to Image Translation. *arXiv* **2022**, arXiv:2207.01909. [CrossRef]
40. Hore, A.; Ziou, D. Image quality metrics: PSNR vs. SSIM. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2010; pp. 2366–2369.
41. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6629–6640
42. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; Computer Vision Foundation: New York, NY, USA, 2018; pp. 586–595.
43. Lu, M.; Xu, F.; Zhao, H.; Yao, A.; Chen, Y.; Zhang, L. Exemplar-Based Portrait Style Transfer. *IEEE Access* **2018**, *6*, 58532–58542. [CrossRef]
44. Hertz, A.; Voynov, A.; Fruchter, S.; Cohen-Or, D. Style Aligned Image Generation via Shared Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; Computer Vision Foundation: New York, NY, USA, 2024; pp. 4775–4785.
45. Karras, T.; Aittala, M.; Hellsten, J.; Laine, S.; Lehtinen, J.; Aila, T. Training generative adversarial networks with limited data. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 12104–12114.
46. Lucic, M.; Kurach, K.; Michalski, M.; Gelly, S.; Bousquet, O. Are gans created equal? A large-scale study. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 698–707.
47. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
48. Wright, M.; Ommer, B. Artfid: Quantitative evaluation of neural style transfer. In *DAGM German Conference on Pattern Recognition*; Springer: Cham, Switzerland, 2022; pp. 560–576.
49. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep Learning Face Attributes in the Wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*; Computer Vision Foundation: New York, NY, USA, 2015.
50. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861. [CrossRef]
51. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; Computer Vision Foundation: New York, NY, USA, 2016; pp. 770–778.
52. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. Imagenet: A large-scale hierarchical image database. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2009; pp. 248–255.
53. Taverni, G.; Paul Moeys, D.; Li, C.; Cavaco, C.; Motsnyi, V.; San Segundo Bello, D.; Delbruck, T. Front and Back Illuminated Dynamic and Active Pixel Vision Sensors Comparison. *IEEE Trans. Circuits Syst. II Express Briefs* **2018**, *65*, 677–681. [CrossRef]
54. Posch, C.; Matolin, D.; Wohlgenannt, R. A QVGA 143 dB Dynamic Range Frame-Free PWM Image Sensor With Lossless Pixel-Level Video Compression and Time-Domain CDS. *IEEE J. Solid-State Circuits* **2011**, *46*, 259–275. [CrossRef]
55. Zheng, H.; Wu, Y.; Deng, L.; Hu, Y.; Li, G. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*; Association for the Advancement of Artificial Intelligence: Washington, DC, USA, 2021; Volume 35, pp. 11062–11070. [CrossRef]
56. Fang, W.; Yu, Z.; Chen, Y.; Masquelier, T.; Huang, T.; Tian, Y. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; Computer Vision Foundation: New York, NY, USA, 2021; pp. 2661–2671.
57. Zhou, Z.; Zhu, Y.; He, C.; Wang, Y.; Yan, S.; Tian, Y.; Yuan, L. Spikformer: When Spiking Neural Network Meets Transformer. In Proceedings of the Eleventh International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
58. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 3–7 May 2021.
59. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.