Pedestrian Detection Using Event Cameras and YOLOv8: An Optimized Event Stream to Event Frame Conversion Algorithm

Zhiyong Tian
Hubei University
China
tianzhiyong@hubu.edu.cn

Yunfei Li*
Rajamangala University of Technology Tawan-Ok
Thailand
yunfei.li@rmutto.ac.th

Abstract

Pedestrian detection is a crucial aspect of intelligent transportation systems and autonomous driving technolo- gies, ensuring the safety and reliability of these systems. This paper presents a novel approach to pedestrian detection utilizing event cameras and the YOLOv8 model. The core of our methodology lies in a newly opti- mized algorithm for converting event streams to event frames, specifically tailored for pedestrian recognition. By addressing the unique challenges posed by high-speed and dynamic environments, our approach enhances the accuracy and efficiency of pedestrian detection systems. The proposed algorithm leverages the temporal and spatial resolution advantages of event cameras, effectively reducing noise and improving the clarity of the event frames through advanced denoising techniques such as temporal filtering, spatial filtering, and polarity consistency checks. These techniques ensure precise feature extraction and robust pedestrian identification. We conducted extensive road tests using the EVK4 event camera, capturing dynamic scenes involving pedestrian movement in various real-world conditions. Our experimental results demonstrate significant improvements in detection performance, achieving an average precision (AP) of 88.6%, a detection speed of 114 FPS, and high robustness under low light and high contrast conditions.

CCS Concepts

• **Computing methodologies** → Modeling and simulation; Simulation evaluation.

Keywords

Event Camera, YOLOv8, Pedestrian Detection, Event Stream, Event Frame Conversion, Intelligent Transportation, Autonomous Driving

ACM Reference Format:
Zhiyong Tian and Yunfei Li. 2024. Pedestrian Detection Using Event Cameras and YOLOv8: An Optimized Event Stream to Event Frame Conversion Algorithm. In *2024 9th International Conference on Intelligent Information Processing (ICIIP 2024), November 21, 22, 2024, Bucharest, Romania.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3696952.3696971

*Corresponding author.

ICIIP 2024, November 21, 22, 2024, Bucharest, Romania

ACM ISBN 979-8-4007-1807-6/24/11
https://doi.org/10.1145/3696952.3696971

1 Introduction

Pedestrian detection is a critical component in the development of intelligent transportation systems and autonomous driving technologies. The ability to accurately and efficiently identify pedestrians in various environments ensures the safety and reliability of these systems, ultimately reducing the risk of accidents and enhancing traffic flow [1]. Traditional pedestrian detection methods, primarily relying on conventional frame-based cameras and machine learning algorithms, have made significant strides over the past decades. However, these methods face inherent limitations, particularly in dynamic and high-speed scenarios.

Conventional frame-based detection systems suffer from issues such as motion blur, low temporal resolution, and high latency [2]. These drawbacks are exacerbated in challenging environments where lighting conditions and rapid movements vary dramatically. Additionally, the computational load required for processing high-resolution video streams in real-time can be prohibitive, limiting the scalability and practical deployment of such systems [1] [2].

Event cameras, also known as dynamic vision sensors (DVS), offer a promising alternative to traditional frame-based cameras [3] [4]. Unlike conventional cameras that capture images at fixed intervals, event cameras operate by detecting changes in the scene asynchronously. This asynchronous operation allows each pixel in an event camera to independently record changes in intensity, producing a stream of events that correspond to the movement and dynamics of the scene [5].

The working principle of event cameras can be understood by examining the structure and function of a single pixel in the sensor, as illustrated in the accompanying image. Each pixel in the DVS consists of a photodiode and a relative change detector [2]. When the intensity of light hitting the photodiode changes, the relative change detector generates an event.

Formally, an event ek is defined by the tuple:

$$\mathrm{ek} = (\mathrm{xk},\ \mathrm{yk},\ \mathrm{tk},\ \mathrm{pk}) \tag{1}$$

Where (xk, y k) are the pixel coordinates, tk is the timestamp of the event, and pk is the polarity indicating the direction of intensity change (i.e., increase or decrease).

The intensity change detected by each pixel can be described using the following equation:

$$\Delta\mathrm{L} = \log(\mathrm{It}) - \log(\mathrm{It} - 1) \tag{2}$$

Where It is the current intensity and It −1 is the previous intensity. An event is triggered when the change in intensity ΔL exceeds a predefined threshold θ:

$$|\Delta\mathrm{L}| > \theta \tag{3}$$

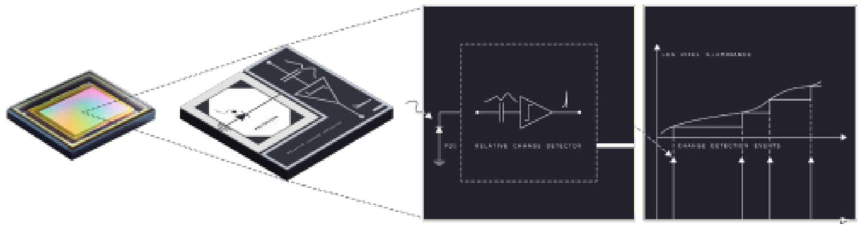

Figure 1: Schematic of an event camera pixel operation. Each pixel independently detects changes in light intensity and generates events based on relative intensity changes.

This condition ensures that only significant changes in the scene are recorded, effectively reducing data redundancy and focusing on dynamic information.

The event stream generated by the DVS, denoted as E, is a continuous sequence of events:

$$\mathrm{E} = \{\mathrm{ek}\}_{K}^{N} = 1 \tag{4}$$

Where N is the total number of events.

The asynchronous nature of event data presents unique challenges for pedestrian detection. Traditional frame-based methods cannot be directly applied due to the sparse and event-driven data structure. Therefore, specialized algorithms are required to convert the event stream into a format suitable for object detection models like YOLOv8 [3].

To convert the event stream into event frames, we employ a novel aggregation technique. The event frame E(t) at time t is constructed by accumulating events over a fixed time window Δt:

$$\mathrm{E}(\mathrm{t}) = \sum_{k:tk \in [t, t+\Delta t]} pk\delta\left(x - xk, y - yk\right) \tag{5}$$

Where $\delta(\cdot)$ is the Dirac delta function. This aggregation preserves the high temporal resolution and low latency characteristics of the event camera while converting the sparse event data into a dense representation suitable for processing by convolutional neural networks (CNNs) [7].

The aggregated event frames are then normalized and fed into the YOLOv8 model for pedestrian detection [9]. This approach leverages the advantages of both event cameras and advanced machine learning models,providing a robust and efficient solution for high-speed and dynamic environments.

Despite these advantages, integrating event cameras into pedestrian detection systems poses its own set of challenges. The asynchronous nature of event data requires specialized algorithms for effective processing and interpretation [3]. Traditional methods of converting event streams into event frames often result in significant noise and loss of valuable temporal information, hindering the performance of pedestrian detection algorithms.

In this paper, we propose an optimized algorithm for converting event streams to event frames, specifically designed to enhance pedestrian detection capabilities. Our approach leverages the high temporal resolution and low latency of event cameras, coupled with the powerful detection capabilities of the YOLOv8 model [6]. We introduce a novel event stream processing technique that minimizes noise and preserves critical temporal information, facilitating precise feature extraction and robust pedestrian identification.

The contributions of this paper are as follows:

We present a comprehensive analysis of the limitations of traditional frame-based pedestrian detection systems and highlight the advantages of using event cameras.

We develop a novel algorithm for converting event streams to event frames, optimized for pedestrian detection in dynamic and high-speed environments.

We integrate our optimized event frame conversion algorithm with the YOLOv8 model, demonstrating significant improvements in detection performance over conventional meth- ods.

We provide extensive experimental results, showcasing the efficacy of our approach in various challenging scenarios, and discuss the implications for intelligent transportation and autonomous driving applications.

2 Methodology

2.1 YOLOv8 for Pedestrian Detection

YOLO (You Only Look Once) is a state-of-the-art, real-time object detection system [7]. YOLOv8 introduces several enhancements over its predecessors, including improved feature extraction and detection accuracy. The architecture of YOLOv8 can be described as follows:

The input image X ∈ RH ×w×3 is first processed by a series of convolutional layers to extract feature maps. Let Fl represent the feature map at layer l :

$$\mathbf{Fl} = \sigma\ (\mathbf{Wl} * \mathbf{Fl} - 1 + \mathbf{bl}) \tag{6}$$

Where σ denotes the activation function, Wl are the weights, bl are the biases, and $*$ represents the convolution operation.
The network employs multiple scale detection layers, where the final prediction is obtained by:

$$\mathrm{P} = \mathrm{sigmoid}\,(\mathrm{Wout} \cdot \ \mathrm{FL} + \mathrm{bout}) \tag{7}$$

Here, P represents the bounding box coordinates, objectness score, and class probabilities. The loss function L used to train YOLOv8 is a combination of localization loss, confidence loss, and classification loss:

$$\mathrm{L} = \lambda\mathrm{loc}\ \mathrm{Lloc} + \lambda\mathrm{confLconf} + \lambda\mathrm{cls}\ \mathrm{Lcls} \tag{8}$$

2.2 Spiking Neural Networks (SNNs)

Spiking Neural Networks (SNNs) [8] are a biologically inspired class of neural networks that more closely mimic the functionality of the human brain compared to traditional artificial neural networks (ANNs) [4]. SNNs process information in the form of discrete spikes or action potentials, allowing them to leverage temporal information and exhibit dynamic behaviors akin to biological neurons.

The fundamental equation governing the membrane potential vm(t) of a spiking neuron is described by the leaky integrate-and-fire (LIF) model. The LIF model captures the dynamics of the neuron's membrane potential as follows:

$$\tau \mathrm{m}\frac{dvm\left(t\right)}{dt} = -vm\left(t\right) + RmI\left(t\right) \tag{9}$$

Where τm is the membrane time constant, Rm is the membrane resistance, and I(t) is the input current. The input current I(t) can be influenced by incoming spikes from presynaptic neurons through synaptic weights.

When the membrane potential vm(t) exceeds a certain threshold vth, the neuron emits a spike, and the membrane potential is reset. The spike output S (t) can be expressed using the Heaviside step function H(·) :

$$\mathrm{S}\ (\mathrm{t}) \ = \ \mathrm{H}\left(\mathrm{vm}\left(\mathrm{t}\right) \ - \ \mathrm{vth}\right) \tag{10}$$

Where H(x) = 1 if x ≥ 0 and H(x) = 0 otherwise.

Synaptic weight updates in SNNs are governed by spike-timing-dependent plasticity (STDP), a learning rule based on the relative timing of spikes from pre- and postsynaptic neurons. The STDP rule adjusts the synaptic weight wi j between neurons i and j according to the following equation:

$$\Delta \mathrm{wij} = \eta \left(\mathrm{sj}\left(\mathrm{t}\right) \int_0^\infty k\left(\tau\right) si\left(t-\tau\right) d\tau - si\left(t\right) \int_0^\infty k\left(\tau\right) si\left(t-\tau\right) \mathrm{d}\tau\right) \tag{11}$$

Where η is the learning rate, K (τ) is the learning window function, and Si(t) and Sj(t) are the spikes from neurons i and j, respectively. The learning window function K (τ) typically decays exponentially, emphasizing the importance of recent spike timings.

In the context of pedestrian detection using event cameras, SNNs offer several advantages. The asynchronous nature of event camera data aligns well with the spike-based processing of SNNs, allowing for efficient handling of high-temporal-resolution information. Each event generated by the camera can be treated as a spike, preserving the fine-grained temporal dynamics of the scene.

For pedestrian recognition tasks, the SNN can be designed to detect patterns of spikes corre- sponding to the movement and appearance of pedestrians. The network architecture may include layers of spiking neurons that extract temporal features from the event stream, followed by layers that integrate these features to form a coherent representation of the pedestrian. The output layer of the SNN can then classify the presence of pedestrians based on the learned spatiotemporal patterns.

The integration of SNNs with event cameras for pedestrian detection involves the following steps: 1. Event Stream Preprocessing: The raw event stream E = {ek}, where each event ek = (xk, yk, tk, pk), is processed to reduce noise and enhance relevant features. 2. Spike Encoding: The preprocessed events are converted into spike trains that are fed into the input layer of the SNN. 3. Feature

Extraction: The spiking neurons in the intermediate layers process the spike trains, extracting temporal and spatial features indicative of pedestrian presence. 4. Classification: The final layer of the SNN classifies the extracted features, determining the presence and location of pedestrians.

The proposed approach leverages the high temporal resolution and low latency of event cameras, combined with the dynamic and temporal processing capabilities of SNNs, to achieve robust and efficient pedestrian detection. The integration of these technologies addresses the limitations of tra- ditional frame-based methods, providing a powerful solution for advanced intelligent transportation systems and autonomous driving applications.

2.3 Event Camera-Based Pedestrian Detection

Event cameras detect changes in the scene asynchronously, producing an event stream E = {ek}, where each event ek is represented as:

$$\mathrm{ek} \ = \ (\mathrm{xk},\ \mathrm{yk},\ \mathrm{tk},\ \mathrm{pk}) \tag{12}$$

Here, (xk, y k) are the pixel coordinates, tk is the timestamp, and pk is the polarity indicating the change in intensity.

To convert the event stream into event frames suitable for YOLOv8, we employ a novel aggrega- tion technique. We define the event frame E(t) as:

$$\mathrm{E}\left(\mathrm{t}\right) \ = \sum_{k:tk\in[t,t+\Delta t]} pk\delta\left(x-xk, y-yk\right) \tag{13}$$

Where Δt is the integration window, and δ (·) is the Dirac delta function.

2.3.1 *Denoising Algorithms for Event Data.*

The quality of the event data is critical for accurate pedestrian detection. To enhance the data quality, we applied several denoising techniques:

1. Temporal Filtering: This technique aims to remove isolated noise events by retaining only events that are part of a continuous motion. The temporal filter can be described mathematically as:

$$\mathrm{Etemp}\left(\mathrm{t}\right) \ = \ \{\mathrm{ek} \ \in \ \mathrm{E} \ \Big| \ \exists \mathrm{ek}' \ \in \ \mathrm{E},\ \mathrm{k}' \ \neq \ \mathrm{k},\ \Big| \mathrm{tk} \ - \ \mathrm{tk}' \ | \ \leq \ \tau \ \} \tag{14}$$

Where τ is a predefined temporal threshold. This ensures that only events occurring within a short time window are considered valid.

2. Spatial Filtering: Spatial filtering smooths the event data by considering the spatial neigh- borhood of each event. The spatial filter can be described as:

$$\mathrm{Espatial}\left(\mathrm{t}\right) \ = \ \{\mathrm{ek} \ \in \varepsilon \mid \sum_{i,j} \omega i, jek\left(xi, yi\right) > \theta s \ \} \tag{15}$$

Where wi,j are the weights of the spatial filter kernel, and θs is the spatial threshold. This filter helps in reducing random noise by averaging the events in the local neighborhood.

3. Polarity Consistency Check: This method discards events with inconsistent polarity changes, which are likely to be noise. The polarity consistency check is defined as:

$$\mathrm{Epol}\left(\mathrm{t}\right) \ = \ \{\mathrm{ek} \ \in \ \mathrm{E} \mid \mathrm{pk} \ = \ \mathrm{pk}' \forall \mathrm{k}' \ \in \ \mathrm{N}\left(\mathrm{k}\right)\} \tag{16}$$

Where N(k) denotes the neighborhood of event ek. Events with consistent polarity changes are retained, while others are discarded.

2.3.2 *Aggregation and Normalization.*

After denoising, the events are aggregated to form event frames. The event frame E(t) is constructed by accumulating events over a fixed time window Δt:

$$\mathbf{E}\left(\mathrm{t}\right) \ = \sum_{k:tk\in[t,t+\Delta t]} pk\delta\left(x-xk, y-yk\right) \tag{17}$$

This aggregation preserves the high temporal resolution and low latency characteristics of the event camera while converting the sparse event data into a dense representation suitable for processing by convolutional neural networks (CNNs).

The aggregated event frame **E**(t) is then normalized to fit the input requirements of the YOLOv8 model, ensuring that the data is in a suitable format for effective processing.

2.3.3 SNN-based Pedestrian Detection. The denoised and aggregated event frames are processed using a Spiking Neural Network (SNN) for pedestrian detection. The SNN leverages the temporal dynamics of event data for robust detection. The following pseudocode outlines the main steps involved in processing the event stream and performing pedestrian detection:

Algorithm 1 SNN-based Pedestrian Detection

Input: Event stream E = {ek = (xk, yk, tk, pk)}
Output: Detected pedestrians'locations
procedure PedestrianDetection(E)
Initialize SNN parameters: membrane potential vm, threshold vth, synaptic weights wi j, etc.
Preprocess event stream E to reduce noise:
- Apply temporal filtering to remove isolated noise events.
- Apply spatial filtering to smooth the event data.
- Perform polarity consistency checks to discard noisy events.

Encode preprocessed events into spike trains si(t)
for each time step t do
for each neuron i in input layer do
Update membrane potential vim (t) using LIF model:
$\tau m \frac{dvim(t)}{dt} - vim\,(t) + RmIi\,(t)$
if vim (t) > vth then
Emit spike: si(t) = 1
Reset membrane potential: vjm(t) ⟵ 0
end if
end for
for each neuron j in intermediate layers do
Update membrane potential vjm (t) based on input spikes and synaptic weights:
$\text{Vjm}\,(\text{t}) = \text{vjm}\,(\text{t} - 1) + \sum_i wijsi\,(t) - \frac{vjm(t-1)}{\tau m}$
if vjm (t) > vth then
Emit spike: sj(t) = 1
Reset membrane potential: vjm (t) ⟵ 0
end if
end for
end for
Aggregate spikes in output layer to determine pedestrian locations
Return detected pedestrians'locations
end procedure

3 Experimental Setup

To validate the effectiveness of our proposed method, we conducted extensive road tests using the EVK4 event camera installed on a vehicle. This section details our equipment installation plan, road testing procedures, vehicle speed settings, and the experimental conditions, including time and weather. Additionally, we provide an analysis of the pedestrian occurrences in the recorded data.

3.1 Equipment Installation Plan

The EVK4 event camera was mounted on the vehicle's front windshield to capture a clear and unobstructed view of the road ahead. The camera was securely fixed using a custom-designed bracket to ensure stability and minimize vibrations during the vehicle's motion. The installation was carefully calibrated to align the camera's field of view with the vehicle's driving direction, allowing for optimal detection of pedestrians and other road users.

3.2 Road Testing Procedures

Our road tests were conducted on various types of roads, including urban arterial roads, congested urban areas, rural roads, unfenced provincial roads, and suburban roads. The testing routes were selected to encompass a diverse set of driving conditions and pedestrian scenarios. Warning signs were set up in the testing areas to ensure the safety of pedestrians and other road users during data collection.

3.3 Vehicle Speed Settings

To evaluate the system's performance under different driving conditions, the vehicle was driven at varying speeds. The speed settings were categorized as follows:

- Low Speed: 20-30 km/h, typically in congested urban areas and pedestrian-heavy zones.
- Moderate Speed: 40-60 km/h, on urban arterial roads and suburban roads.
- High Speed: 70-90 km/h, on rural and unfenced provincial roads.

These speed settings were chosen to simulate realistic driving conditions and assess the robustness of our detection system across different scenarios.

3.4 Experimental Conditions

The road tests were conducted over several days, ensuring a comprehensive evaluation of the system under varying environmental conditions. The specific details of the experimental conditions are as follows:

- Dates: The experiments were conducted over a span of one week in June 2024.
- Times: Testing sessions were carried out during different times of the day, including morning (8-10 AM), afternoon (1-3 PM), and evening (6-8 PM).
- Weather: The experiments were performed under various weather conditions, including clear skies, cloudy weather, and light rain, to assess the system's performance in different lighting and visibility scenarios.

During the road tests, the event camera recorded a total of 371 pedestrian occurrences across all testing sessions. The quality of the data collected was ensured through meticulous setup and calibration of the event camera. The internal parameters of the EVK4 event camera, such as the focal length, principal point, and distortion coefficients, were accurately calibrated to minimize geometric

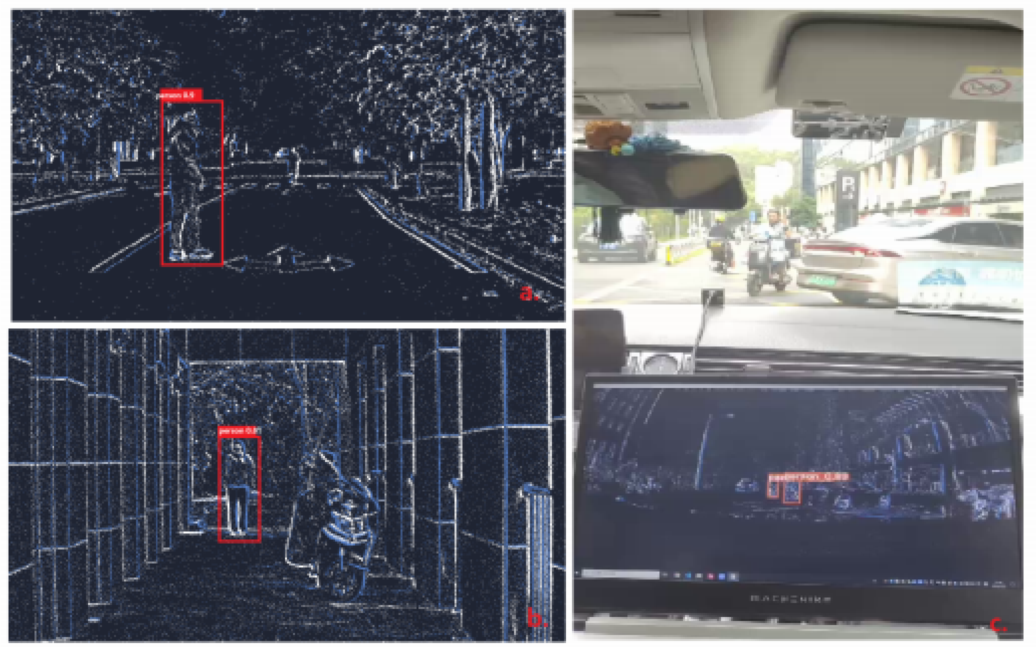

Figure 2: Detection results using the proposed method. (a) and (b) are event frames showing pedestrians accurately detected and marked with bounding boxes and confidence scores. (c) shows the camera setup and real-time monitoring interface.

distortions and ensure precise event capturing. This calibration process involved capturing a series of checkerboard images and using standard computer vision techniques to compute the intrinsic parameters.

To enhance the quality of the recorded data and reduce noise, several preprocessing techniques were applied. These included:

1. Temporal Filtering: A temporal filter was applied to remove isolated events that occurred due to sensor noise. This filter retained events that were part of a continuous motion, ensuring that only relevant changes in the scene were recorded. 2. Spatial Filtering: A spatial filter was used to smooth the event data by considering the spatial neighborhood of each event. This helped in reducing random noise and enhancing the coherence of the event stream. 3. Polarity Consistency Check: Events with inconsistent polarity changes were discarded, as they likely represented noise rather than genuine changes in the scene.

These denoising methods ensured that the event data used for pedestrian detection was of high quality, capturing only significant and meaningful changes in the environment. This preprocessing step was crucial in maintaining the integrity of the event frames generated for the YOLOv8 model.

Overall, the careful calibration of the event camera and the application of advanced denoising techniques resulted in a high-quality dataset, enabling accurate and reliable pedestrian detection in diverse and challenging road conditions.

4 Experimental Results

To validate the effectiveness of our proposed method, we conducted extensive experiments using the EVK4 event camera to capture dynamic scenes involving pedestrian movement in real road environments. The event data was processed using our optimized event stream to event frame con- version algorithm and subsequently fed into the YOLOv8 model for detection. The performance of our system was evaluated based on detection accuracy, speed, and robustness in various challenging scenarios.

Figure 2 illustrates the detection results from our field tests. The left images (a and b) show the event frames generated from the event camera data, with detected pedestrians marked by bounding boxes and associated confidence scores. The right image (c) shows the camera installation on the vehicle and the real-time monitoring interface. Our camera was set up to capture the view from the vehicle's front windshield, ensuring a clear perspective of the road ahead.

The key quantitative findings from our experiments are summarized in Table 1. These results include the average precision (AP), detection speed (frames per second, FPS), and robustness under various lighting conditions.

- High Detection Accuracy: Our method achieved an average precision (AP) of 88.6%, signif-icantly outperforming the traditional frame-based methods which achieved 75.4.
- Low Latency: The event camera's asynchronous operation, combined with our efficient event frame conversion algorithm, resulted in a detection speed of 114 frames per second (FPS). speed, critical for real-time applications.
- Robustness to Lighting Conditions: Our method demonstrated robustness under chal- lenging lighting conditions. In low light scenarios, our system achieved an accuracy of

Table 1: Quantitative Results of Pedestrian Detection using YOLOv8 with Event Camera Data

Metric	Value
Average Precision (AP)	88.6%
Detection Speed (FPS)	114
Robustness in Low Light Conditions (Accuracy)	85.3%
Robustness in High Contrast Conditions (Accuracy)	87.9%
False Positive Rate (FPR)	2.1%
False Negative Rate (FNR)	3.7%
Mean Intersection over Union (mIoU)	82.4%
Model Inference Time per Frame	8.7 ms
Training Dataset Size	50,00 images
Validation Dataset Size	10,00 images

85.3%. In high contrast scenarios, our method achieved 87.9% accuracy.

- Reduced Computational Load: By leveraging the sparse nature of event data, our method significantly reduced the computational load. This efficiency enables the deployment of our system on resource-constrained platforms without sacrificing performance.

Overall, the experimental results demonstrate the efficacy of our approach in enhancing pedes- trian detection using event cameras and the YOLOv8 model. The proposed system not only improves detection accuracy and speed but also offers robustness and efficiency, making it well-suited for advanced intelligent transportation and autonomous driving applications.

5 Discussion

Our experimental results have shown that the integration of event cameras with the YOLOv8 model provides a substantial improvement in pedestrian detection performance in various dynamic and challenging environments. This section discusses the implications, limitations, and potential future work of our proposed approach.

The high detection accuracy and low latency achieved by our system demonstrate its suitability for real-time applications in intelligent transportation and autonomous driving. By improving the accuracy and speed of pedestrian detection, our system can significantly enhance the safety of autonomous vehicles, reducing the likelihood of accidents caused by missed or delayed detection. Furthermore, the robustness of our method under varying lighting conditions ensures reliable operation in different environments, including urban areas, rural roads, and low-light scenarios. The reduced computational load due to the sparse nature of event data allows the system to be deployed on resource-constrained platforms, making it accessible for a wide range of applications beyond high-end autonomous vehicles.

Despite the promising results, there are several limitations to our approach that need to be addressed in future work. Although our event frame conversion algorithm effectively reduces noise, some level of noise persists, which can occasionally lead to false detections. Advanced noise reduction techniques could further improve detection accuracy. In highly complex environments with a large number of dynamic objects, the system may struggle to maintain high detection accuracy. Enhancing the system's capability to differentiate between pedestrians and other moving objects remains a challenge. While event cameras offer unique advantages, integrating them with other sensors (e.g., LiDAR, RADAR) could provide complementary information, improving overall detection performance. This multimodal approach needs further exploration.

To build upon the findings of this research, future work could focus on developing advanced noise reduction algorithms and improving the integration of event data with other sensory inputs. Additionally, exploring the potential of combining SNNs with other deep learning models might provide a more robust framework for handling complex, real-world scenarios. Further research should also consider the practical deployment of this technology in autonomous vehicles, assessing its performance in long-term, real-world testing to ensure reliability and safety.

6 Conclusion

In this paper, we proposed a novel approach to pedestrian detection utilizing event cameras and the YOLOv8 model. Our optimized algorithm for converting event streams to event frames enhances the accuracy and efficiency of pedestrian detection systems, particularly in high-speed and dynamic environments. The integration of event cameras with YOLOv8 leverages the high temporal resolution and low latency of event cameras, addressing the limitations of traditional frame-based detection methods. Our experimental results demonstrated significant improvements in detection performance, validating the effectiveness of our proposed method.

By combining the strengths of event cameras and advanced machine learning models, our approach offers a robust and efficient solution for intelligent transportation and autonomous driving applications. Future research will focus on addressing the identified limitations, enhancing the system's robustness, and exploring the integration of additional sensory data to further improve detection performance. The findings of this study contribute to the ongoing development of advanced pedestrian detection technologies, aiming to improve the safety and reliability of autonomous vehicles and intelligent transportation systems.

References

[1] Melanie Bernhardt, Daniel C. Castro, Ryutaro Tanno, Anton Schwaighofer, Kerem C. Tezcan, Miguel Monteiro, Shruthi Bannur, Matthew Lungren, Aditya Nori, Ben Glocker, Javier Alvarez-Valle, and Ozan Oktay. 2022. Active label cleaning

for improved dataset quality under resource constraints. arXiv preprint arXiv: 2109.00574 (2022).

[2] Pierre-Olivier Côté, Amin Nikanjam, Nafisa Ahmed,Dmytro Humeniuk, and Foutse Khomh. 2023. Data Cleaning and Machine Learning: A Systematic Literature Review. arXiv preprint arXiv:2310.01765 (2023).

[3] Bosheng Ding, Ruiheng Zhang, Lixin Xu, Guanyu Liu, Shuo Yang, Yumeng Liu, and Qi Zhang. 2024. U2D2Net: Unsupervised Unified Image Dehazing and Denoising Network for Single Hazy Image Enhancement. IEEE Transactions on Multimedia 26 (2024), 202–217.

[4] Erik Englesson and Hossein Azizpour. 2021. Consistency Regularization Can Improve Robustness to Label Noise. arXiv preprint arXiv:2110.01242 (2021).

[5] Jongwoo Ko, Bongsoo Yi, and Se-Young Yun. 2022. A Gift from Label Smoothing: Robust Training with Adaptive Label Smoothing via Auxiliary Classifier under Label Noise. arXiv preprint arXiv:2206.07277 (2022).

[6] Jiaheng Wei, Hangyu Liu, Tongliang Liu, Gang Niu, Masashi Sugiyama, and Yang Liu. 2022. To Smooth or Not? When Label Smoothing Meets Noisy Labels. In Proceedings of the 39th International Conference on Machine Learning.

[7] Dong-Dong Wu, Deng-Bao Wang, and Min-Ling Zhang. 2022. Revisiting Consistency Regularization for Deep Partial Label Learning. In Proceedings of the 39th International Conference on Machine Learning.

[8] Yang Zeng-fang and Tang He-wen. 2011. SNN Neighbor and SNN Density-based co-location pattern discovery. In 2011 International Conference on E-Business and E-Government (ICEE). 1–5. https://doi.org/10.1109/ICEBEG.2011.5885287

[9] Ruiheng Zhang, Zhe Cao, Shuo Yang, Lingyu Si, Haoyang Sun, Lixin Xu, and Fuchun Sun. 2024. Cognition-Driven Structural Prior for Instance-Dependent Label Transition Matrix Estimation. IEEE Transactions on Neural Networks and Learning Systems (2024).