Intrusion Detection at Railway Tunnel Entrances Using Dynamic Vision Sensors

Colin Gebler and Regina Pohle-Fröhlich
Institute for Pattern Recognition, Niederrhein University of Applied Sciences, Krefeld, Germany

Keywords: Intrusion Detection, Transportation, Dynamic Vision Sensor.

Abstract: The surveillance of railway tunnel entrances is integral to ensure the security of both people and infrastructure. Since 24/7 personal surveillance is not economically possible, it falls to automated solutions to ensure that no persons can intrude unseen. We investigate the use of Dynamic Vision Sensors in fulfilling this task. A Dynamic Vision Sensor differs from a traditional frame-based camera in that it does not record entire images at a fixed rate. Instead, each pixel outputs events independently and asynchronously whenever a change in brightness occurs at that location. We present a dataset recorded over three months at a railway tunnel entrance, with relevant examples assigned labeled as featuring or not featuring intrusions. Furthermore, we investigate intrusion detection by using neural networks to perform image classification on images generated from the event stream using established methods to represent the temporal information in that format. Of the models tested, MobileNetV2 achieved the best result with a classification accuracy of 99.55% on our dataset when differentiating between Event Volumes that do or do not contain people.

1 INTRODUCTION

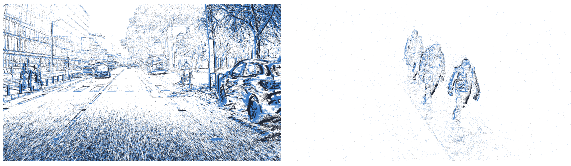
Rail-based transportation systems represent open systems in which large volumes of passengers or goods are transported every day. To ensure transportation, people must have quick and easy access to stations and trains, resulting in numerous access points (D'Amore and Tedesco, 2015). For this reason, it is necessary to ensure a high level of security at these points using automatic surveillance technology. In the numerous existing tunnels, for example, it must be ensured that no person can enter unnoticed, as this would increase the risk of accidents or disrupt rail traffic. The motives for unauthorized entry into tunnels can be very diverse, e.g. vandalism, aggression against others, tests of courage, homelessness or suicide plans. If such unauthorized entry is detected by the surveillance systems, the tunnel must be closed and manually controlled, causing delays and disruptions to train traffic.

Technical monitoring of tunnel entrances with conventional cameras is challenging due to highly variable and extreme lighting conditions such as trains approaching with their headlights turned on. In addition, rain, snow and the high speed of trains can cause turbulence, which can have very different visual appearances. On the other hand, when using Fiber Bragg Grating sensors (Catalano et al., 2017), which have relevant advantages in such environmental conditions, the classification of the triggering signal (e.g., distinguishing whether a person or an animal has stepped on the mat, for example) is a major challenge. LiDAR sensors have also been tested because they operate independently of ambient light. As a result, they provide a three-dimensional point cloud of the scanned environment. However, the dependence between the achievable spatial and temporal resolution is problematic with this technology. A high temporal resolution, such as that needed to identify fast-moving trains, can only be achieved at a low spatial resolution, so that smaller objects, such as people, have only a few scan points in the recorded LiDAR point cloud. This means that reliable identification of moving people is not possible, which can lead to false alarms, as in the case of cameras and floor mats, but also to missed detections. To avoid false alarms, intrusion detection often combines multiple technologies to maximize confidence in the results (Siraj et al., 2004).

In this paper, we investigate the feasibility of using event cameras, also called Dynamic Vision Sensors (DVS), to reduce the false alarm rate and improve the practicality of automated surveillance. Event cameras differ from conventional cameras in that they do not

Gebler, C. and Pohle-Fröhlich, R.
Intrusion Detection at Railway Tunnel Entrances Using Dynamic Vision Sensors.
DOI: 10.5220/0012558600003654

In *Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2024)*, pages 902-909
ISBN: 978-989-758-684-2; ISSN: 2184-4313



(a) Example recorded with the DVS in motion (b) Example recorded with a stationary DVS

Figure 1: Comparison of recordings with a moving or stationary DVS.

record images, but only an asynchronous event data stream in which individual changes in pixel brightness are output independently when they occur. This allows for a very high temporal resolution in the data. In addition, the sensors are very sensitive to light, so they can operate in areas with changing lighting or in very dark environments such as tunnel entrances, producing signals that are far less influenced by ambient light levels than, for example, traditional RGB cameras. This makes them particularly suitable for detecting people at tunnel entrances. To facilitate the application of event-based vision in this area we provide:

- An outdoor dataset recorded in a novel setting and auxiliary sequences with classification labels to amend the low number of DVS Datasets, especially high resolution ones, currently available.
- Results of a baseline approach to facilitating intrusion detection using image classification on generated event frames, including scenarios possible in this context which cannot be recorded directly due to concerns regarding the actor's security. These were generated by combining multiple recordings.

1.1 Related Works

A large motivator for utilizing Dynamic Vision Sensors in pedestrian detection is the automotive context. Prophesee, the manufacturer of the DVS used in recording our dataset, provides two large datasets in this area. One is the GEN1 dataset (de Tournemire et al., 2020) with manually created bounding box labels. The other is the higher resolution 1 Megapixel Automotive Detection dataset (Perot et al., 2020), which is annotated with bounding boxes extracted from a traditional frame-based RGB recording acquired in parallel. Most of the data in these datasets are recorded with the DVS in motion. This leads to immobile objects in the environment generating many events. This trait creates a significant difference between the data contained in these datasets and our use case, as showcased in Figure 1 by comparing an event frame from the 1 Megapixel Automotive Detection dataset and our dataset. Additionally, the positioning of the DVS leads to people being recorded straight on, while surveillance cameras usually record an overhead view. Empirically, applying the RED Model for bounding box detection (Perot et al., 2020) trained on the 1 Megapixel Automotive Detection Dataset to our data showed that the features learned on the former do not translate well to our use case.

(Jiang et al., 2019) explores the combination of per-pixel confidence scores calculated from DVS signals and traditional frames for bounding box pedestrian detection. This is not possible for us since we were not able to acquire corresponding frame-based data. (Miao et al., 2019) provides a small 346×260 benchmark dataset featuring 12 clips of around 30 seconds for bounding box pedestrian detection in different settings. (Bisulco et al., 2020) investigates bounding box pedestrian detection on a small non-public 480×320 dataset with a focus on bandwidth reduction. (Wan et al., 2021) provides a 488-second bounding box pedestrian detection dataset with a 346×260 pixel resolution recorded in various settings and investigates pedestrian detection on this dataset. (Alonso and Murillo, 2019; Bolten et al., 2021) provide datasets with semi-automatically generated semantic segmentation labels featuring pedestrians. (Bolten et al., 2023) provides a pedestrian dataset with instance segmentation labels generated by recording persons wearing easily differentiated suits. The lower resolutions and differing scenarios these datasets were recorded in make them difficult to apply to our use case. (Iaboni et al., 2023) provides a $70.75min$ 640×480 bounding box annotated dataset of aerial recordings in urban settings, which include pedestrians. Due to the DVS being drone mounted, this dataset exhibits similar egomotion issues as the automotive datasets described above.

In a surveillance context, human intrusion detection using Dynamic Vision Sensors is investigated in (Perez-Cutino et al., 2021), obtaining input data with a drone mounted DVS and detecting persons in two stages by first detecting moving objects and subsequently determining whether they are humans.

2 DATASET

2.1 Event-Based Vision

2.1.1 Dynamic Vision Sensor

Instead of capturing entire frames at a fixed rate, a Dynamic Vision Sensor asynchronously reports changes in light intensity for each pixel of the pixel array. For each registered change, it outputs an event $E = (x, y, t, p)$ where (x, y) are the coordinates at which the change occurred, t is the time at which the change occurred, and p indicates the direction of the change. $p = 1$ indicates that the pixel got brighter, $p = 0$ indicates that the pixel got darker. The exact values p takes can differ depending on the sensor used. These events are output in a continuous stream without a fixed frame association. Because redundant areas in which no changes occur do not generate any data, less data is transmitted than by an equivalent frame camera, especially considering the high temporal resolution in the order of microseconds.

2.1.2 Event Encoding

Applying neural networks to continuous event streams requires converting the events to a format with a fixed size. One way to achieve this is to split the event stream into temporal *bins* of a given length and generating a dense representation of the events in each bin. We choose the length of the bins $T = 50000\mu s$, resulting in 20 bins per second. The spatial information contained in the events is used to determine their position in the dense representation. There are different options for the impact of timestamp t and polarity p on the dense representation.

Linear Time Surface. The dense representation has the dimensions $H \times W \times 2$, where H is the height and W is the width of the pixel array the event stream originates from. Each channel is assigned to one of the two possible polarities. The value at each position is assigned according to the latest occurrence of an event in that position according to the formula:

$$T(x, y, p) = \frac{t_{max(x,y,p)} - t_0}{T},$$

where $t_{max(x,y,p)}$ is the timestamp of the latest event with the given polarity p in the position (x, y), t_0 is the timestamp at the beginning of the current bin and T is the length of a bin. While the sensor manufacturer's Metavision SDK (Prophesee, 2023) documentation refers to this encoding as a linear time surface[1], this approach is often referred to as a *surface of active events (SAE)* in literature (Wan et al., 2021; Mueggler et al., 2015; Benosman et al., 2014). For the rest of this paper the term *Time Surface* will refer to Linear Time Surfaces as described here when it appears.

Event Volume. An event volume (Zhu et al., 2019) represents the spatial position of each event in the first two dimensions and represents the timestamp as a combination of the third dimension and value. The third dimension further subdivides the time bin into separate micro bins which are split by event polarity, meaning that the structure essentially consists of separate event volumes for each polarity. Each event then distributes a contribution of one between the two closest microbins, so that the exact distribution of input events could be reconstructed down to a rounding error if each voxel is only contributed to by one event. We generate the event volume with six total micro bins, three for each polarity.

We perform the encoding using the Metavision (Prophesee, 2023) implementations of linear time surfaces and event volumes.

Due to the artificial lighting at the tunnel entrance, there is significantly more noise in the recordings than there is in the staged recordings taken in naturally illuminated scenes. The amount of noise also fluctuates depending on what lighting is turned on at any given time. In order to suppress this difference, we spatio-temporally filter out events with no prior events occurring in the 8-point neighborhood within the last 50ms (Delbruck, 2008).

2.2 Recording Setup

Eleven weeks of event streams were recorded at a railway tunnel entrance from February 21st 2023 to May 7th 2023. Additionally, staged material featuring pedestrians walking in front of the event camera was recorded in three separate scenes. The tunnel entrance is a restricted area. Consequently, the only instances of people appearing in the recordings are occasional authorized personnel. These instances would not provide sufficient material for training by themselves. Of the staged scenes, two were recorded on campus and one was recorded at the tunnel entrance.

All recordings were performed using *Metavision EVK3 – Gen4.1*[2] Dynamic Vision Sensors by Prophesee.

Each recording is taken from a top-down view, with the camera mounted at a height of 3 m to 4 m and a downward angle of 15°.

The bulk of the data in this dataset was recorded at a railway tunnel entrance involved in regular traffic. The DVS was mounted 8 m into the tunnel on the tunnel wall, facing slightly away from the wall and towards the tunnel entrance. The view close to the wall onto the boardwalk at the tunnel's edge is partially obstructed by a cable tray running below the DVS.

[1] https://docs.prophesee.ai/stable/tutorials/ml/data_processing/event_preprocessing.html

[2] https://www.prophesee.ai/event-based-evk-3/

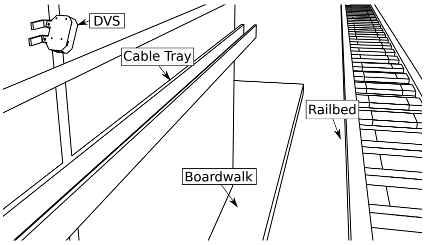


Figure 2: The setup used to acquire data at the tunnel entrance. Not to scale.

The setup is pictured in Figure 2. The DVS at the tunnel entrance was fitted with a *Tamron M117FM08-RG* objective lens. The lens' focal length of 8 mm results in a horizontal Field of View(FOV) of approximately 42° and a vertical FOV of approximately 24°.

Additional examples featuring pedestrians were recorded on campus with a DVS fitted with a *Foctek CS-5IR* objective lens. This lens has a lower focal length of 5 mm. This results in a horizontal FOV of about 72°, a vertical FOV of about 40°. For one recording, the DVS was mounted on a telescope mast and raised approximately 3 m in order to replicate the top-down view seen at the tunnel entrance. For a second recording the DVS was hung out of a first floor window and aligned similarly to the setup at the tunnel entrance.

2.3 Content of the Dataset

2.3.1 Recorded at the Tunnel Entrance

The recording taken at the tunnel entrance is split into one hour segments. The recording is interrupted for a few minutes every two weeks because the hard drive was exchanged. The location features some flickering artificial lighting. This causes some noise along the ground and other objects, such as rails, in the environment which varies over time depending on the time of day and what lighting is turned on. An overwhelming majority of the recordings taken at the tunnel entrance contains no pedestrians and depicts a small variety of similar or identical situations. To remedy this class imbalance, the recordings containing no pedestrians are manually cut down to relevant examples of different situations. This facilitates an economical use of computational resources when training and avoids trained models becoming overly biased towards detecting no pedestrians or overfitting on common situations.

The staged samples containing People recorded at the tunnel entrance contain three actors and one dog. The scenarios include:

- Walking in and out of the tunnel along the boardwalk at the wall the DVS is mounted on
- Walking in and out of the tunnel inside the railbed
- Intermittently stopping while walking through the railbed, causing very few events to be triggered
- Running around the railbed waving with both arms in a wide motion

The terminology is explained in Figure 2. The first three scenarios are chosen because we are focused on intrusion detection, meaning we are mainly interested in people entering or exiting the tunnel. The fourth scenario is included to provide an example of persons with a different silhouette from normal walking.

Most of the scenarios are recorded with all actors involved except for the last scenario, in which the dog and its handler are not involved for reasons of practicality. The cable tray (see Figure 2) near the wall the DVS is mounted on results in people occasionally appearing cut off or being hidden entirely.

There are some events besides the appearance of persons which cause activity in the event stream:

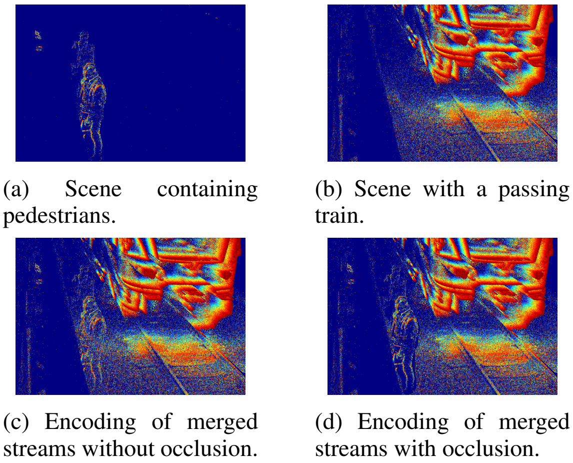
- The artificial lighting being turned on or off, causing bursts of events
- Rain and snow
- Passing trains on three visible rails in different distances
- Reflections on wet floor or rails caused by the lights of passing trains
- Artificial lighting reflecting on wet floor or rails caused by rain
- Flying insects
- Birds flying past or landing at the tunnel entrance
- One instance of a deer entering the tunnel

The remaining recordings taken at the tunnel entrance mostly consist of nothing happening, meaning the only events generated are noise, mostly caused by flickering artificial lights.

2.3.2 Recorded On Campus

The following scenarios were recorded outdoors with the DVS mounted on a telescope mast. A total of four actors were involved in creating these recordings.

- Single person walking away from and back towards the camera, repeated by each actor individually
- Group of three walking away from and back towards the camera fanned out
- Group of three walking away from and back towards the camera in a single file line
- All four actors walking randomly through the recording area



(a) Scene containing pedestrians.

(b) Scene with a passing train.

(c) Encoding of merged streams without occlusion.

(d) Encoding of merged streams with occlusion.

Figure 3: Example of composite frames. Depicted are visualized linear time surfaces. Best viewed in color.

- All four actors walking away from and back towards the camera spread out while stopping intermittently
- Group of two walking away from and back towards the camera
- All four actors walking away from and then back towards the camera while waving both arms in a wide motion

The scenarios were selected according to our use case. Specifically, this means most examples consist of actors walking towards or away from the camera in a straight line, since we are investigating surveillance at a tunnel entrance.

Additional scenarios were recorded with the DVS hung out of a window in order to include examples of rounding corners and moving along the wall, which are relevant in the context of a tunnel entrance, in the dataset.

2.4 Composite Samples

One situation which could not be recorded directly was people entering the tunnel while a train was passing through. Even though there is enough space to walk on the boardwalk near the tunnel wall in this situation, doing so is not safe. Due to this concern, we were not able to create staged recordings of this situation. However, when trying to detect unauthorized entry, this situation cannot be ruled out. Consequently, it must be included in the dataset. To do this, we generate the examples artificially by combining the acquired recordings.

The method we use to achieve this is merging the event streams prior to encoding them into the input format for the detectors. As the event stream is, barring technical errors, always sorted by t, the merge itself is performed like a merge sort step with the event streams as input, potentially with an offset applied to t to merge sections which occur at different times in their respective recordings. For example, combining the event streams visualized in Figure 3a and Figure 3b results in Figure 3c.

This method does not account for occlusion. Since events are, barring noise, only generated at non-homogenous moving areas, any homogeneous parts will appear transparent. To remedy this, we classify one of the event streams to be merged as *foreground* and the other as *background*. After performing noise filtering on the foreground event stream, we use a morphological closing to obtain a mask which shows the approximate occlusion caused by moving *foreground* objects in the scene. Events covered by the mask are then removed from the *background* stream or ignored while merging. The result is pictured in Figure 3d. The events occurring behind persons are removed, but some events erroneously get removed at concave sections of the person's contours.

There are some remaining limitations to this method that have not been solved here. Firstly, the classification of each event stream as either *foreground* or *background* does not allow for three-dimensional depth. This can be problematic, for example when trying to generate recordings featuring rain or snow, where one would expect some droplets to fall in front and other droplets to fall behind the persons or other objects moving through. Secondly, major lighting changes in either scene often do not affect the other scene as they should. For example, when a train passes through with the headlights turned on, one might expect the passing light to sweep across nearby persons, generating events. This effect is not replicated by simply merging the event streams.

2.5 Label Assignment

The recordings are divided into clips and are manually assigned one of two labels based on frames generated from the event stream. We choose to frame our problem as an image classification task instead of an object detection task because we are only interested in the presence of persons, not their exact location. Simplifying the problem in this way allows us to focus on reducing the false positive rate of overall detections while maintaining the near zero false negative rate of the existing system. The label *People* is assigned to clips featuring People, while the label *NoPeople* is assigned to any other clips.

When dividing into training and validation datasets, it is important to select completely different clips for each. Just selecting generated frames randomly from obtained recordings would skew the results. For example, a detector overfitted on three consecutive frames would conceivably still perform well on the middle frame if it was not seen in training while the remaining two were.

The training dataset contains 20min15s of non-composite recordings labeled as containing people which were recorded on campus and 41min10s of recordings labeled as not containing people which were recorded on campus and at the tunnel entrance. Additionally, 1min42s of composite recordings are generated and added as both the merged bins as examples containing people and bins generated from the background only as examples not containing people. Frames are sampled with a stride according to the speed of objects in the scene in order to avoid overfitting. Most scenes, including those just featuring humans, are sampled with a stride of 20, meaning one bin per second of recording. Scenes featuring trains are sampled with a stride of five, including composite scenes. Scenes featuring extremely short-lived events such as flashes of light are not sampled with a stride, or, equivalently, are sampled with a stride of one.

The validation dataset contains 9min16s of non-composite recordings labeled as containing people recorded at the tunnel entrance and 2min48s of regular recordings labeled as not containing people. As was done for the training dataset, 27s of composite recordings are added as corresponding positive and negative examples. All examples in the validation dataset were recorded at the tunnel entrance. No stride is applied to validation data.

The dataset and other supplementary material is available at https://github.com/TuNuKi-DVS/intrusion-railway-dvs.

3 INTRUSION DETECTION

We use three different neural network architectures to classify the encoded dense representations of the event stream as either containing people or not containing people.

3.1 Basic Encoder-Classifier Structure

As a low complexity baseline approach, we consider the following structure:

- One batch normalization layer
- Three encoder blocks consisting of:
  - One 2D convolutional layer with a 3 × 3 kernel
  - One 50% dropout layer
  - One 2 × 2 max pooling layer
- One densely connected layer consisting of ten nodes
- One densely connected output node

The input shape is 360 × 640 × 2 for time surfaces and 360 × 640 × 6 for event volumes.

3.2 MobileNetV2

We investigate the performance of the MobileNetV2 (Sandler et al., 2018) architecture. Being designed with memory efficiency in mind, examining its performance is useful in looking towards an efficient final implementation, which is economically relevant considering the full-time surveillance application. The weights of the encoder portion are initialized with the ImageNet (Russakovsky et al., 2015) weights provided by TensorFlow (Abadi et al., 2016). The classification head is replaced by a two-class head. Since the input layer expects a three-channel image a convolutional layer is inserted in front of the encoder in order to learn a three-channel representation. The input shape is 360 × 640 × 2 for time surfaces and 360 × 640 × 6 for event volumes.

3.3 Yolov8

We investigate the performance of YOLOv8 (Jocher et al., 2023) on our dataset because it is a state-of-the-art architecture in frame based image classification. In order to use YOLOv8 a three-channel visual representation of the time surface encoding is generated using the method provided by Metavision. This converts the time information previously represented by the gray tones of each channel to color values. These images are then used as input to the *yolov8m-cls* model for both training and validation. The weights are initialized to weights pretrained on the ImageNet dataset. The visualizations are input at a resolution of 736 × 736. The change in aspect ratio is achieved by padding the vertical axis with the visualization's background color on both sides, meaning all data in the original image stays intact without cropping. This is important because the classification of images in this case can depend entirely on information around the edges.

3.4 Results

The results achieved by the models are presented in Table 1. The metrics are defined as

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \qquad (1)$$

Table 1: Performance of each model on the validation dataset.

Model	Encoding	Accuracy	Precision	Recall
Basic	Time Surface	0.9530	0.9948	0.9158
Basic	Event Volume	0.9875	**0.9967**	0.9795
MobileNetV2	Time Surface	0.9915	0.9876	**0.9965**
MobileNetV2	Event Volume	**0.9955**	0.9949	**0.9965**
Yolov8	Visualization	0.9913	0.9933	0.9901

$$Precision = \frac{TP}{TP+FP}, \tag{2}$$

$$Recall = \frac{TP}{TP+FN}. \tag{3}$$

TP(true positive) refers to the number of frames containing people classified correctly, *TN(true negative)* refers to the number of frames not containing people classified correctly, *FP(false positive)* refers to the number of frames not containing people falsely classified as containing people, and *FN(false negative)* refers to the number of frames containing people falsely classified as not containing people.

The best overall results are achieved using MobileNetV2 on Event Volumes, followed closely by MobileNetV2 on Time Surfaces and Yolov8 on Time Surface visualizations. The baseline approach trails behind in Accuracy on both Time Surfaces and Event Volumes. This indicates that both MobileNetV2 and Yolov8 would be suitable candidates for an operational system. The choice would mostly depend on other factors such as runtime performance and available hardware.

4 CONCLUSION

In this paper we have presented a dataset recorded over several months at a railway tunnel entrance. While we have investigated the use of this dataset in the context of intrusion detection, the dataset can also be of interest regarding the application of DVS in outdoor settings in general. Additionally, it provides many recordings of moving railway vehicles.

We have approached the intrusion detection task as an image classification problem. While this approach facilitates ease of generating labeled data for training and validation and simplifies the problem to compute, it also suffers from drawbacks. One issue is that the classification output does not provide information on the location of detected persons within the image. Depending on the setup of the camera, this can make it difficult to determine whether the persons detected are actually in the restricted area or in an open area but still in frame. It is also possible that localizing interesting clusters of event prior to classification and classifying only a corresponding section of the generated frames may improve classification performance. Future works should investigate the robustness of bounding box or segmentation based approaches, such as the approach presented in (Perez-Cutino et al., 2021), investigating how this approach improves with higher resolution data and how it performs with the large amount of events generated by passing trains. In addition, the only classes considered at this stage are ”Persons in Frame” and ”No Persons in Frame” or effectively ”background”. This results in the detector effectively performing presence detection, which could be more efficiently achieved by other types of sensors, such as thermal sensors, without the need for a neural network. The potential of the chosen approach lies in further analyzing the situation, such as differentiating between entry by humans and entry by animals, and whether entering animals pose a significant risk of causing an accident. Realizing this potential will require the collection of additional data in future works. Additional data collection will also be required to evaluate the generalizability of the trained detector, since our validation data were collected exclusively using the setup depicted in Figure 2.

Furthermore, the networks we tested in this work were trained and run on a GPU. Considering that this is a surveillance application, it is worthwhile to investigate energy efficiency and reducing the number of operations performed during inference while maintaining classification quality and real-time performance in future works.

ACKNOWLEDGMENTS

This work was supported by the Federal Ministry for Digital and Transport under the mFUND program grant number *19F1115A* as part of the project *TUNUKI*.

In addition, we would like to thank *DB Station&Service AG*, the German *Federal Police* and *Masasana GmbH* for their assistance in this research.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. arXiv:1603.04467 [cs].

Alonso, I. and Murillo, A. C. (2019). EV-SegNet: Semantic Segmentation for Event-Based Cameras. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1624–1633, Long Beach, CA, USA. IEEE.

Benosman, R., Clercq, C., Lagorce, X., Ieng, S.-H., and Bartolozzi, C. (2014). Event-Based Visual Flow. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2):407–417. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

Bisulco, A., Cladera Ojeda, F., Isler, V., and Lee, D. D. (2020). Near-Chip Dynamic Vision Filtering for Low-Bandwidth Pedestrian Detection. In *2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 234–239. ISSN: 2159-3477.

Bolten, T., Neumann, C., Pohle-Fröhlich, R., and Tönnies, K. (2023). N-MuPeTS: Event Camera Dataset for Multi-Person Tracking and Instance Segmentation:. In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 290–300, Lisbon, Portugal. SCITEPRESS - Science and Technology Publications.

Bolten, T., Pohle-Frohlich, R., and Tonnies, K. D. (2021). DVS-OUTLAB: A Neuromorphic Event-Based Long Time Monitoring Dataset for Real-World Outdoor Scenarios. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1348–1357, Nashville, TN, USA. IEEE.

Catalano, A., Bruno, F. A., Galliano, C., Pisco, M., Persiano, G. V., Cutolo, A., and Cusano, A. (2017). An optical fiber intrusion detection system for railway security. *Sensors and Actuators A: Physical*, 253:91–100.

de Tournemire, P., Nitti, D., Perot, E., Migliore, D., and Sironi, A. (2020). A Large Scale Event-based Detection Dataset for Automotive. arXiv:2001.08499 [cs, eess].

Delbruck, T. (2008). Frame-free dynamic digital vision. In *Proceedings of Intl. Symp. on Secure-Life Electronics, Advanced Electronics for Quality Life and Society*, volume 1, pages 21–26. Citeseer.

D'Amore, P. and Tedesco, A. (2015). Technologies for the Implementation of a Security System on Rail Transportation Infrastructures. In Setola, R., Sforza, A., Vittorini, V., and Pragliola, C., editors, *Railway Infrastructure Security*, volume 27, pages 123–141. Springer International Publishing, Cham. Series Title: Topics in Safety, Risk, Reliability and Quality.

Iaboni, C., Kelly, T., and Abichandani, P. (2023). NU-AIR – A Neuromorphic Urban Aerial Dataset for Detection and Localization of Pedestrians and Vehicles. arXiv:2302.09429 [cs].

Jiang, Z., Xia, P., Huang, K., Stechele, W., Chen, G., Bing, Z., and Knoll, A. (2019). Mixed Frame-/Event-Driven Fast Pedestrian Detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8332–8338, Montreal, QC, Canada. IEEE.

Jocher, G., Chaurasia, A., and Qiu, J. (2023). YOLO by Ultralytics (Version 8.0.0) [Computer software]. https://github.com/ultralytics/ultralytics.

Miao, S., Chen, G., Ning, X., Zi, Y., Ren, K., Bing, Z., and Knoll, A. (2019). Neuromorphic Vision Datasets for Pedestrian Detection, Action Recognition, and Fall Detection. *Frontiers in Neurorobotics*, 13.

Mueggler, E., Forster, C., Baumli, N., Gallego, G., and Scaramuzza, D. (2015). Lifetime estimation of events from Dynamic Vision Sensors. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4874–4881. ISSN: 1050-4729.

Perez-Cutino, M., Eguiluz, A. G., Dios, J. M.-d., and Ollero, A. (2021). Event-based human intrusion detection in UAS using Deep Learning. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 91–100, Athens, Greece. IEEE.

Perot, E., de Tournemire, P., Nitti, D., Masci, J., and Sironi, A. (2020). Learning to Detect Objects with a 1 Megapixel Event Camera. In *Advances in Neural Information Processing Systems*, volume 33, pages 16639–16652. Curran Associates, Inc.

Prophesee (2023). Metavision SDK by Prophesee (Version 4.1.0) [Computer Software].

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, Salt Lake City, UT. IEEE.

Siraj, A., Vaughn, R., and Bridges, S. (2004). Intrusion sensor data fusion in an intelligent intrusion detection system architecture. In *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*, page 10 pp., Big Island, HI, USA. IEEE.

Wan, J., Xia, M., Huang, Z., Tian, L., Zheng, X., Chang, V., Zhu, Y., and Wang, H. (2021). Event-Based Pedestrian Detection Using Dynamic Vision Sensors. *Electronics*, 10(8):888. Number: 8 Publisher: Multidisciplinary Digital Publishing Institute.

Zhu, A. Z., Yuan, L., Chaney, K., and Daniilidis, K. (2019). Unsupervised Event-Based Learning of Optical Flow, Depth, and Egomotion. pages 989–997.