

# EV-SegNet: Semantic Segmentation for Event-based Cameras

Iñigo Alonso  
Universidad de Zaragoza, Spain  
inigo@unizar.es

Ana C. Murillo  
Universidad de Zaragoza, Spain  
acm@unizar.es

## Abstract

Event cameras, or Dynamic Vision Sensor (DVS), are very promising sensors which have shown several advantages over frame-based cameras. However, most recent works on real applications of these cameras are focused on 3D reconstruction and 6-DOF camera tracking. Deep learning based approaches, which are leading the state-of-the-art in visual recognition tasks, could potentially take advantage of the benefits of DVS, but some adaptations are needed still needed in order to effectively work on these cameras. This work introduces the first baseline for semantic segmentation with this kind of data. We build a semantic segmentation CNN based on state-of-the-art techniques which takes event information as the only input. Besides, we propose a novel representation for DVS data that outperforms previously used event representations for related tasks. Since there is no existing labeled dataset for this task, we propose how to automatically generate approximated semantic segmentation labels for some sequences of the DDD17 dataset, which we publish together with the model, and demonstrate they are valid to train a model for DVS data only. We compare our results on semantic segmentation from DVS data with results using corresponding grayscale images, demonstrating how they are complementary and worth combining. Code: <https://github.com/Shathe/Ev-SegNet>

## 1. Introduction

Event cameras, as Dynamic Vision Sensor (DVS) [21], are promising sensors which register intensity changes of the captured environment. In contrast to conventional cameras, this sensor does not acquire images at a fixed frame-rate. These cameras, as their name suggests, capture events and record a stream of asynchronous events. An event indicates an intensity change at a specific moment and at a particular pixel (more details on how events are acquired in Section 3.2). Event cameras offer multiple advantages over more conventional cameras, 1) mainly its very high temporal resolution, which allows the capture of multiple events in

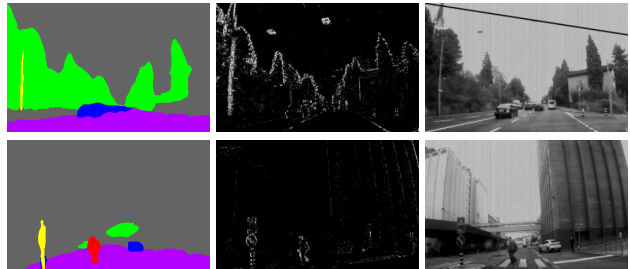


Figure 1. Two examples of semantic segmentation (left) from event based camera data (middle). The semantic segmentation is the prediction of our CNN, fed only with event data. Grayscale images (right) are displayed only to facilitate visualization. Best viewed in color.

microseconds; 2) its very high dynamic range, which allows the information capture at difficult lighting environments; 3) its low power and bandwidth requirements. Maqueda et al. [24] show how visual recognition tasks can benefit from these advantages in their work emphasizing that event cameras are natural motion detectors and automatically filter out any temporally-redundant information. Besides, they show that these cameras provide richer information than just subtracting consecutive conventional images.

These cameras offer a wide range of new possibilities and features that could boost solutions for many computer vision applications. However, new algorithms still have to be developed in order to fully exploit their capabilities, especially regarding recognition tasks. Most of the latest achievements based on deep learning solutions for image data have not yet been even attempted on event cameras. One of the main reasons is the output of these cameras: they do not provide standard images, and there is not yet a clearly adopted way of representing the stream of events to feed a CNN. Another challenge is the lack of labeled training data, which is key to training most recognition models. Our work includes simple but effective novel ideas to deal with these two challenges. They could be helpful in many DVS applications, but we focus on an application not explored yet with this sensor, semantic segmentation.

This work proposes to combine the potential of event cameras with deep learning techniques on the challenging

task of semantic segmentation. Semantic segmentation may intuitively seem a task much better suited to models using appearance information (from common RGB images). However, we show how, with an appropriate model and representation, event cameras provide very promising results for this task.

Figure 1 shows two visual results as an example of the output of our work. Our main contributions are:

- First results, up to our knowledge, on semantic segmentation using DVS data. We build an Xception-based CNN that takes this data as input. Since there is no benchmark available for this problem, we propose how to generate approximated semantic segmentation labels for some sequences of the DDD17 event-based dataset. Model and data are being released.
- A comparison of different DVS data representation performance on semantic segmentation (including a new proposed representation that is shown to outperform existing ones), and an analysis of benefits and drawbacks compared to conventional images.

## 2. Related work

### 2.1. Event Camera Applications

As previously mentioned, event cameras provide valuable advantages over conventional cameras in many situations. We find recent works which have proved these advantages in several tasks typically solved with conventional vision sensors. Most of these works focus their efforts on 3D reconstruction [29, 19, 36, 35] and 6-DOF camera tracking [30, 11]. Although 3D reconstruction and localization solutions are very mature on RGB images, existing algorithms cannot be applied exactly the same way on event cameras. The aforementioned works propose different approaches for adapting them.

We find recent approaches that explore the use of these cameras for other tasks, such as optical flow estimation [12, 23, 37] or, closer to our target tasks, object detection and recognition [27, 6, 20, 31]. Regarding the data used in these recognition works, Orchard et al. [27] and Lagorce et al. [20] performed the recognition task on small datasets, detecting mainly characters and numbers. The most recent works, start to use more challenging (but scarce) recordings in real scenarios, such as N-CARS dataset, used in Sironi et al. [31], or DDD17 dataset [2], which we use in this work because of the real world urban scenarios it contains.

Most of these approaches have a common first step: encode the event information into an image-like representation, in order to facilitate its processing.

We discuss in detail different previous work event representations (encoding spatial and temporal information) as

well as our proposed representation (with a different way of encoding the temporal information) in Sec. 3.

### 2.2. Semantic Segmentation

Semantic segmentation is a visual recognition problem which consists of assigning a semantic label to each pixel in the image. State-of-the-art on this problem is currently achieved by deep learning based solutions, most of them proposing different variations of encoder-decoder CNN architectures [5, 4, 17, 16].

Some of the existing solutions for semantic segmentation target an instance-level semantic segmentation, e.g., Mask-RCNN [15], that includes three main steps: region proposal, binary segmentation, and classification. Other solutions, such as DeepLabv3+ [5], target class-level semantic segmentation. DeepLabv3+ is a fully convolutional extension of Xception [7], which is also a state-of-the-art architecture for image classification and the base architecture of our work. A survey on image segmentation by Zhu et al. [38] provides a detailed compilation of more conventional solutions for semantic segmentation, while Garcia et al. [13] present a discussion of more recent deep learning based approaches for semantic segmentation, covering from new architectures to common datasets.

The works discussed so far show the effectiveness of CNNs for semantic segmentation using *RGB* images. Closer to our work, we find additional works which prove great performance in semantic segmentation tasks using additional input data modalities to the standard *RGB* image. For example, a common additional input data for semantic segmentation is depth information. Cao et al. [3] and Gupta et al. [14] are two good examples of how to combine *RGB* images with depth information using convolutional neural networks. Similarly, a very common sensor in the robotics field, the Lidar sensor, has also been shown to provide useful additional information when performing semantic segmentation [33, 10]. Other works show how to combine less frequent modalities such as fluorescence information [1] or how to perform semantic segmentation on multi-spectral images [10]. Semantic segmentation tasks for medical image analysis [22] also typically apply or adapt CNN based approaches designed for *RGB* images to different medical imaging sensors, such as MRI images [18, 25] and CT data [8].

Our work is focused on a different modality, event camera data, not explored in prior work for semantic segmentation. Following one of the top performing models on semantic segmentation for *RGB* images [5], we base our network on the Xception design [7] to build an encoder-decoder architecture for semantic segmentation on event images. Our experiments show good semantic segmentation results using only event data from a public benchmark [2], close to what is achieved on standard imagery from the

same scenarios. We also demonstrate the complementary benefits that this modality can bring when combined with standard cameras to solve this problem more accurately.

### 3. From Events to Semantic Segmentation

In this section, we will discuss different event representations used in visual recognition tasks in order to end up proposing a rich encoding of the event data for semantic segmentation.

#### 3.1. Event data

Event cameras capture the changes in intensities for each pixel. The output of an event camera is not a 3-dimensional image (height, width, and channels) as conventional cameras but a stream of events. An event represents the positive or negative change in the log of the intensity signal (over an established threshold  $\sigma$ ):

$$\log(I_{t+1}) - \log(I_t) \geq \sigma, \quad (1)$$

being  $I_{t+1}$  and  $I_t$  the intensity captured at two consecutive timestamps. Each event ( $e_i$ ) is then defined by four different components: two coordinates ( $x_i, y_i$ ) of the pixel where the change has been measured, a polarity ( $p_i$ ) that can be positive or negative, and a timestamp ( $t_i$ ):

$$e_i = \{x_i, y_i, p_i, t_i\}. \quad (2)$$

Events are asynchronous and have the described encoding that, by construction, does not provide good input for broadly used techniques in visual recognition tasks, such as CNNs. Perhaps the most straightforward representation would be a  $n \times 4$  matrix, with  $n$  the number of events. But obviously, this representation does not encode the spatial relationship between events. Several strategies have been proposed to encode this information into a dense representation successfully applied in different applications.

#### 3.2. Event Representation

**Basic dense encoding of event location.** The most successfully applied event data representation creates an image with several channels encoding the following information. It stores at each location ( $x_i, y_i$ ) information from the events that happened there at any time  $t_i$  within an established integration interval of size  $T$ . Variations of this representation have been used by many previous works, showing great performance in very different applications: optical flow estimation [37], object detection [6], classification [20, 28, 31] and regression tasks [24], respectively.

Earlier works used only one channel to encode event occurrences. Nguyen et al. [26] stores the information of the last event that has occurred in each pixel, i.e., the corresponding value chosen to represent a positive event, negative event or absence of events. One important drawback is that only the last event information remains.

In a more complete representation, a recent work for steering wheel angle estimation, from Maqueda et al. [24], stores the positive and negative event occurrences into two different channels. In other words, this representation (*Hist*) encodes the 2D histogram of positive and negative events that occurred at each pixel ( $x_i, y_i$ ), as follows:

$$Hist(x, y, p) = \sum_{i=1, t_i \in W}^N \delta(x_i, x) \delta(y_i, y) \delta(p_i, p), \quad (3)$$

where  $\delta$  is the Kronecker delta function (the function is 1 if the variables are equal, and 0 otherwise),  $W$  is the time window, or interval, considered to aggregate the event information, and  $N$  is the number of events occurred within interval  $W$ . Therefore, the multiplication  $\delta(x_i, x) \delta(y_i, y) \delta(p_i, p)$  denotes whether an event  $e_i$  matches its coordinates  $x_i, y_i$  with  $x, y$  values and its polarity  $p_i$  with  $p$ . This representation has two channels, one per polarity  $p$  (positive and negative events). Our proposed representation described later, will make use of these two histogram channels.

Note that all the representations discussed so far only use the temporal information (timestamps  $t_i$ ) to see the time interval where each event belongs to.

#### Dense encodings including temporal information.

However, temporal information, i.e., the timestamp of each event  $t_i$ , contains useful information for recognition tasks, and it has been shown that including this non-spatial information of each event into the image-like encodings is useful. Lagorce et al. [20] propose a 2-channel image, one channel per polarity, called *time surfaces*. They store, for each pixel, information relative only to the most recent event timestamp during the integration interval  $W$ . Later, Sironi et al. [31] enhance this previous representation by changing the definition of the *time surfaces*. They now compute the value for each pixel combining information from all the timestamps of events that occurred within  $W$ .

Another recently proposed approach by Zhu et al. [37] introduces a more complete representation that includes both channels of event occurrence histograms from Maqueda et al. [24], and two more channels containing temporal information. These two channels (*Recent*) store, at each pixel ( $x_i, y_i$ ), the normalized timestamp of the most recent positive or negative event, respectively, that occurred in that location during the integration interval:

$$Recent(x, y, p) = \max_{t_i \in W} t_i \delta(x_i, x) \delta(y_i, y) \delta(p_i, p). \quad (4)$$

All these recent representations normalize the event timestamps and histograms to be relative values within the interval  $W$ .

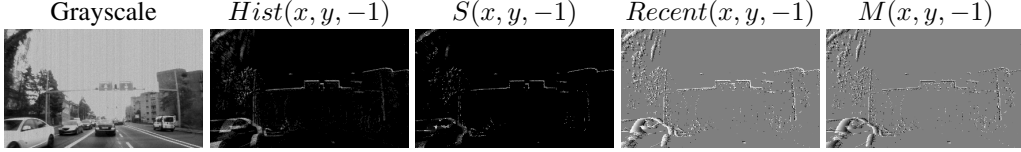


Figure 2. Visualization (between 0 and 255 gray values) of different 1-channel encodings of data from events with negative polarity ( $p = -1$ ) explained in the Sect. 3.2. In these examples the event information has been integrated for a time interval of 50ms ( $T = 50ms$ ). Grayscale is shown as reference.

Inspired by all this prior work, we propose an alternative representation that combines the best ideas demonstrated so far: the 2-channels of event histograms to encode the spatial distribution of events, together with information regarding all timestamps occurred during the integrated time interval.

**Our proposed representation.** We propose a 6-channel image representation. The first two channels are the histogram of positive and negative events (eq. 3). The remaining four channels are a simple but effective way to store information relative to all event timestamps happening during interval  $W$ . We could see it as a way to store how they are distributed along  $T$  rather than selecting just one of the timestamps. We propose to store the mean ( $M$ ) and standard deviation ( $S$ ) of the normalized timestamps of events happening at each pixel  $(x_i, y_i)$ , computed separately for the positive and negative events, as follows:

$$M(x, y, p) = \frac{1}{Hist(x, y, p)} \sum_{i=1, t_i \in W}^N t_i \delta(x_i, x) \delta(y_i, y) \delta(p_i, p), \quad (5)$$

$$S(x, y, p) = \sqrt{\frac{\sum_{i=1, t_i \in W}^N (t_i \delta(x_i, x) \delta(y_i, y) \delta(p_i, p) - Mean(x, y, p))^2}{Hist(x, y, p) - 1}}. \quad (6)$$

Then, our representation consists of these six 2D-channels:  $Hist(x, y, -1)$ ,  $Hist(x, y, +1)$ ,  $M(x, y, -1)$ ,  $M(x, y, +1)$ ,  $S(x, y, -1)$ ,  $S(x, y, +1)$ . Figure 2 shows a visualization of some of these channels. In the event representation images, the brighter the pixels, the higher the value encoded, e.g., white means the highest number of negative events in the  $Hist(x, y, -1)$ .

### 3.3. Semantic Segmentation from Event Data

CNNs have already been shown to work well on dense event-data representations, detailed in previous section [24, 37], therefore we explore a CNN based architecture to learn a different visual task, semantic segmentation.

Semantic segmentation is often modeled as a per-pixel classification, and therefore the output of semantic segmentation models has the same resolution that the input. As

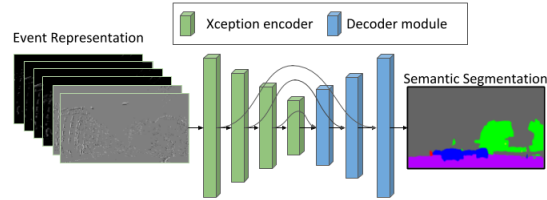


Figure 3. Semantic segmentation from event based cameras. We process the different 2D event-data encodings with our encoder-decoder architecture based on Xception [7] (Sec. 3.3 for more details). Best viewed in color.

previously mentioned, there are plenty of recent successful CNN-based approaches to solve this problem both using RGB data and additional modalities. We have built an architecture inspired on current state-of-the-art semantic segmentation CNNs, slightly adapted to use the event data encodings. Related works commonly follow an encoder-decoder architecture, as we do. As the encoder, we use the well-known Xception model [7], which has been shown to outperform other encoders, both in classification [7] and semantic segmentation tasks [5]. As the decoder, also following state-of-the-art works [4, 5], we build a *light* decoder, concentrating the heavy computation on the encoder. Our architecture also includes features from the most successful recent models for semantic segmentation, including: the use of skip connections to help the optimization of deep architectures [16, 17] to avoid the vanishing gradient problem and the use of an auxiliary loss [34] which also improves the convergence of the learning process. Fig. 3 shows a diagram of the architecture built in this work, with the multi-channel event representation as network input.

As similar architectures, we perform the training optimization via back-propagation of the loss, calculated as the summation of all per-pixel losses, through the parameter gradients. We use the common soft-max cross entropy loss function ( $\mathcal{L}$ ) described in eq. (7):

$$\mathcal{L} = -\frac{1}{N} \sum_{j=1}^N \sum_{c=1}^M y_{c,j} \ln(\hat{y}_{c,j}), \quad (7)$$

where  $N$  is the number of labeled pixels and  $M$  is the number of classes.  $y_{c,j}$  is a binary indicator of pixel  $j$  belonging

<b>Dataset Classes:</b>	<i>flat</i> (road and pavement), <i>background</i> (construction and sky), <i>object</i> , <i>vegetation</i> , <i>human</i> , <i>vehicle</i>	
<b>Train Sequences</b>	Selected suitable sequence intervals	Num. Frames
1487339175	[0, 4150), [5200, 6600)	5550
1487842276	[1310, 1400), [1900, 2000), [2600, 3550)	1140
1487593224	[870, 2190)	995
1487846842	[380, 500), [1800, 2150), [2575, 2730), [3530, 3900)	1320
1487779465	[1800, 3400), [4000, 4700), [8400, 8630), [8800, 9160), [9920, 10175), [18500, 22300)	6945
	<b>TOTAL:</b>	<b>15950</b>
<b>Test Sequences</b>	Selected suitable sequence intervals	Num. Frames
1487417411	[100, 1500), [2150, 3100), [3200, 4430), [4840, 5150)	<b>3890</b>

Table 1. Summary of Ev-Seg Data which consists of several intervals of some sequences of the DDD17 dataset.

to class  $c$  (ground truth).  $\hat{y}_{c,j}$  is the CNN predicted probability of pixel  $j$  belonging to class  $c$ .

#### 4. Ev-Seg: Event-Segmentation Data

The Ev-Seg data is an extension for semantic segmentation of the DDD17 dataset [2]. The DDD17 dataset consists of 40 sequences of different driving set-ups. These sequences were recorded on different scenarios (e.g., motorways and urban scenarios). This dataset provides synchronized grayscale and event-based information but, it does not provide semantic segmentation labels.

Our extension includes generated (automatically generated, non-manual annotations) semantic segmentation labels to be used as ground truth for a large subset of that dataset. Besides the labels, to facilitate replication and further experimentation, together with the labeling, we also publish the selected subset of grayscale images and corresponding event data encoded with three different representations (Maqueda et al. [24], Zhu et al. [37] and the new one proposed in this work).

**Generating the labels.** Besides the obvious burden of manually labeling a semantic segmentation per-pixel ground truth, if we think of performing this task directly on the event-based data it turns out even more challenging. We only need to look at any of the event representations available (see Fig. 1), to realize that for the human eye is hard to distinguish many of the classes there if the grayscale image is not side-by-side. Other works have shown how CNNs are robust to training with noise [32] or approximated labels[1], including the work of Chen et al. [6] that also successfully uses generated labels from grayscale for object detection in event-based data. We then propose to use the corresponding grayscale images to generate an approximated set of labels for training, which we demonstrate is enough to train models to segment directly on event-based data.

To generate these approximated semantic labels, we performed the following three steps.

First, we have trained a CNN for semantic segmentation on the well known urban environment dataset *Cityscapes* [9], but using grayscale images. The architecture used for this step is the same architecture described in

subsection 3.3, which follows state-of-the-art components for semantic segmentation. This grayscale segmentation model was trained for 70 epochs with a learning rate of  $1e-4$ . The final model obtains 83% categories MIoU on the *Cityscapes* validation data. This is still a bit far from the top results obtained on that dataset with RGB images (92% MIoU), but enough quality for our the process.

Secondly, with this grayscale model, we obtained the semantic segmentation on all grayscale images of the selected sequences (we detail next which sequences were used and why). These segmentations are what we will consider the labels to train our event-based segmentation model.

Lastly, as a final post-processing step on the ground truth labels, we cropped the bottom part of all the images, i.e., 60 bottom rows of the image it always contains the car dashboard and it only introduces noise into the generated labels.

**Subset of DDD17 sequences selection.** As previously mentioned, we have not generated the labels for all the DDD17 data. We next discuss the reasons and selection criteria that we followed.

Some DDD17 sequences did not give good labels when being pass through the CNN. There are several reasons for this. As the data domain available to train the base grayscale semantic segmentation model was *Cityscapes* data, which is an urban domain, we selected only the DDD17 sequences from urban scenarios. Besides, only images with enough contrast (not too bright, not too dark) are likely to provide a good generated ground truth. Therefore, we only selected sequences which were recorded during day-time, with no extreme overexposure. Given these restrictions, only six sequences approximately matched them. Therefore, we performed a manual more detailed annotation of the intervals in each of these sequences where the restrictions applied (details on Table 1).

**Data summary.** Table 1 shows a summary of the contents of the Ev-Seg data. From the six sequences selected as detailed previously, five sequences were used as training data and one sequence was used for testing. We chose for testing the sequence with more homogeneous class distribution, i.e., that contained more amount of labels of categories

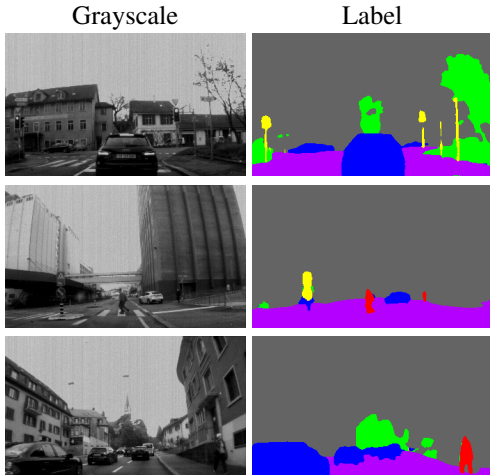


Figure 4. Three examples of the Ev-Seg data generated for the test sequence. Semantic label images (right) have been generated from the grayscale images (left) through a CNN trained on a grayscale version of Cityscapes. Best viewed in color.

which appears less such as the human/pedestrian label.

The labels have the same categories than the well-known *Cityscapes* dataset [9] (see Table 1), with the exception of *sky* and *construction* categories. Although these two categories were properly learned in the *Cityscapes* dataset, when performing inferences on the *DDD17* dataset with grayscale images, these categories were not correctly generated due to the domain-shift. Therefore in our experiments, those two categories are learned together, as if they were the same thing. This domain shift between the *Cityscapes* and *DDD17* datasets was also the cause of generating the *Cityscapes* categories instead of its classes.

Figure 4 shows three examples of grayscale images and corresponding generated segmentation that belong to our extension of the *DDD17* dataset. We can see that although the labels are not as perfect as if manually annotated (and as previously mentioned, classes such as *building* and *sky* were not properly learned using only grayscale), they are pretty accurate and well defined.

## 5. Experimental Validation

### 5.1. Experiment Set-up and Metrics

**Metrics.** Our work addresses the semantic segmentation problem, i.e., per pixel classification, using event cameras. Thus, we evaluate our results on the standard metrics for classification and semantic segmentation: *Accuracy* and *Mean Intersection over Union (MIoU)*.

In semantic segmentation, given a predicted image  $\hat{y}$  and a ground-truth image  $y$ , and being  $N$  their number of pixels, which can be classified in  $C$  different classes, the accuracy

metric, eq. (8) is computed as:

$$Accuracy(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N \delta(y_i, \hat{y}_i), \quad (8)$$

and the MIoU is calculated per class as:

$$MIoU(y, \hat{y}) = \frac{1}{C} \sum_{j=1}^C \frac{\sum_{i=1}^N \delta(y_{i,c}, 1) \delta(y_{i,c}, \hat{y}_{i,c})}{\sum_{i=1}^N \max(1, \delta(y_{i,c}, 1) + \delta(\hat{y}_{i,c}, 1))}, \quad (9)$$

where  $\delta$  denotes the Kronecker delta function,  $y_i$  indicates the class where pixel  $i$  belongs to, and  $y_{i,c}$  is a boolean that indicates if pixel  $i$  belongs to a certain class  $c$ .

**Set-up.** We perform the experiments using the CNN explained in Sec. 3.3. and the Ev-Seg data detailed in Sec. 4. We train all model variations from scratch using: the Adam optimizer with an initial learning rate of  $1e-4$  and a polynomial learning rate decay schedule. We train for  $30K$  iterations using a batch size of 8 and during training we perform several data augmentation steps: crops, rotations ( $-15^\circ$ ,  $15^\circ$ ), vertical and horizontal shifts ( $-25\%$ ,  $25\%$ ) and horizontal flips. Regarding the event information encoding, for training, we always use an integration time interval  $T = 50ms$  which has been shown to perform well on this dataset [24].

### 5.2. Event Semantic Segmentation

**Input representation comparison.** A good input representation is very important for a CNN to properly learn and exploit the input information. Table 2 compares several semantic segmentation models trained with different input representations. The top three rows correspond to event-based representations. We compare a basic dense encoding of event locations, a dense encoding which also includes temporal information and our proposed encoding (see Sec.3.2. for details). Our event encoding performs slightly but consistently better on the semantic segmentation task on the different metrics and evaluations considered. Fig. 5 shows a few visual examples of these results.

All models (same architecture, just trained with different inputs) have been trained with data encoded using integration intervals of 50ms, but we also evaluate them using different interval sizes. This is an interesting evaluation because by changing the time interval, in which the event information is aggregated, we somehow simulate different camera movement speeds. In other words, intervals of 50ms or 10ms may encode exactly the same movement but at different speeds. This point is pretty important because, in real scenarios, models have to perform well at different speeds. We can see that all models perform just slightly worse on test data encoded with different intervals sizes (10ms, 250ms) that the integration time used during

Input representation	Accuracy 50ms	MIoU 50ms	Accuracy 10ms	MIoU 10ms	Accuracy 250ms	MIoU 250ms
Basic dense encoding [24]	88.85	53.07	85.06	42.93	87.09	45.66
Temporal dense encoding [37]	88.99	52.32	86.35	43.65	85.89	45.12
Ours	<b>89.76</b>	<b>54.81</b>	<b>86.46</b>	<b>45.85</b>	<b>87.72</b>	<b>47.56</b>
Grayscale	94.67	64.98	94.67	64.98	94.67	64.98
Grayscale & Ours	<b>95.22</b>	<b>68.36</b>	<b>95.18</b>	<b>67.95</b>	<b>95.29</b>	<b>68.26</b>

Table 2. Semantic segmentation performance of different input representations on the test Ev-Seg data. Models trained using time intervals ( $T$ ) of 50ms but tested with different  $T$  values: 50ms, 10ms and 250ms.

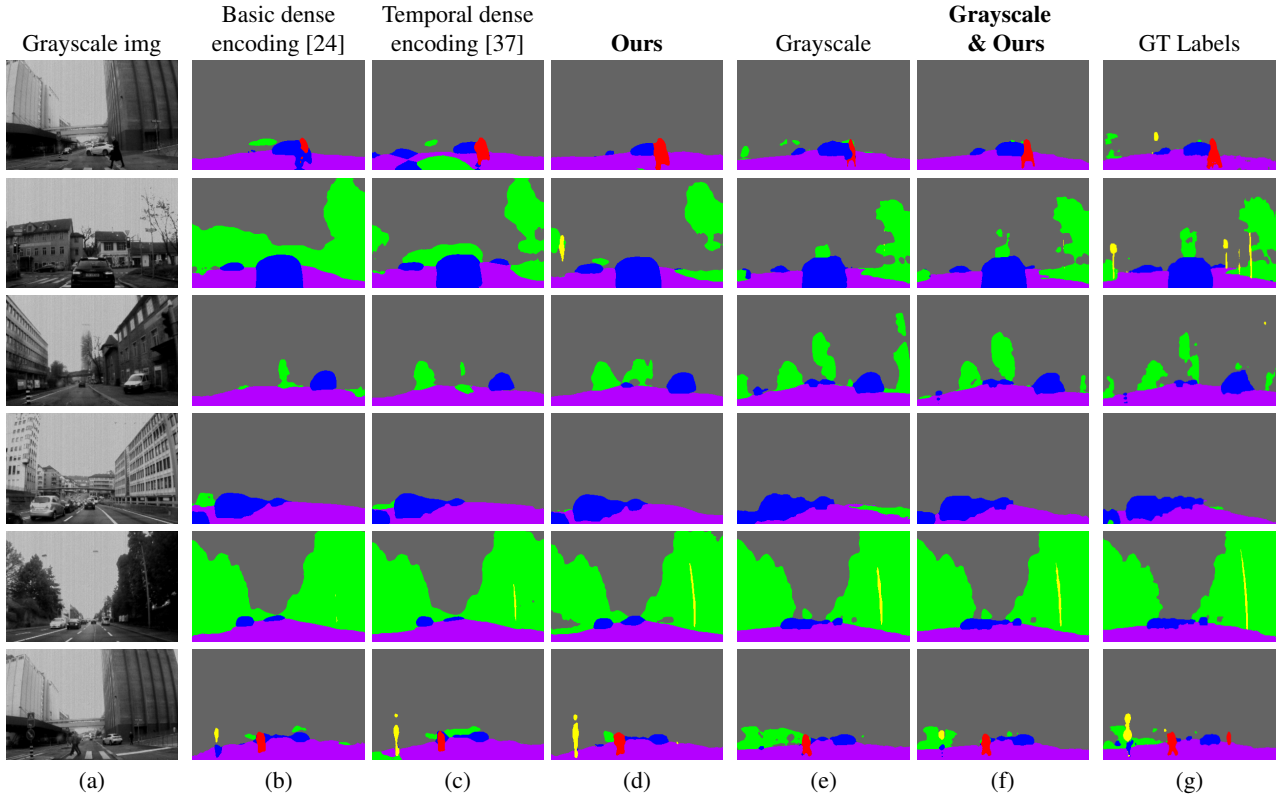


Figure 5. Semantic segmentation on several test images from Ev-Seg data. Results using different input representations of event data only, (b) to (d), or using grayscale data (e) and (f). Grayscale original image (a) and ground truth labels are shown for visualization purposes. Models trained and tested on time intervals of 50ms. Best viewed in color. See supplementary material video for the full sequence.

training (50ms), see Fig. 6 examples. There are two main explanations for why the models are performing similarly on different integration intervals: 1) the encodings are normalized and 2) the training data contains different camera speeds. Both things help to generalize better at different time intervals or movement speeds.

**Event vs conventional cameras.** Table 2 also includes, in the two bottom rows, results using the corresponding grayscale image for the semantic segmentation task.

Although conventional cameras capture richer pure appearance information than event cameras, event cameras provide motion information, which is also very useful for

the semantic segmentation task. In examples of results using grayscale data from Fig. 5(e), (f), we can see how event information helps for example to better segment moving objects, such as pedestrians (in red in those examples) or to refine object borders. While conventional cameras suffer detecting small objects and in general, with any recognition on extreme illumination (bright or dark) conditions, event cameras suffer more in recognizing objects with no movement (because they move at the same speed than the camera or because they are too far to appreciate their movement).

Conventional cameras perform better on their own for semantic segmentation than event-based cameras on their own. However, our results show that semantic segmenta-

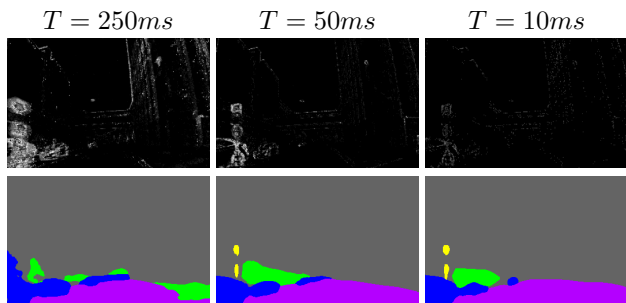


Figure 6. Semantic segmentation results (bottom) using different integration interval size ( $T$ ) for the event data representation (top). Results obtained with a model trained only on 50ms integrated event information encoded with our proposed representation. Best viewed in color.

tion results are better when combining both of them. This suggests they are learning complementary information. Interestingly, we should note that the data available for training and evaluation is precisely data where we could properly segment the grayscale image, therefore slightly more beneficial for grayscale images than event-based data (i.e., there is no night-time image included in the evaluation set because there is no ground truth for those).

Two clear complementary situations from our experiments: 1) On one hand, it is already known that one the major drawback of event cameras is that objects that do not move with respect to the camera do not trigger events, i.e., are invisible. Fig. 7 shows an example of a car waiting at a pedestrian crossing, where we see that while conventional cameras can perfectly see the whole scene, event cameras barely capture any information; 2) On the other hand, event cameras are able to capture meaningful information on situations where scene objects are not visible at all for conventional vision sensors, e.g., difficult lighting environments. This is due to their high dynamic range, Fig. 8 illustrates an example of a situation where neither of the grayscale nor the event-based models have been trained for. The event-based model performs much better due to the minor domain-shift on the input representation.

## 6. Conclusions and Future Work

This work includes the first results on semantic segmentation using event camera information. We build an Xception-based encoder-decoder architecture which is able to learn semantic segmentation only from event camera data. Since there is no benchmark available for this problem, we propose how to generate automatic but approximate semantic segmentation labels for some sequences of the DDD17 event-based dataset. Our evaluation shows how this approach allows the effective learning of semantic segmentation models from event data. Both models and generated labeled data are being released.

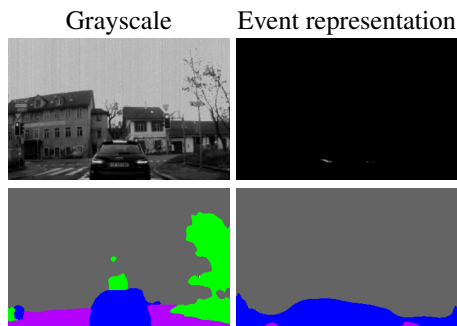


Figure 7. Semantic segmentation result (bottom) on a static part of the sequence, i.e, a car waiting at a crossing. This is an obvious adversarial case for event cameras, due to lack of event information. Best viewed in color.

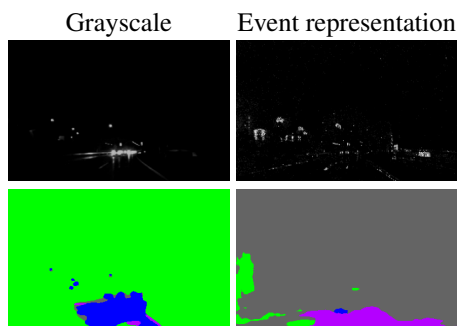


Figure 8. Semantic segmentation (bottom) on extreme lighting conditions (night-time) with different input representations (top): grayscale image and our event data representation. Corresponding models trained only on good illuminated daytime samples. This is an obvious adversarial case for conventional cameras, due to lack of information in the grayscale capture. Best viewed in color.

In order to feed the model, we also propose a novel event camera data representation, which encodes both the event histogram and their temporal distribution. Our semantic segmentation experiments, comparing different representations, show that our approach allows the effective learning of semantic segmentation models and that our approach outperforms other previously used event representations, even when evaluating in different time intervals. We also compare the segmentation achieved only from event data to the segmentation from conventional images, showing their benefits, their drawbacks and the benefits of combining both sensors for this task.

For future work, one of the main challenges is still obtaining and generating more and better semantic segmentation labels, through alternative domain adaptation approaches and/or event camera simulators (they currently do not provide this kind of labels). Besides, it would be also interesting, and not only for the currently explored recognition problem of segmentation, to develop alternative architectures and data augmentation methods more specific for event-based cameras.



## References

- [1] I. Alonso, A. Cambra, A. Munoz, T. Treibitz, and A. C. Murillo. Coral-segmentation: Training dense labeling models with sparse ground truth. In *IEEE Int. Conf. on Computer Vision Workshops*, pages 2874–2882, 2017.
- [2] J. Binas, D. Neil, S.-C. Liu, and T. Delbruck. Ddd17: End-to-end davis driving dataset. *ICML Workshop on Machine Learning for Autonomous Vehicles*, 2017.
- [3] Y. Cao, C. Shen, and H. T. Shen. Exploiting depth from single monocular images for object detection and semantic segmentation. *IEEE Transactions on Image Processing*, 26(2):836–846, 2017.
- [4] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.
- [5] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.
- [6] N. F. Chen. Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under ego-motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 644–653, 2018.
- [7] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, 2017.
- [8] P. F. Christ, M. E. A. Elshaer, F. Ettliger, S. Tatavarty, M. Bickel, P. Bilic, M. Rempfler, M. Armbruster, F. Hofmann, M. DANastasi, et al. Automatic liver and lesion segmentation in ct using cascaded fully convolutional neural networks and 3d conditional random fields. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 415–423. Springer, 2016.
- [9] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of IEEE conf. on CVPR*, pages 3213–3223, 2016.
- [10] C. Dechesne, C. Mallet, A. Le Bris, and V. Gouet-Brunet. Semantic segmentation of forest stands of pure species combining airborne lidar data and very high resolution multispectral imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 126:129–145, 2017.
- [11] G. Gallego, J. E. Lund, E. Mueggler, H. Rebecq, T. Delbruck, and D. Scaramuzza. Event-based, 6-dof camera tracking from photometric depth maps. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2402–2412, 2018.
- [12] G. Gallego, H. Rebecq, and D. Scaramuzza. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *IEEE Int. Conf. Comput. Vis. Pattern Recog.(CVPR)*, volume 1, 2018.
- [13] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.
- [14] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European Conference on Computer Vision*, pages 345–360. Springer, 2014.
- [15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *IEEE Int. Conf. on Computer Vision*, pages 2980–2988, 2017.
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of IEEE CVPR*, pages 770–778, 2016.
- [17] S. Jégou, M. Drozdal, D. Vazquez, A. Romero, and Y. Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *CVPRW*, pages 1175–1183. IEEE, 2017.
- [18] B. Kayalibay, G. Jensen, and P. van der Smagt. Cnn-based segmentation of medical imaging data. *arXiv preprint arXiv:1701.03056*, 2017.
- [19] H. Kim, S. Leutenegger, and A. J. Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *European Conference on Computer Vision*, pages 349–364. Springer, 2016.
- [20] X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman. Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1346–1359, 2017.
- [21] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128×128 120 db 15μs latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008.
- [22] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.
- [23] M. Liu and T. Delbruck. Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. Technical report, 2018.
- [24] A. I. Maqueda, A. Loquercio, G. Gallego, N. Garcia, and D. Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5419–5427, 2018.
- [25] F. Milletari, N. Navab, and S.-A. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.
- [26] A. Nguyen, T.-T. Do, D. G. Caldwell, and N. G. Tsagarakis. Real-time pose estimation for event cameras with stacked spatial lstm networks. *arXiv preprint arXiv:1708.09011*, 2017.
- [27] G. Orchard, C. Meyer, R. Etienne-Cummings, C. Posch, N. Thakor, and R. Benosman. Hfirst: a temporal approach to object recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(10):2028–2040, 2015.
- [28] P. K. Park, B. H. Cho, J. M. Park, K. Lee, H. Y. Kim, H. A. Kang, H. G. Lee, J. Woo, Y. Roh, W. J. Lee, et al. Performance improvement of deep learning based gesture recognition using spatiotemporal demosaicing technique. In *Image*

- Processing (ICIP), 2016 IEEE International Conference on*, pages 1624–1628. IEEE, 2016.
- [29] H. Rebecq, G. Gallego, E. Mueggler, and D. Scaramuzza. Emvs: Event-based multi-view stereo3d reconstruction with an event camera in real-time. *International Journal of Computer Vision*, pages 1–21, 2017.
- [30] H. Rebecq, T. Horstschaefer, G. Gallego, and D. Scaramuzza. Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robotics and Automation Letters*, 2(2):593–600, 2017.
- [31] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1731–1740, 2018.
- [32] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *IEEE Int. Conf. on Computer Vision*, 2017.
- [33] Y. Sun, X. Zhang, Q. Xin, and J. Huang. Developing a multi-filter convolutional neural network for semantic segmentation using high-resolution aerial imagery and lidar data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2018.
- [34] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.
- [35] Y. Zhou, G. Gallego, H. Rebecq, L. Kneip, H. Li, and D. Scaramuzza. Semi-dense 3d reconstruction with a stereo event camera. *ECCV*, 2018.
- [36] A. Z. Zhu, Y. Chen, and K. Daniilidis. Realtime time synchronized event-based stereo. *CVPR*, 2018.
- [37] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898*, 2018.
- [38] H. Zhu, F. Meng, J. Cai, and S. Lu. Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. *Journal of Visual Communication and Image Representation*, 34:12–27, 2016.