SUPPLEMENTAL MATERIAL
How Many Events Make an Object?
Improving Single-frame Object Detection on the 1 Mpx Dataset

Alexander Kugele[1,2], Thomas Pfeil[3], Michael Pfeiffer[1], Elisabetta Chicca[2]
[1] Bosch Center for Artificial Intelligence, Renningen, Germany
[2] University of Groningen, Groningen, Netherlands [3] Recogni GmbH, Munich, Germany
{alexander.kugele,michael.pfeiffer3}@de.bosch.com, tom@recogni.com, e.chicca@rug.nl

1. Failure cases of the bounding box memory

To understand when our proposed bounding box memory fails, we extracted single frames and short sequences from the validation set and show them in Fig. 1. Two rows in a subfigure depict the bounding boxes without (top) and with memory. If only one row is shown, it is the predictions with memory. In Fig. 1a and Fig. 1b, two success cases are shown: In a), The memory remembers a prediction from the past and adds it to the current prediction because of the low event count. A new prediction updates the memory, the older bounding box is forgotten. In b), a prediction is deleted due to the low event count. Comparing to the ground truth, we see that it is a false positive. The other cases are failure cases: In c), the detector manages to make predictions under low event counts (130 and 250 events) that coarsely fit a ground truth box. These detections are deleted due to the low event count, but there also exists no bounding box in memory to compensate. In d), boxes are forgotten because the threshold is crossed much faster for smaller boxes. Having an area-dependent threshold could improve this. In e), the memory forgets the box due to the high event count, but the detector also misses a prediction. The scene in f) shows a car coming from the left that is occluded by another car. The detector detects a box with the wrong shape, due to the occlusion. The memory remembers this shape, which leads to multiple false positives due to the low event count. In g), an occluded object appears again, but does not move and therefore does not generate events. The detector cannot detect it at any point in time, and therefore also the memory does not help. In h), a car is correctly detected in a frame, but it is not labeled in the dataset, most likely it was missed by the automatic labeling procedure. Our memory (correctly) remembers it, but as there is no ground truth label, it leads to many more (wrong) false positives than without the memory.

2. Details of single-frame architecture

The following section lists a few details of our architecture to enable reproducible results. Our architecture consists of a backbone, a neck and a detection head (see Fig. 3b, and Fig. 2 for the backbone). Our single-frame single-shot detector has a ResNet [2] or ResNeXt [12] backbone. The first layer of the pretrained backbone is replaced to fit the number of channels of the event volume: As in [6], we use five time bins with two polarities each, resulting in ten input channels. This layer is randomly initialized. One additional convolutional layer is added after the backbone which downsamples the number of features to 512, to reduce the number of parameters in the neck and head. The heads use sigmoid (one-vs-all) outputs for the classes and regress on the relative locations

$$\mathrm{loc}_{\mathrm{xy}} = \frac{\mathrm{box}_{\mathrm{xy}} - \mathrm{prior}_{\mathrm{xy}}}{0.1\mathrm{prior}_{\mathrm{xy}}} \tag{1}$$

$$\mathrm{loc}_{\mathrm{wh}} = \log\left(\frac{\mathrm{box}_{\mathrm{wh}}}{0.2\mathrm{prior}_{\mathrm{wh}}}\right), \tag{2}$$

as in Faster R-CNN [8]. To tune prior box parameters, we maximize the intersection-over-union (IoU) between prior and ground truth bounding boxes of the 1 Mpx Dataset on a subset of the training labels and find that prior boxes with side lengths of (18, 40, 61, 101, 151, 202) and aspect ratios of (0.5, 1, 2, 3) perform best. Hard negative mining is used to always have an objects-to-background ratio of at least $1:3$. Random cropping around a random bounding box is used as data augmentation. We use the full training set during each epoch. After each epoch, the network is evaluated on a validation set and training is stopped if the validation mAP does not improve over a fixed amount of epochs. A learning rate schedule with an initial learning rate of 0.002 and decays of 0.2 at 5, 85 and $90\,\%$ of epochs is used in conjunction with the Adam optimizer [3].

When using the 1 Mpx Dataset, we always do one full

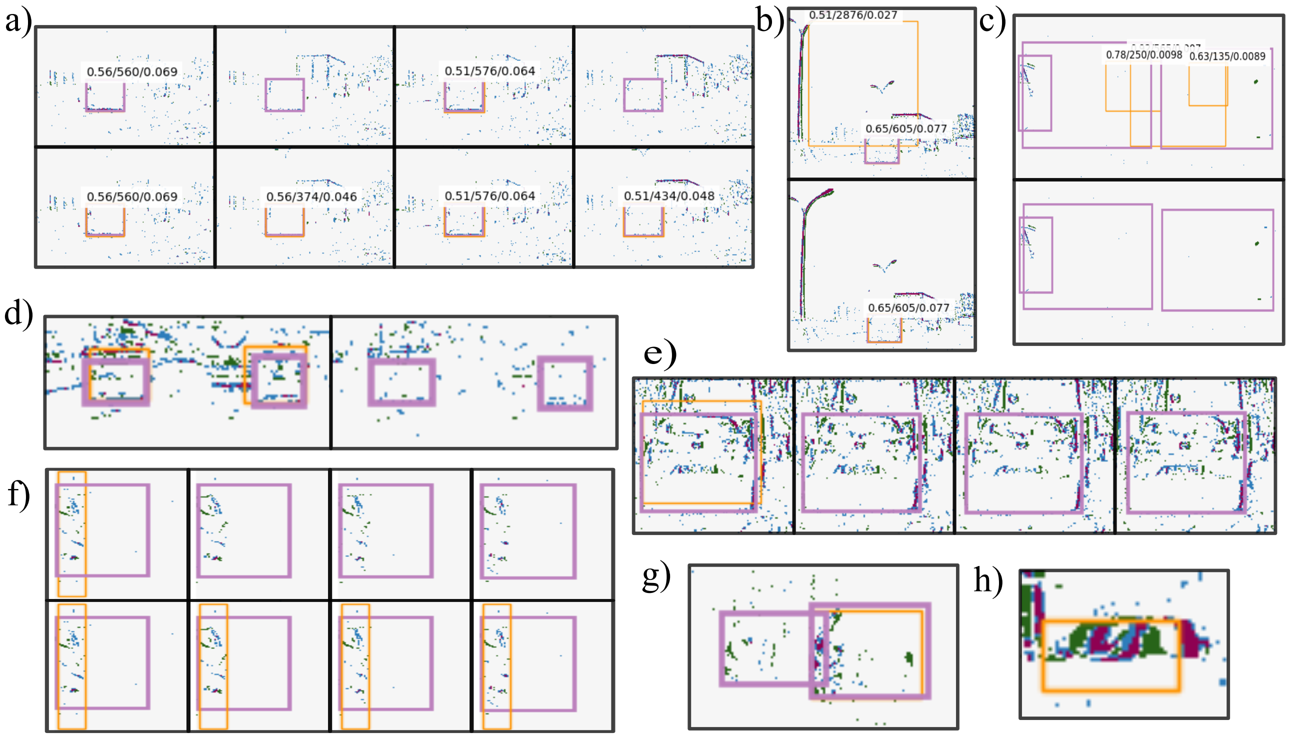

Figure 1. Success and failure cases of the memory. Purple is the ground truth, orange the prediction. **a)** Memory remembers box. **b)** A false positive is deleted because of the low event count. **c)** True positives are deleted because of low event count. **d)** Memory forgets box due to noise **e)** Memory forgets box due to high event count, but detector also misses detection **f)** Detector detects wrong shape, is kept in memory due to low event count **g)** An occluded object appears but with a low event count, is never detected **h)** An object is correctly detected, but there's no ground truth label.

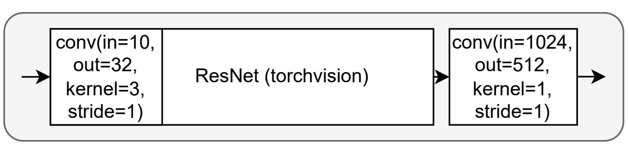

Figure 2. Backbone of our single-shot detector. We use a ResNet-18 and ResNeXt-50 from torchvision and replace the first convolutional layer to fit the number of channels of our voxel input. The last layer downsamples the number of feature maps.

pass through the training set and then evaluate on 70 % of the validation set. This saves time and has proven to give similar results to using the full validation set. The best network (on the validation set) is saved and training is done after 10 epochs or after the validation mAP did not change after 6 consecutive epochs. Additionally, we replace the ResNet-18 backbone with a ResNeXt-50 networks. We did not need this complexity for the simpler RM-MNIST, but it improves the results on the 1 Mpx Dataset significantly.

A comparison between the architecture design of RED and our single-frame detector with memory is shown in Fig. 3. While RED uses ConvLSTM layers in their neck to memorize an abstract internal state, our proposed bounding box memory memorizes the bounding boxes.

3. Details of the RM-MNIST simulation and training

During simulation, we generate a new frame every 1 ms to simulate events with the Event Camera Simulator [5]. Digits are rescaled randomly between 20x20 pixels and 320x180 pixels, but stay the same in a sequence. Random digits from the training set of MNIST are used for the training set of RM-MNIST, and the same for validation and test. To generate the bounding boxes, we search for the first and last non-zero pixel in the x- and y-direction, such that bounding boxes are tight around each digit.

The network backbone is a ResNet-18 from torchvision. Networks are trained for 30 epochs. For the experiments with ConvLSTM layers, each layer has 128 output channels.

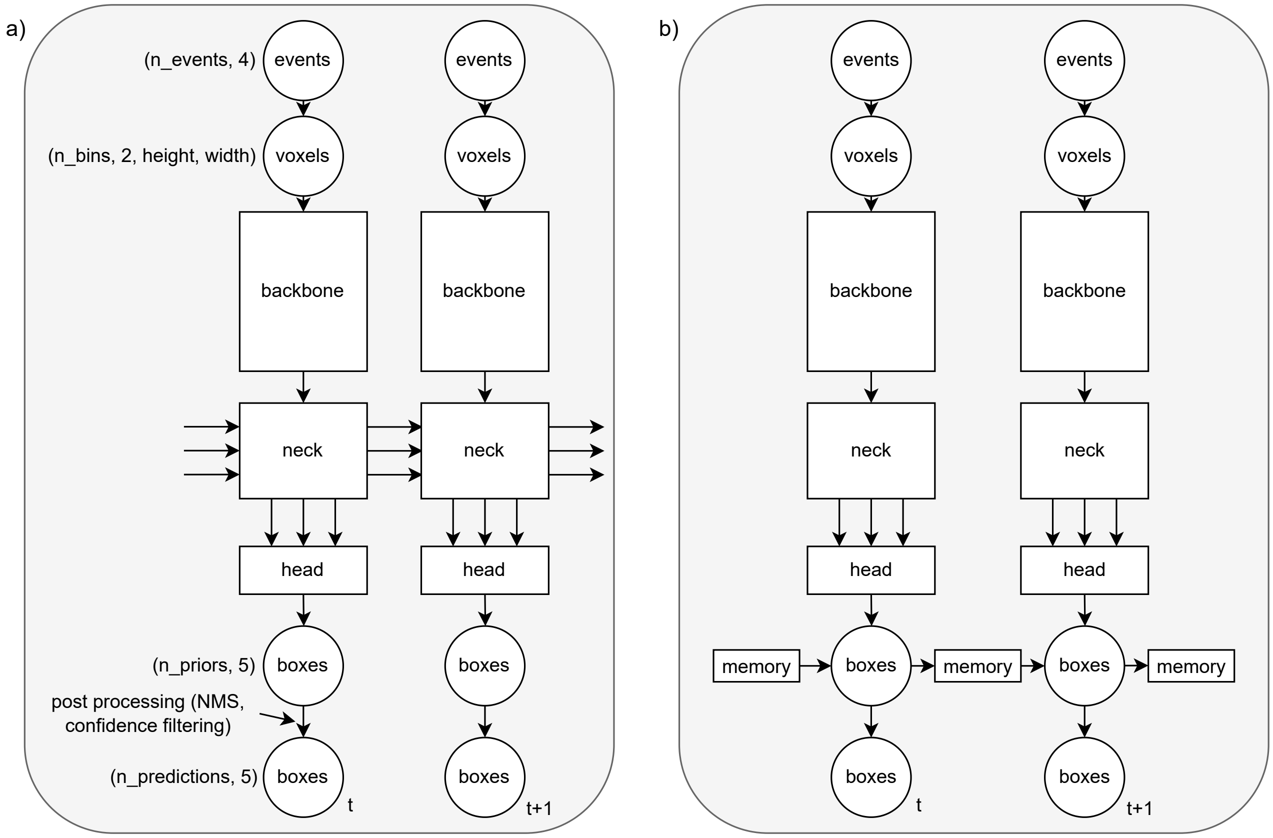

Figure 3. Schematic of **a)** RED and **b)** our single-frame approach. The main difference is how information is memorized in time. RED choses a neck of ConvLSTM layers, while we utilize our bounding box memory. Time goes from left to right.

4. More detailed description of RED

For the interested reader, we summarize here the main features of the Recurrent Event Detector (RED) architecture that is proposed in conjunction with the 1 Mpx Dataset [6] and to this date the only architecture (besides this paper) that works on the full 1 Mpx Dataset.

The baseline architecture is a single-frame, single-shot detector [4] with a feature extractor built from Squeeze-and-Excitation blocks [9] and multiple heads to detect objects at multiple scales. Three different event representations are compared: Histogram, Timesurface [11] and event volumes (time-space voxels) [1, 13], where event volumes perform best. Additionally, the authors evaluate a frame-like representation from a trained events-to-frames conversion network [7], which has a lower mAP than their network. The authors replace the last convolutional layers by convolutional LSTM (ConvLSTM) layers [10], which leads to a significant boost in mAP. At each box head, bounding boxes are predicted for the current time step and also for one time step in the future; the authors argue that these Dual Regression Heads improve detection consistency. This network is called Recurrent Event-camera Detector (RED).

Our method in comparison is much simpler: We don't use ConvLSTM layers or Dual Regression Heads, and instead rely on dataset filtering and a simple memory mechanism as described in Sec. 3.4 (main paper) and Sec. 3.5 (main paper). As backbone, we use an off-the-shelf ResNet-18 [2] and ResNeXt-50 [12] from torchvision, which is described in more detail in Sec. 3.6 (main paper).

5. Detailed results of RM-MNIST and 1 Mpx Dataset

We report the values for the bar charts (Fig. 5 and Fig. 6b of the main paper) in Tab. 1 and Tab. 2. Results are discussed in the main text.

architecture	mAP
single-frame (SF)	0.2685
SF + dataset filtering (DF)	0.309 ± 0.036
SF + DF + memory	0.825 ± 0.053
SF + ConvLSTM	0.263 ± 0.061
frames: SF	0.855 ± 0.015
frames: SF +DF	0.855 ± 0.015
frames: SF + ConvLSTM	0.769 ± 0.026
frames: SF + filter train & test	0.762 ± 0.020
events: SF + filter train & test	0.780 ± 0.022

Table 1. Results on RM-MNIST. When filtering train and test, the results are not comparable to the previous experiments, due to the change in the test dataset. This experiment just confirms that when we remove all frames without events from the test dataset, that the single-frame detector performance is the same, regardless of using frames or events.

architecture	mAP
single-frame (SF)	$0.180\,03 \pm 0.000\,38$
SF + dataset filtering (DF)	$0.204\,45 \pm 0.000\,85$
SF + DF + memory	$0.213\,95 \pm 0.000\,78$
SF + ConvLSTM	0.1604 ± 0.0042
SF + filter@0	$0.225\,67 \pm 0.000\,95$
SF + filter@10	$0.230\,47 \pm 0.000\,50$
SF + filter@100	$0.257\,57 \pm 0.000\,45$
SF + filter@1000	0.3246 ± 0.0011
SF + filter@10 000	0.3902 ± 0.0012

Table 2. Results on the 1 Mpx Dataset. filter@100 means that we filter out all bounding boxes with at most 100 events from the training, validation and test dataset. The experiments where we filter the test set show that objects that contain a lot of events are easier to detect. They cannot be compared to RED or our other experiments, because we change the test dataset.

References

[1] Daniel Gehrig, Antonio Loquercio, Konstantinos G. Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 3

[3] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014. 1

[4] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *Lecture Notes in Computer Science*, pages 21–37, 2016. 3

[5] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research*, 36(2):142–149, Feb 2017. 2

[6] Etienne Perot, Pierre de Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. Learning to detect objects with a 1 megapixel event camera. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16639–16652. Curran Associates, Inc., 2020. 1, 3

[7] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015. 1

[9] Abhijit Guha Roy, Nassir Navab, and Christian Wachinger. Recalibrating fully convolutional networks with spatial and channel 'squeeze & excitation' blocks, 2018. 3

[10] Xingjian SHI, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 3

[11] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad 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. 3

[12] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 3

[13] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3