Accuracy-efficiency trade-off for using event-based data when performing bounding box-based object detection

Pascal Benschop
Supervisors: Nergis Tömen, Ombretta Strafforello, Xin Liu
EEMCS, Delft University of Technology, The Netherlands

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 19, 2022

Abstract

Event-based cameras do not capture frames like an RGB camera, only data from pixels that detect a change in light intensity, making it a better alternative for processing videos. The sparse data acquired from event-based video only captures movement in an asynchronous way. In this paper an evaluation is made on the efficiency and accuracy of object detection, specifically localization, between sparse and dense representations of data. Convolutional Neural Networks are used to train and test on images and event-based data. The results show a positive trade-off in terms of accuracy and efficiency for using sparse event-based data instead of dense data like images. These results provide a basis for an argument to use event-based cameras instead of RGB cameras when dealing with object detection. The code for this research is available on GitHub[1].

1 Introduction

Object detection is an important aspect of computer vision, it forms the basis of other computer vision tasks such as segmentation and tracking [1]. In this research the emphasis is on bounding box-based object detection. A bounding box is a rectangle containing an object that defines where the object is located in a frame and what dimensions the object has. A machine learning model can learn to detect the bounding boxes in a new frame given the ground truth bounding boxes for each object in the training frames. For image-based object detection, Convolutional Neural Networks (CNNs) are often used. In this paper event-based data is used for object detection with these neural networks. There are multiple sources on processing event-based data [2], however there is minimal research on the comparisons of processing event-based data as opposed to image-based data.

Event-based data is represented as a stream of asynchronous events which capture changes in light intensity of a pixel. An event contains data of the location, timestamp and polarity of the event. The location is stored as an x and y coordinate. The polarity indicates whether an event occurred when the light intensity decreases or increases over a certain threshold. For all events in this research, the polarity value is discarded. The reason for this is that an object at a certain time interval can consist of events that have a positive polarity and events that have a negative polarity. The stream of asynchronous events can be converted to a sparse input representation for a CNN compared to dense image-based data.

The difference in data representations as input to an object detection CNN is interesting to investigate because of the performance increase that could be achieved by processing less data. CNNs do not perform well on sparse data, therefore events are often converted to an image-like representation [3]. The time surface representation [2] is the only event-based representation adapted for this research that resembles an image. However, it is not the most sparse representation. In this research the term "time surface" is exchanged with "time frame" since it represents a frame of time values. In a time frame a large portion of data is not useful since only pixels where events occurred have a value. The time frame representation is used in this research since it retains the temporal data of events and is further explained in section 3. Events can also be transformed to a set of points in two or three dimensions. By keeping only the x and y coordinates of events a set of 2D coordinates is created. When the timestamp is also added a set of 3D coordinates is created. These sets of points are used as sparse event-based data representations.

The research question is: "What is the accuracy-efficiency trade-off of an object detection convolutional neural network for using sparse event-based data instead of dense image-based data?". For this research question two hypotheses are made:

1. Using event-based data is more efficient and similar in accuracy compared to using images as input for an object detection CNN.
2. Using event-based data can lead to a better accuracy for object detection than image-based data at a similar efficiency.

The paper first reviews existing literature related to the research topic, then goes on to explain the experiments for the hypotheses. After the experiments, the results are presented and discussed. Finally the paper provides conclusions for the research and future research recommendations.

2 Related work

A popular object detection model for conventional images is YOLO (you only look once) [4], a single convolutional network that predicts multiple bounding boxes and class probabilities for these boxes. Since YOLO is fast and accurate it provides a good basis for a model that can be used to test the research question.

Cannici et al. [5] proposed an event-based adaptation of YOLO and an event-based CNN. The object detection model YOLE (you only look events) takes as input events that are integrated into a frame-based representation. The structure of this model is used as inspiration for the model constructed in this paper. Also, Cannici et al discussed a method called eFCN (event-based fully convolutional object detection) to detect objects from events without any preprocessing. However, the eFCN model itself is not used in this research. For the more sparse representation of events in this research another approach is taken, this can be seen in section 3.2.

The representations of event-based data used in this research either transform the input into another dimension (3D point cloud), disregard the use of temporal data

[1] https://github.com/pascalbenschopTU/Event-based-object-detection

(2D point cloud) or disregard the sparsity of events (time frame). Messikommer et al. [3] proposed a model that utilizes the sparse nature of events combined with the temporal data to perform object detection. The model itself is not used in this research, however, the results from the paper complement the findings in this research. These results are shown in figure 1, the event-based model constructed in the paper is significantly more efficient compared to a standard convolutional model and has a high accuracy.

		N-Caltech101		Gen1 Automotive	
	Representation	mAP ↑	MFLOP ↓	mAP ↑	MFLOP ↓
YOLE [30]	Leaky Surface	0.398	3682	-	-
Standard Conv.	Event Queue	0.619	1977	**0.149**	2614
Ours		0.615	**200**	0.119	**205**
Standard Conv.	Event Histogram	0.623	1584	0.145	2098
Ours		**0.643**	**200**	0.129	**205**

Figure 1: Event-based Asynchronous Sparse CNN results. "Ours" describes the model used by the paper, "MFLOP" describes the efficiency of the model in how many millions of floating point operations were executed, and "mAP" describes the accuracy of the model. Source: [3]

Gehrig et al. [6] conducted experiments on the accuracy and efficiency of different event-based representations for classification. This paper inspired the formulation of the research question. Nevertheless, the research contains no description on how event-based representations compare to image-based representations and is not focused on object detection.

Perot et al. [7] investigated the performance of a real-time recurrent neural network architecture on a high-resolution event-based detection dataset. The neural network performed similarly in accuracy with a lower runtime compared to Gray-RetinaNet, which is an adaptation of a frame-based neural network for grayscale images. The results of this experiment can be seen in figure 2. The paper describes that when colors are used the accuracy of the RetinaNet detector increased to 0.56 mAP for the 1Mpx Detection Dataset, a 30% increase from 0.43 mAP achieved with Gray-RetinaNet. The Recurrent Event-camera Detector is not perfect since the sparsity of events is not fully exploited. There is also no direct comparison of using event-based data with using images in color in terms of accuracy and efficiency.

The discussed papers give results on the speed and accuracy of predictions, however, results supporting an advantage of using event-based data compared to images for object detection are missing. In this research direct comparisons between event-based data representations and images are made.

3 Methodology

For the proposed hypotheses, two experiments are carried out. These experiments complement each other in

	1Mpx Detection Dataset			Gen1 Detection Dataset	
	mAP	runtime (ms)	params (M)	mAP	runtime (ms)
MatrixLSTM [50]	-	-	-	0.31*	-
SparseConv [13]	-	-	-	0.15	-
Events-RetinaNet	0.18	44.05	32.8	0.34	18.29
E2Vid-RetinaNet	0.25	840.66	43.5	0.27	263.28
RED (ours)	**0.43**	**39.33**	**24.1**	0.40	**16.70**
Gray-RetinaNet	**0.43**	41.43	32.8	**0.44**	17.35

* Provided by the authors, using a pretrained YOLOv3.

Figure 2: Recurrent Event-camera Detector (RED) results, Gray-RetinaNet is the only image-based detector. Source: [7]

exploring what the accuracy-efficiency trade-off is for using events instead of images. Before the methodology of the experiments is stated, the definitions of the metrics used in these experiments are given. In the first experiment, event-based data and images are used in two simple CNN models. This experiment provides results on the trade-off for using event-based data with the focus on efficiency. In the second experiment, a novel model is used on images and frames constructed from events. This experiment provides results on the accuracy trade-off for using event-based data.

3.1 Metrics

The efficiency mentioned in the research question is measured in both time needed for training the model, and time needed for predicting a selection of inputs. The idea behind this is that efficiency can be measured as the amount of operations per time interval, which is equal to measuring the time taken for a certain amount of operations. Since all experiments are executed on the same hardware and software, the amount of time a single operation takes is constant on average. For this reason the times were measured instead of the total amount of operations.

The accuracy of the models used in the first experiment is measured with intersection over union (IOU) of predicted and ground truth bounding boxes. IOU is used in this research to test how close the bounding box is to the ground truth and it is stated in percentages. The procedure for finding the IOU is explained in figure 3.

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} =$$

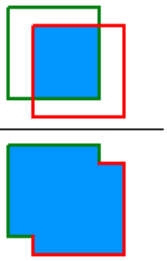

Figure 3: IOU, Source: [8]

In the second experiment in this research the accuracy of the predictions is stated in mean Average Precision

(mAP). This metric is calculated with precision and recall values. The precision is defined as the ability of a model to identify only relevant objects, the recall is defined as the ability of a model to find all relevant objects [8]. The mAP metric is more difficult to calculate than IOU and generally used when multiple objects need to be detected in a frame.

3.2 CNNs with event-based data

For this experiment events are converted to data representations mentioned in section 1. Simple Convolutional Neural Networks are used to test the accuracy and efficiency of bounding box-based object detection across different input data representations.

The first data representation in the experiment is not event-based, it is an image that is used as a comparison to the event-based data representations. The image representation used in this research is in red, green and blue (RGB) format, this means that there are 3 channels or simply said 3 frames in red, green and blue per image. These frames consist of normalized values from 0 to 1, these values are acquired by dividing the value of each of the colors red, green and blue by 256. Each image is scaled to frames of 128 by 128 pixels. The image representation is the most dense data representation used in this research with a total of 49152 values.

A less dense and event-based representation is the time frame where events are encoded onto a 2D map based on the recency of events. The encoding is as follows: a 2D map is made where the X and Y coordinate of events are used as indexes and the timestamp as value. The values are taken and normalized over a time period of 10ms, consequently, the final 2D map contains only values from 0 to 1. These values are then squared to highlight the most recent events. The resulting 2D map is resized to a 128 by 128 pixel grayscale image which contains a total of 16384 values.

The simplest and most sparse representation is having only the coordinates of the events as a set of points. The amount of points used as input for the model is specified with N where N = 500. This representation does not make use of the temporal data from events, however for 2D bounding box-based object detection this is not strictly necessary. To incorporate the time value of events a 3D point set was made. The X and Y coordinate combined with the timestamp form a 3D set of points of length N. The timestamp, originally in microseconds, is converted to milliseconds by dividing by 1000. This improved the accuracy of the model. The 2D set of points contains 1000 values and the 3D set of points contains 1500 values for each input.

Models: With the event-based data in the respective formats, different models are constructed to train and test with the data. All code is executed on a graphics processing unit (GPU) to speed up the process. The GPU used is a NVIDIA RTX 2060 super.

A simple object detection model is made to evaluate the performance of bounding box regression on these formats. The model is made on the basis of the YOLE model [5] and is presented in figure 4. The model consists of 2 stages. In the first stage 5 blocks of convolutional and max-pool layers are used with leaky rectified linear unit activation (leaky ReLu) layers. In the second stage 3 blocks of fully connected layers are used to retain features from the data. In between the stages a dropout is used to prevent the model from over-fitting. The model uses a mean-squared-error loss and an Adam [9] optimizer with a learning rate of 0.001. All data is trained over 40 epochs with a batch size of 40. The model is implemented with TensorFlow[2] and Keras[3] in Python.

The same model is constructed using sparse layers since the object detection model does not exploit the sparsity of the event-based data. The sparse model is constructed using sparse convolutional and max pooling layers in the same layout as the simple model. These layers are implemented using code from [10]. The leaky ReLu layers and the fully connected layers remain the same. The sparse model is implemented with PyTorch[4] in Python.

Data setup: The proposed models are used on a selection of the original dataset Caltech101 [11] and the event-based dataset N-Caltech101 [12]. The selection consists of 4 classes: airplanes, cars, helicopters and motor bikes. The training data is first parsed to the correct representation. When the data preprocessing is complete, the training data is split up for training and testing in an 80-20 percent split. The part for training is split up again for training and validation in an 80-20 percent split. The training and validation data is used as input for the model. The model is thus trained with only 80 percent of the data, leaving some unused data for testing.

3.3 YOLOv3 with event-based data

The models used in the previous experiment are not optimized for object detection, for this reason a novel CNN model is used to train on images and event-based data. The model used is YOLOv3 [13], and this model is trained on the same GPU.

For the input of the YOLOv3 model, the data is similar to the data used in the experiment from section 3.2. The event-based representation used is a time frame, which is converted to a JPG (image) file. The images are already in JPG format. The same selection of classes from the same datasets of the previous experiment is used for training.

The YOLOv3 model can be trained via scripts that read a dataset in a specified format. The format consists of two folders named images and labels, each containing two folders for training and validation. The labels contain the coordinates of the bounding boxes for each object in a frame. To create the dataset in the required

[2]https://github.com/tensorflow/tensorflow
[3]https://github.com/fchollet/keras
[4]https://github.com/pytorch/pytorch

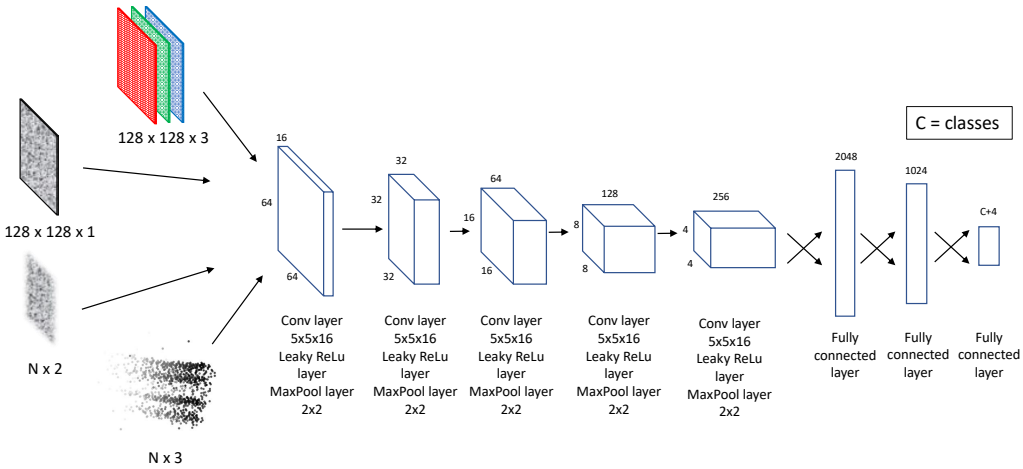

Figure 4: Simple Convolutional Neural Network model, the input representations from top to bottom are as follows: image, time frame, 2D point cloud, 3D point cloud.

format, the preprocessed data from the previous experiment is stored in the correct folders. For the command that executes the YOLOv3 training script the batch size was chosen at 40, the amount of epochs is set at 20 and the input image size is 256 by 256 pixels.

In order to verify the experiment and its results, the model is also trained on the entire Caltech101 and N-Caltech101 datasets. The same procedure is used to transform the data into usable input for the YOLOv3 model. The entire Caltech101 dataset consists of 101 classes of objects. From these 101 classes 100 were used since one class was not properly represented in both the standard and event-based version of the Caltech dataset. Since the data size increased, the amount of epochs to train the model is increased from 20 to 40.

4 Results

For the experiments performed in section 3, results are demonstrated that support the answer to the hypotheses. In the first subsection the results of the first experiment regarding simple models are presented. In the second subsection the results of the second experiment regarding the YOLOv3 model are given.

4.1 Accuracy vs efficiency

The results from the simple neural network are shown in figure 5. The image-based representation performs the best in terms of accuracy which is expected since this representation contains the most data. It can also be seen that the more sparse the representation becomes, the faster the model prediction timing becomes. The accuracy, measured in IOU, is lower for a more sparse representation.

To test whether the results are accurate, the models are trained 20 times. Each time a model is trained the accuracy is tested. The test is run 10 times with 1000 random samples of the entire training dataset each time. The average IOU score is taken from these 10 runs for the test. The average IOU scores are shown in the scatter plots in figures 5 and 6 for each representation. The final

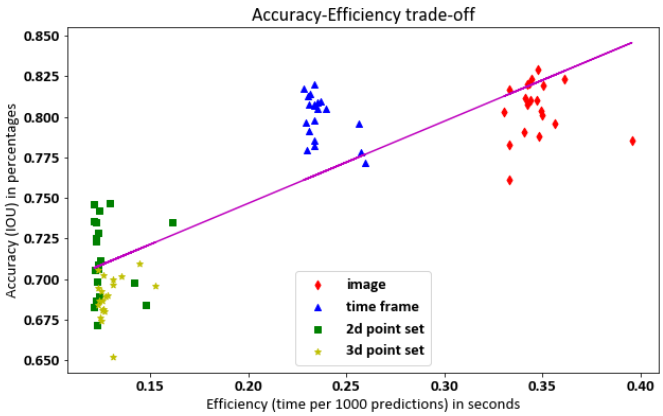

Figure 5: Results from training and testing the simple model 20 times for each representation. The trade-off for using event-based data is illustrated with the trend line. The accuracy achieved when using a time frame representation is on par with using images while the efficiency is much better.

average IOU score of the 20 tested models can be seen in tables 1 and 2. The reason for training a model 20 times is that the accuracy of a model is not deterministic.

When looking at the IOU scores in figure 5 of the simple neural network, the sparser event-based data representations show a lower score. As acknowledged in section 3, the model used is not optimized for sparse data representations and for this reason another model was made to test whether the accuracy scores can be improved.

The results of the sparse model are shown in figure 6. These results also show that event-based data representations are more efficient, however the difference in accuracy between all data representations is smaller. The trade-off between accuracy and efficiency is significantly better than for the simple model. The model has a high variance however in predictions, and is not made

for standard images.

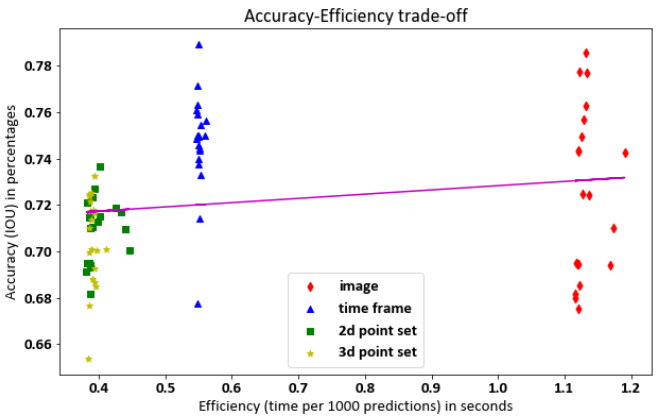

Figure 6: Results from training and testing the sparse model 20 times for each representation. The trade-off for using event-based data is better when looking at the trend line. However, the accuracy values of the predictions have a high variance and therefore are not reliable.

The accuracy-efficiency trade-off of both models is most noticeable between the image and the point sets. The models are more efficient with the point sets since these contain less data, however the predictions are also less accurate. The best event-based representation is the time frame, the accuracy is similar to the image-based representation and the efficiency is better. This finding is in line with the first hypothesis. For more information about the trade-off, refer to tables 1 and 2. In these tables the model training time is presented together with the average accuracy and efficiency score per representation. The model training times show a similar relation as the model prediction times.

4.2 Accuracy with YOLOv3

When using event-based data, the YOLOv3 model performed better over a longer time window. The difference between the results from running the model on the input data, is the accuracy of the model with respect to mean Average Precision (mAP) scores as shown in figure 7. The results consist of the accuracy scores of running the model on the 4 classes specified in section 3.2, and the scores of running the model on the entire dataset. For some examples of the predictions made with the model see appendix A.

In the results the model training times and prediction times are not used since these are equal over different inputs. For example, the logs from running the model show that the time needed for predicting both images and time frames, is about 1 ms on batches of 32 inputs. The only thing that differs in terms of efficiency is the data creation and preparation time. The data preparation time for an image to be converted to a correct input representation is roughly 0.3ms, and for a time frame roughly 0.04ms. Since these times are almost negligible compared to the data creation time, the total time of object detection depends primarily on how long it takes to capture an image or gather enough events.

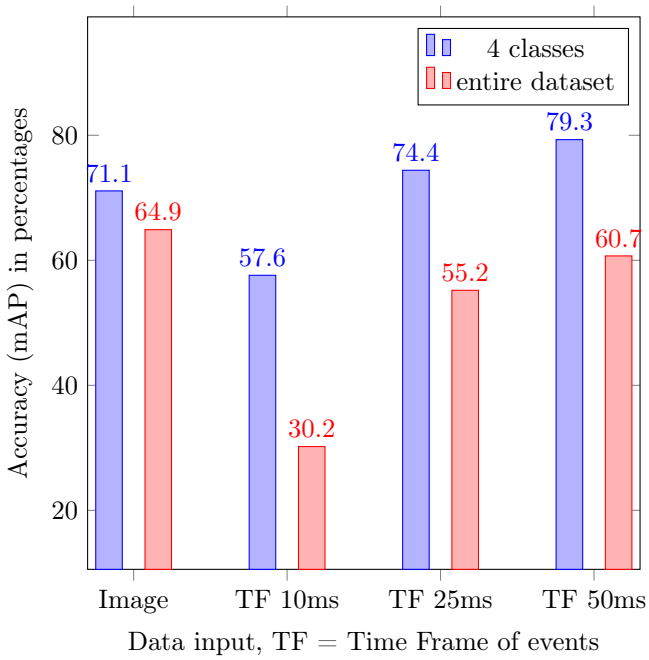

Figure 7: Results from training and testing the YOLOv3 model on different representations. When only 4 classes are used, event-based data can match and improve the accuracy of image-based data. For the entire dataset, the model achieves the best accuracy when trained on images.

When more events are used to create the time frame, the predictions of the model become more accurate. The limiting factor here is how many frames can be generated per second. Creating a time frame of 50ms means that a maximum of 20 frames can be generated and processed per second. For processing images, this depends on how many frames per second an RGB camera can produce. An interesting observation is that the time frame of 25ms already outperforms the image-based representation in terms of accuracy when using a dataset of 4 classes. When the model is trained on the entire dataset however the event-based data achieves a lower accuracy score than images. These results contradict each other, therefore the second hypothesis cannot be supported.

5 Responsible Research

The models used in this research are carefully designed to test the performance of object detection when using different input formats. The simple models are adapted from the model YOLE as described in section 3. And the YOLOv3 model is used directly from the source. However, there exists bias in this research: the sparse model that is used with dense data is an example of algorithmic bias. The result of this bias: the model performing

Table 1: Results of training and testing the model from figure 4. The data preparation time is the time taken on average to construct the dataset in the given representation. The model training time is the time taken on average to train the model over 40 epochs with the given representation. The prediction time is the time used to predict bounding boxes for 1000 random inputs on average. The IOU value is the accuracy of the predicted bounding boxes on average.

Representation	**Data preparation time**	**Model training time**	**Prediction time**	**IOU**
Image	6.9 s	31.46 s	0.35 s	80.5%
Time frame	23.0 s	26.07 s	0.24 s	80.0%
2D point cloud	9.5 s	15.46 s	0.13 s	71.3%
3D point cloud	9.9 s	16.19 s	0.13 s	68.9%

Table 2: Results of training and testing the model from figure 4 with sparse layers. For more information on the data, see table 1

Representation	**Model training time**	**Prediction time**	**IOU**
Image	412.20 s	1.13 s	72.5%
Time frame	39.84 s	0.55 s	74.7%
2D point cloud	25.39 s	0.40 s	70.9%
3D point cloud	26.30 s	0.39 s	70.4%

poorly, is still used as a comparison.

Another important ethical aspect of this research is whether the data is used responsibly. In this research two datasets are used, namely Caltech101 [11] and N-Caltech101 [12]. These datasets are processed by the models to obtain results about the model's performance in terms of accuracy and efficiency. The results, with respect to object detection, of the machine learning models are not used other than for demonstrative purposes. For the experiments that use a subset of the datasets, a sample bias is introduced. This bias could mean that the models predict poorly on data outside of this selection. For the purpose of this research the sample bias is not a problem, only the efficiency and accuracy of the model on the selected data is measured.

A different topic is whether the results of the research are useful and positive. Because negative results are still scientifically significant, the results from training the YOLOv3 model on the entire dataset, which are not in line with the hypothesis for that experiment, are used nonetheless. The model achieves a higher accuracy when using images as input than when using event-based data as input. This result is discussed in section 6.

The results from this research can be reproduced by following the method from section 3. The results in section 4 are obtained from running the models on the hardware specified in section 3. For different hardware the exact same timing data cannot be produced, however the relation between the results with respect to accuracy and efficiency will remain.

6 Discussion

From the first experiment the general trade-off for using events instead of images can be seen as losing accuracy for gaining efficiency. The results from the simple model nicely portray the accuracy-efficiency trade-off and show that event-based data can be preferred over image-based data when efficiency is of importance. When using the sparse model, event-based data even outperforms image-based data in terms of accuracy. However, this result has little meaning because the sparse model is not designed for dense image-based data. Furthermore, the sparse model is slower than the simple model and has lower accuracy scores. If a better model is constructed that can exploit the sparsity of events, the accuracy-efficiency trade-off will have more value for the conclusion.

The second experiment, described in section 3.3, uses a YOLOv3 model that is more optimized for object detection than the simple models. This model is however made for images, therefore it does not fully utilize the sparse and temporal nature of the events. Even though it is not optimized for event-based data, a time frame of events can perform better in terms of accuracy than standard images when using a small dataset. The reason for this is probably because images contain a lot of noise, which is less present in the time frame of events. Nonetheless, the model achieves a higher accuracy when using images instead of event-based data when it is trained on the entire dataset. A possible reason for this is that having colors in the input data is an advantage for object detection, the results from paper [7] mentioned in section 2 support this theory. Unfortunately, since the YOLOv3 model only accepts image files as input, the input data size does not change, and thus no change in efficiency of the model could be measured. The main findings from this experiment is that event-based data can achieve a similar accuracy as image-based data for the same amount of processing time.

Regarding the experiments, the data used is not representative for real life object detection. The simple models have an output layer that predicts the class label and the bounding box coordinates of a single object. The YOLOv3 model on the other hand can predict multiple objects, the model is not utilized fully since there was only one object in each data instance. The objects that the models are trained and tested on are from Caltech101, the objects are all centered and of similar size. Even though the data is not optimal, the correlation between efficiency and accuracy shown from testing this setup can be recreated using different data. The only comparison that is not made is comparing efficiency and accuracy of each data representation with the most optimal model for that representation.

The results achieved in this research are comparable with results achieved in other papers. For instance, the paper [3] mentioned in section 2 shows a model that is actually optimized for sparse event-based data. The model achieves a similar accuracy score with a much better efficiency score for using event-based data instead of images. In the paper [7] a similar conclusion can be made, although the difference in efficiency is smaller.

7 Conclusions and Future Work

The research question was made to test whether using sparser event-based data can be more efficient to train a model without losing a significant amount of accuracy. The question specifically asks what the trade-off is with respect to accuracy and efficiency for using event-based data.

The results discussed in section 6 show a clear relation between sparsity of data representations and efficiency of each model. The accuracy-efficiency trade-off for the simple models is favorable for using event-based data. The loss in accuracy is small compared to the gain in efficiency. The experiment with YOLOv3 shows that for the same efficiency of the model, a similar accuracy can be achieved by using event-based data. When an ideal model for event-based data is used, the accuracy-efficiency trade-off for using event-based data instead of image-based data is negligible.

In future research the best models with respect to each data representation should be used to compare the accuracy-efficiency trade-off. Furthermore, research can be done on the best possible event-based data representation and model for object detection. Finally, the accuracy-efficiency trade-off for using events with color values instead of regular events or images is interesting to research.

References

[1] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," 2019. [Online]. Available: https://arxiv.org/abs/1905.05055

[2] G. Gallego, T. DelbrÃŒck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, 2022.

[3] N. Messikommer, D. Gehrig, A. Loquercio, and D. Scaramuzza, "Event-based asynchronous sparse convolutional networks," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 415–431.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[5] M. Cannici, M. Ciccone, A. Romanoni, and M. Matteucci, "Asynchronous convolutional networks for object detection in neuromorphic cameras," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 1656–1665.

[6] D. Gehrig, A. Loquercio, K. Derpanis, and D. Scaramuzza, "End-to-end learning of representations for asynchronous event-based data," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 5632–5642.

[7] E. Perot, P. de Tournemire, D. Nitti, J. Masci, and A. Sironi, "Learning to detect objects with a 1 megapixel event camera," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS'20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[8] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2020, pp. 237–242.

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[10] Y. Yan, "Spconv: Spatially sparse convolution library," https://github.com/traveller59/spconv, 2021.

[11] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *2004 Conference on Computer Vision and Pattern Recognition Workshop*, 2004, pp. 178–178.

[12] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to

spiking neuromorphic datasets using saccades," *Frontiers in Neuroscience*, vol. 9, 2015. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2015.00437

[13] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018, cite arxiv:1804.02767Comment: Tech Report. [Online]. Available: http://arxiv.org/abs/1804.02767

A YOLOv3 prediction example

Figure 8: Prediction results, from left to right: image, time frame of 10ms, time frame of 25ms, time frame of 50ms.