Convolutional Neural Network for Detection and Classification with Event-based Data

Joubert Damien[1], Konik Hubert[3] and Chausse Frederic[2]
[1]*DEA-SAR, Groupe Renault, 1 Avenue du Golf, Guyancourt, France*
[2]*Universit Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, F-63000 Clermont-Ferrand, France*
[3]*Univ Lyon, UJM-Saint-Etienne, CNRS, Tlcom Saint-Etienne, Laboratoire Hubert Curien UMR 5516, F-42023, Saint-Etienne, France*

Keywords: Event-based Sensor, Convolutional Neural Network, SSD, Faster-RCNN, Transfer Learning.

Abstract: Mainly inspired by biological perception systems, event-based sensors provide data with many advantages such as timing precision, data compression and low energy consumption. In this work, it is analyzed how these data can be used to detect and classify cars, in the case of front camera automotive applications. The basic idea is to merge state of the art deep learning algorithms with event-based data integrated into artificial frames. When this preprocessing method is used in viewing purposes, it suggests that the shape of the targets can be extracted, but only when the relative speed is high enough between the camera and the targets. Event-based sensors seems to provide a more robust description of the target's trajectory than using conventional frames, the object only being described by its moving edges, and independently of lighting conditions. It is also highlighted how features trained on conventional greylevel images can be transferred to event-based data to efficiently detect car into pseudo images.

1 INTRODUCTION

Embedded applications still face major issues such as world perception, energy consumption, learning in uncontrolled environments, among many others. The recent progresses made by deep learning approaches seem promising at first sight as the performance of many systems have been dramatically increased. But when these systems have to be implemented in embedded hardware, in order to obtain a sufficient speed and energy consumption with lower computing power, some applications are no longer improved. Concerning issues of target detection and classification, the diversity of user cases make the validation and the comparison of systems difficult. Most of the time, some databases such as KITTI (Geiger et al., 2013) are shared to compare different algorithms. This neglects the sensor, and because of lighting conditions often make the system misbehave, performances measured on these databases is biased compared to these of the final system. Currently, the main solution is to increase the size of the database, but it does not guarantee that every scenarios are tackled. This issue can be addressed if the sensor used to acquire data is independent of the environment. Many conventional grey level sensors now support an high dynamic range (HDR) response, but most of the time this latter dynamically depends on the radiometry of the scene. Event-based sensors provide a response independent of the lighting conditions, then it must be investigated if features they provide can be efficiently used to describe targets robustly. Detection and classification in automotive scenarios is one of the most challenging issue nowadays, with the prospect of the autonomous vehicle. This paper investigates how event-based sensor can be used with state of the art deep learning techniques to detect cars, as proposed in a recent work (Chen, 2017). Our approach will be compared to it latter. Our main contributions are :

- An analysis of the dynamic of automotive scenarios and how it must influence the design of an event-based detector.
- An event-based data processing technique designed to build representative features, also used to train state of the art convolutional neural networks.
- An annotation of state of the art dataset to measure the performances of our method.

This paper is organized as follows : the first

Damien, J., Hubert, K. and Frederic, C.
Convolutional Neural Network for Detection and Classification with Event-based Data.
DOI: 10.5220/0007257002000208
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 200-208
ISBN: 978-989-758-354-4

section shows how event-based systems can increase detection and classification performances. The second section explains the process to adapt two state of the art convolutional neural networks to detect and classify data provided by an event-based sensor, and finally the third section presents the results and draws the conclusions.

2 EVENT-BASED SENSORS

2.1 Neuromorphic Systems

From decades, it is commonly accepted that numeric sensors and computing systems are mostly synchronous : cameras acquire images with a given framerate or computers process it with a given clock. One recently popularized embedded application is advanced driver assistance systems (ADAS), thanks to the development of autonomous driving prototypes. Today, the performance achieved by these systems is not sufficient enough. Every year, the computing power and the number of sensors (cameras, lidars, radars..) increases, but the human, fitted with "only" two eyes and a brain, both being less powerful and less energy consuming, is still better. The neuromorphic community tries to inspire from biology to build smarter systems with reasonable energy consumption. A fundamental difference is how data are processed over time. In nature, the information is carried asynchronously, and data bandwidth depends on the relevance of the data provided. Following these principles, some key achievements (Liu et al., 2015) have been made recently in the neuromorphism community, and it supposes that the scale of the industrialization will be reached in next years.

2.2 Dynamic Vision Sensors

Many researches have been driven in the medical field to understand and model how eyes acquire and transfer data to the brain. For example, cat eyes are sensitive to the gradient of the light stimulus, preprocessing light information with filters similar to Gabor ones (Hubel and Wiesel, 1962). The human eye approximately works alike: some ganglion cells process in several ways the light collected by rods and cones in order to extract spatio temporal patterns (Masland, 2012). Many attempts have been made to implement these functionalities into chips, some adding analog spatial filters at pixel level (Ruedi et al., 2003) while others analogically detect temporal contrast of light (Delbrück and Mead, 1989). A key achievement was realized with the DVS128 prototype (Lichtsteiner et al., 2008), which spread into many applications domains such as high speed tracking (Delbruck and Lang, 2013), trajectory estimation (Mueggler et al., 2015) and pattern detection and classification (Bichler et al., 2012). Each pixel belonging to this sensor has the ability to detect and timestamp temporal contrast variation according to its polarity, and to transfer this information outside the camera with a microsecond latency precision.

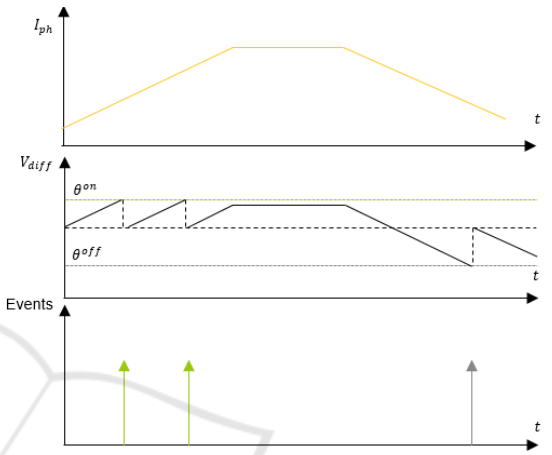


Figure 1: Event-based pixel process to subsample light signal.

As described in Fig. 1, pixels only react if the flux they collect will change enough at its surface. Sensors characteristics were analyzed (Lichtsteiner et al., 2008) (Posch and Matolin, 2011) and directly influence the quantification of the light signal. When a pixel detects a change, it generates a spike containing the change information (x, y, t, p), where (x, y) is the position in the imager, t the timestamp and p the polarity of the change. An example of artificial retina (or event-based) data is provided in Fig 2. A grey level measurement has been added to the next generation of artificial retina to provide more information.

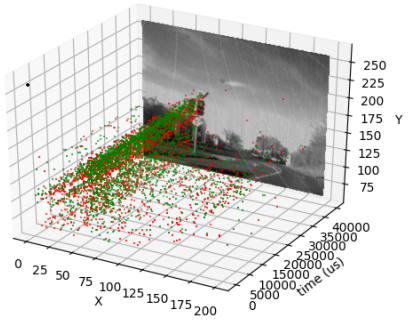


Figure 2: Events generated during 50 ms when the camera is approaching a road sign. Positive events are in red and negative in green.

Two different architectures were proposed: the DAVIS (Brandli et al., 2014) and the ATIS (Posch et al., 2011). Nowadays, these families of sensors benefit of new manufacturing process such as color filters (Li et al., 2015) and backside illuminated CMOS architecture (Son et al., 2017) (Taverni et al., 2018).

3 THE EVENT-BASED DETECTOR

Since the DVS128, many algorithms have been proposed to detect and track targets in many application domains. We address the issue of both classify and detect targets in the same algorithm.

3.1 Related Work

In this section the design of the event-based cars detector and classifier is explained. Such problems have been massively tackled in conventional grey level images. The first approaches consisted in extracting spatial features such as Haar wavelets (Viola et al., 2003), HOG (Dalal and Triggs, 2005) or SIFT (Lowe, 1999) based descriptor, and then applying a classifier in the feature space, for example a SVM (Scholkopf and Smola, 2001) or a boosting tree (Freund et al., 1996). The features computation and classification process is no more competitive against new deep learning techniques. Neural networks were studied in the end of the 90s, but recent improvements in computing capacities make them available for a large part of the research community in many fields of application. Deep learning mostly consists in training a deep network of artificial neurons backpropagating the error gradient through the layers. (LeCun et al., 1998). The network is characterized by its architecture and topology, but also with the cost function used in the training process to be adapted to a given task. Deep learning addressed image classification issues (LeCun et al., 1998) (Krizhevsky et al., 2012) (Zeiler and Fergus, 2014). In these convolutional neural networks (CNN) architectures, the first convolutional layers extract information with Gabor like features (Zeiler and Fergus, 2014), as works biological vision systems (Hubel and Wiesel, 1962). In the training process, both feature extraction and classification are optimized at the same time. CNN-based networks such as the VGG (Simonyan and Zisserman, 2014) and the Inception (Szegedy et al., 2016) model achieve state of the art results at classifying objects. But classification is not enough in automotive applications. The 3D or at least the 2D position of the target is also crucial as it determines how dangerous a scenario can be. Classification with a neural network is an expensive operation in computation time and it is not possible to run classification in a pyramid of multi scale images. New architectures were developed to both detect and classify. One part of the network is dedicated to object detection and the other part is dedicated to object classification. In the R-CNN architecture (Girshick et al., 2014), a region proposal network selects some relevant regions on which a classifier network is applied. This entire network is trained with a cost function taking into account both classification and detection errors. Many architectures have been proposed to decrease the execution time (Girshick, 2015) (Ren et al., 2015). In order to address the region proposal and classification at the same time, two network architectures are now popular : the SSD (Liu et al., 2016) and the YOLO (Redmon et al., 2016).

Each previous network is efficient with conventional images, but seems unfitted to use with artificial retina data at first sight because no frames are provided by DVS sensors. The task of classification in event based-data is an active field of research. Inspired by works on conventional images, retina data space can be projected into spatio temporal feature space. An interesting work led to descriptors inspired from HOG (Dalal and Triggs, 2005) : the HOTS (Lagorce et al., 2017) and its improvement, HATS (Sironi et al., 2018). Some neural network applications were adapted to process asynchronous retina data. Asynchronous networks have been developed in order to manage both spatial and temporal information. Biological neurons present a great diversity of transfer functions (Izhikevich, 2004), then most of the time a standard simplified leaky integrate and fire neuron model is implemented. It must be noticed that these Spiking Neural Networks (SNN) were studied before first artificial retina appeared, and that they achieve good performances at classifying and detecting spatio-temporal patterns (Masquelier et al., 2008). The main difficulty is to implement these networks and dedicated hardware, which are still at the prototype stage.

Another not bio-inspired method consists in building pseudo images with event-based data, and applying frame based approach on it (Moeys et al., 2016) (Maqueda et al., 2018). In the work of (Chen, 2017), a reduced version of the Yolo network is trained to achieve car detection and classification. Let us investigate how the preprocessing step must be elaborated in order to train efficiently CNNs.

3.2 Images of Event Activity

Our method does not use the precision of the temporal information encoded into events. We believe this information is not necessary here, as it could be in other applications such as spatiotemporal pattern detection and classification, for example gesture recognition (Amir et al., 2017). The purpose here is to tackle automotive scenarios where it is crucial to robustly detect incoming targets. In our approach, the speed of targets is constant during the detection stage, it is considered that no relevant information can be extracted from timing information to detect and classify. However, after the detection and classification, the timing information could be used to estimate the trajectory and potential time to collision. In automotive scenarios, the diversity of target relative speeds is important then several timing scales must be used. For example, concerning slow targets, it is not necessary to quickly detect them, but rather to precisely detect them and estimate their trajectory.

3.2.1 Integration of Events over Time

In order to track both fast and slow incoming targets, the activity of the sensor is integrated by using several temporal windows, as the opposite of the work of (Chen, 2017), where the integration lasts 10 ms in order to obtain results at 100 Hz. The number and the duration of the temporal windows are estimated depending on the application. In the work presented here, the constraints of automotive applications obliged us to process the event stream using three different temporal windows : one of 40 ms, t_{40}, one of 100 ms, t_{100}, and one of 200 ms, t_{200}. Detection are processed in the three pipelines in parallel, in order to detect fast, normal and slow targets. We suppose that the detection time is less than the shorter temporal window (40 ms) in order to start a new detection as soon as the last t_{40} is available. One up to three temporal windows are used to compute final detection. This latter is obtained taking the maximum of the detections realized in each temporal window. This behavior is illustrated in Figure 3 to sum up how data are processed in our method.

When a pixel detects a change, several events are generated depending on the slope of the change or on the parameters of the sensor, as illustrated in Fig. 1. Then targets detected in t_{200} may generate more events in one pixel compared to those of t_{40}. We chose to not count the number of events on each pixel, but rather to simply set the grey level to one if the last spike is positive, to zeros if it is negative, and 0.5 if nothing happened in the time window. Contrary to (Chen, 2017), we think that preserving the polarity

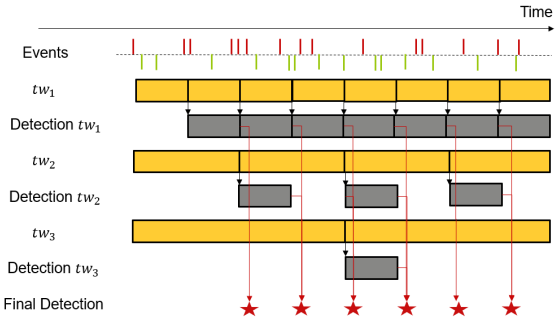


Figure 3: Schematic of the proposed detection architecture to both detect fast and slow targets. Time windows (in Yellow) and detections (in Grey) are obtained in parallel.

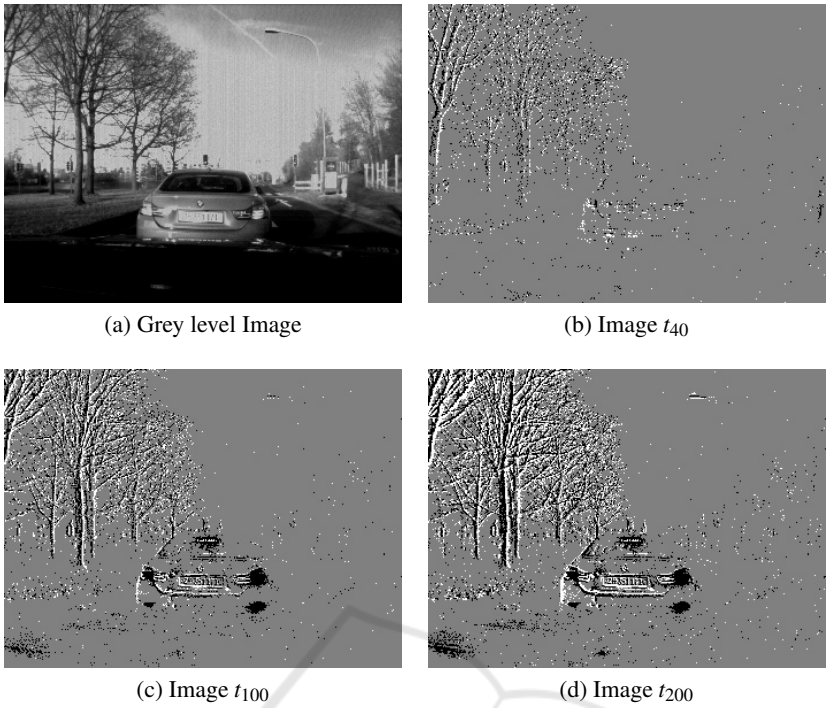
will help to detect a shape of car. Fig. 4 shows images of events considering two temporal windows.

3.2.2 Network Architecture

It is now investigated how these images can be used with standard CNN architectures. The well-known SSD (Liu et al., 2016) and Faster-RCNN (Ren et al., 2015) networks are here adapted to event-based data. Both architectures are kept identical. Concerning the SSD, 11 convolutional layers are trained, the first ones being initialized as the first layers of the VGG network. From 6 layers at different scale levels, features are extracted through a convolutional layer whose filters have a given spatial shape called anchor. These latter are linked to a softmax classifier to obtain the class: car or not car, and also adjust the final bounding box of the object. Then, for each anchor is trained $C+4$ filters where C is the number of classes. Concerning the Faster RCNN, the region proposal network (RPN) which proposed potential regions of interest is initialized with layers of VGG or ZF networks. After the RPN, a small network maps RPN features into a smaller feature space where two sliding fully connected layers predict both class and position of objects. A similar anchor process is used to generate the new feature space.

3.3 Datasets

When applications were developed using the DVS128 sensor, databases were needed to train and test algorithms. The first ones were small (Serrano-Gotarredona and Linares-Barranco, 2015) or artificial (Orchard et al., 2015). Larger databases are now available, but sometimes provided without annotations. For automotive applications, the DDD17 database (Binas et al., 2017) has been recently proposed, and contains 12 hours of acquisitions synchronized with several information of the car such as the steer-



(a) Grey level Image (b) Image t_{40}

(c) Image t_{100} (d) Image t_{200}

Figure 4: Image of events in different time windows for the same scene. In this case, where the speed between the two cars is low, only longer temporal window contains the shape of the car.

ing angle or the GPS position. Our first contribution was to annotate a part of this dataset in the same manner as (Chen, 2017). A dedicated network, different from the final detection network, is specially trained on grey level images from the DDD17 dataset. During initialization, a SSD network pre-trained on RBG images is applied on grey level images in order to keep only scenes containing cars to obtain a first set of annotation A_0.

Around each target detection, 500 ms of event data are kept in order to generate time windows. Detection are made every second of the videos acquired by day. We also annotated 800 grey level images by hand to build the database A_H. A first version of the annotation network was trained on $A_0 : W_0$ and in finetuning the initial multi-class SSD trained on RGB images on A_0. The network W_0 is then applied on the database to provide a new set of annotations which defines a new database A_1. The performance of the network W_0 is also evaluated on A_H. An iterative specialization process is then used to increase the performance of the annotation network. We iterate this process until the performance of the network into the database A_H stops to increase and get W_n. In practice, it takes 3 iterations to converge, and the final network does not detect each car in A_H, mostly because of lighting conditions (grey level images have a small dynamic range), and because of small targets. Using W_3 the final database is annotated. In this latter, no information is provided about the scenario, but the diversity of relative speeds seems not representative enough.

Another similar work has been proposed to build an annotated event-based automotive database (Sironi et al., 2018). Authors apply almost the same annotation process to pick relevant data from automotive scenarios, but they only provide 100 ms of event data in bounding boxes, half belonging to cars and half belonging to the environment. Also, grey level images are not provided. The aim of this database is to evaluate classification performances. In order to adapt it to measure detection and classification performances, we randomly positioned these boxes containing cars in images to generate a pseudo artificial database. This work has been conducted only for testing purposes. As our network is trained with real images, we ensure that cars are close to image sensor (in the range [20% , 80%] of image's height. Two images of 40 ms are built for each acquisition, then 8790 pseudo images contain cars and 8790 contains background.

4 EXPERIMENTS

The final detection networks R_{SDD}, R_{FZF} and R_{FVG} are trained on the labeled DDD17 database (Binas et al., 2017), using ZF or VGG weights at initializa-

tion for the first layers. These filters layers are commonly used on color images, and are effective to extract spatial patterns with high level of abstraction. For the SSD, the learning rate is set to 0.001 and exponentially decreases with a 0.97 factor. Instead of using a momentum solver as proposed in the original SSD architecture, an Adam solver (Kingma and Ba, 2014) was preferred as it does not diverge at the beginning with the same learning rate. Using Adam, $\varepsilon = 0.1$, and the weight decay is $5e^{-5}$. Data augmentation techniques were not used during the training. The Faster-RCNN inspired network uses the same training processes, as they both converge quickly and efficiently. We decided to only train the network for car detection, the resolution of DDD17 image being approximately a quarter VGA ((346*260 pixels) we estimated that there are not enough information to separate car from truck classes.

4.1 DDD17

The final database contains 120k images for training and validation, and 15k images for testing. One main drawback of our annotation method is that it also annotates event-based data when there is no relative motion between targets and camera. In this case, there is almost no event data in each time window.

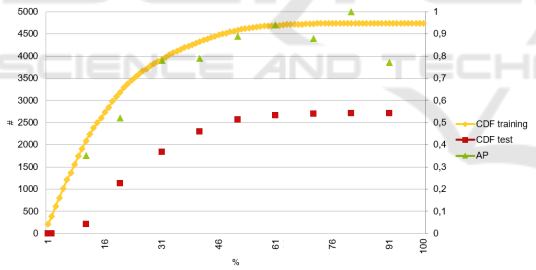


Figure 5: Integration of cumulative distribution function (CDF) of the percentage of spiking pixels into the bounding boxes for the training and testing database. Average precision of the SSD(15 %) on the testing database according to the activity is also computed: if activity is equal to x %, AP is computed for BB whose activity is inside $]x-10\%;x]$.

Figure 5 reveals that most of the pseudo images do not contain significant information: 75 % of bounding boxes contains less than 35 % of spiking pixels inside. In theory, 35% pixels spiking in the box may represent a 35 % occluded target. We trained the network with bounding boxes containing more than 35% or 15 % spiking pixels. This method decreases the number of images in the training database. At the end, 9000 training and 2000 testing images remain considering bounding boxes with more than 15 % spiking pixels. Considering bounding boxes with more than 35 % spiking pixels, it remains 7000 training and 1500 testing images. Network are trained on both datasets. The average precision (AP) is computed for each network using the VOC 2007 standard, and is presented in Table 1. These results not show a significant performance difference between the two databases concerning the SSD network. Thus, R_{FZF} and R_{FVG} have not been trained on bounding box containing more than 35 % spiking pixels. The network using the Faster-RCNN architecture with the ZF weights provide best AP results, but as illustrated in Figure 6, it also generates more false detections. However, if a target generating 14% of spikes inside its bounding box is detected, it is considered as a false positive in the previous results. The influence of the activity on AP is analyzed in Fig. 5 and seems to evolve similarly to the CDF.

Table 1: Average precision of CNN-inspired networks on two DDD17 testing databases with different number of active pixels in bounding boxes.

$R_{SSD}(35\ \%)$	0.693	$R_{FZF}(15\ \%)$	**0.834**
$R_{SSD}(15\ \%)$	0.698	$R_{FVG}(15\ \%)$	0.846

4.2 PropheseeNcar

The database is generated as explained in previous section. SSD results must be filtered to address only classification. Bounding boxes whose intersection with the data from Prophesee is more than 50% are kept. In Table 2 are compared the accuracy: the $R_{SSD}(15\ \%)$ does not perform the best result on this database, but this network was not specialized to this database. Being generated with a different sensor, using different settings, the average precision result obtained on this pseudo dataset is almost identical (0.68) to the average precision obtained on the DV17 database. This indicates that event-based sensors provide an object representation independent of the sensor, which confirms the assumption that event-based data can provide more robust and generic target models. A more sensitive sensor will be more robust to low contrast scenes, but the obtained feature representation would be the same using our preprocessing method.

Table 2: Accuracy of the proposed networks versus state of the art classifiers on the PropheseeNCar database.

R_{SSD}	0.746	HOTS	0.561
H-first	0.683	HATS	**0.902**

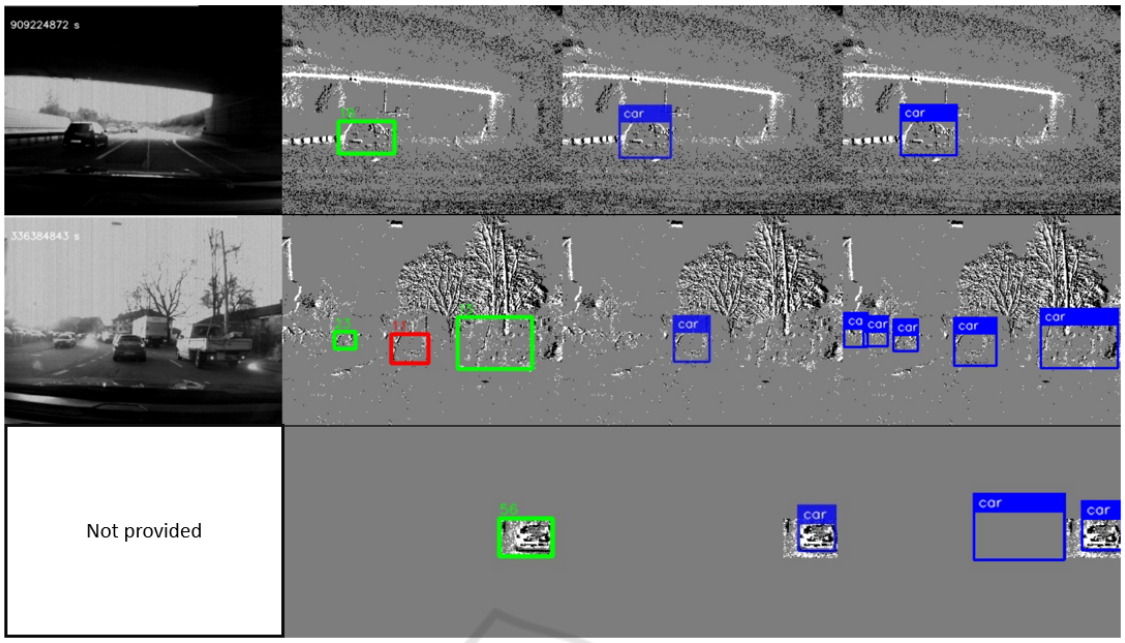


Figure 6: The first column contains grey level images, and the second one draws the ground truth labels on t_{40} images. The red boxes contain less than 15 % of event activity, while in green boxes more than 15 %. Some testing results obtained with R_{SSD}(15 %) are displayed in the third column while the fourth columns contained results from R_{FZF}(15 %). The two first lines correspond to the DDD17 database, while the last lines corresponds to the PropheseeNCar database. Further results are available online https://youtu.be/r1GqjE0jz58.

5 DISCUSSION

In the evaluation of our networks, the ability to solve every automotive scenarios is not really estimated. In order to measure the performance of the network regarding to the relative speed of targets, it should be measured performance regarding to relative speed. This work will be conducted in a future work by realizing tests on EURO NCAP user cases.

Many layers were used to process the information into the networks, and it may be studied if a more reasonable number of parameters can be used to achieve detection and classification task at a same performance level. Finally, the main drawback of our approach is to determine when the sensor is providing enough information to realize classification. The true advantage of event-based sensor, the time information, is neglected in our work, and the pseudo images could also be obtained by computing the difference between two conventional frames, and applying a threshold.

6 CONCLUSION

An adaptation of the SSD and the Faster-RCNN architectures has been proposed in order to process event-based data for automotive scenarios. The data of a recent published database acquired with a state of the art silicon retina is labelled with an iterative process. Automotive constraints influence the way data are packaged before to be processed by the network, in order to maintain a sufficient level of performance for each relative speed between the camera and the targets. Detection performances are measured and confirm results of how transferable are CNN filters to event-based pseudo images. It has also been highlighted that the spiking flow must be carefully analyzed to process the classification with enough data.

REFERENCES

Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., Nayak, T., Andreopoulos, A., Garreau, G., Mendoza, M., et al. (2017). A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7243–7252.

Bichler, O., Querlioz, D., Thorpe, S. J., Bourgoin, J.-P., and Gamrat, C. (2012). Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity. *Neural Networks*, 32:339–348.

Binas, J., Neil, D., Liu, S.-C., and Delbruck, T. (2017). Ddd17: End-to-end davis driving dataset. *arXiv preprint arXiv:1711.01458*.

Brandli, C., Berner, R., Yang, M., Liu, S.-C., and Delbruck, T. (2014). A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341.

Chen, N. F. (2017). Pseudo-labels for supervised learning on event-based data. *arXiv preprint arXiv:1709.09323*.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE.

Delbruck, T. and Lang, M. (2013). Robotic goalie with 3 ms reaction time at 4% cpu load using event-based dynamic vision sensor. *Frontiers in neuroscience*, 7:223.

Delbrück, T. and Mead, C. (1989). An electronic photoreceptor sensitive to small changes in intensity. In *Advances in neural information processing systems*, pages 720–727.

Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *Icml*, volume 96, pages 148–156. Bari, Italy.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237.

Girshick, R. (2015). Fast r-cnn. *arXiv preprint arXiv:1504.08083*.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154.

Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? *IEEE transactions on neural networks*, 15(5):1063–1070.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Lagorce, X., Orchard, G., Galluppi, F., Shi, B. E., and Benosman, R. B. (2017). Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1346–1359.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, C., Brandli, C., Berner, R., Liu, H., Yang, M., Liu, S.-C., and Delbruck, T. (2015). An rgbw color vga rolling and global shutter dynamic and active-pixel vision sensor.

Lichtsteiner, P., Posch, C., and Delbruck, T. (2008). A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576.

Liu, S.-C., Delbruck, T., Indiveri, G., Douglas, R., and Whatley, A. (2015). *Event-based neuromorphic systems*. John Wiley & Sons.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee.

Maqueda, A. I., Loquercio, A., Gallego, G., Garcıa, N., and Scaramuzza, D. (2018). 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.

Masland, R. H. (2012). The neuronal organization of the retina. *Neuron*, 76(2):266–280.

Masquelier, T., Guyonneau, R., and Thorpe, S. J. (2008). Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. *PloS one*, 3(1):e1377.

Moeys, D. P., Corradi, F., Kerr, E., Vance, P., Das, G., Neil, D., Kerr, D., and Delbrück, T. (2016). Steering a predator robot using a mixed frame/event-driven convolutional neural network. In *Event-based Control, Communication, and Signal Processing (EBCCSP), 2016 Second International Conference on*, pages 1–8. IEEE.

Mueggler, E., Gallego, G., and Scaramuzza, D. (2015). Continuous-time trajectory estimation for event-based vision sensors. In *Robotics: Science and Systems XI*, number EPFL-CONF-214686.

Orchard, G., Jayawant, A., Cohen, G. K., and Thakor, N. (2015). Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437.

Posch, C. and Matolin, D. (2011). Sensitivity and uniformity of a 0.18μm cmos temporal contrast pixel array. In *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, pages 1572–1575. IEEE.

Posch, C., Matolin, D., and Wohlgenannt, R. (2011). A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds. *IEEE Journal of Solid-State Circuits*, 46(1):259–275.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Ruedi, P.-F., Heim, P., Kaess, F., Grenet, E., Heitger, F., Burgi, P.-Y., Gyger, S., and Nussbaum, P. (2003). A 128/spl times/128 pixel 120-db dynamic-range vision-sensor chip for image contrast and orientation extraction. *IEEE Journal of Solid-State Circuits*, 38(12):2325–2333.

Scholkopf, B. and Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Serrano-Gotarredona, T. and Linares-Barranco, B. (2015). Poker-dvs and mnist-dvs. their history, how they were made, and other details. *Frontiers in neuroscience*, 9:481.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sironi, A., Brambilla, M., Bourdis, N., Lagorce, X., and Benosman, R. (2018). 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.

Son, B., Suh, Y., Kim, S., Jung, H., Kim, J.-S., Shin, C., Park, K., Lee, K., Park, J., Woo, J., et al. (2017). 4.1 a 640× 480 dynamic vision sensor with a 9μm pixel and 300meps address-event representation. In *Solid-State Circuits Conference (ISSCC), 2017 IEEE International*, pages 66–67. IEEE.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

Taverni, G., Moeys, D. P., Li, C., Cavaco, C., Motsnyi, V., Bello, D. S. S., and Delbruck, T. (2018). Front and back illuminated dynamic and active pixel vision sensors comparison. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 65(5):677–681.

Viola, P., Jones, M. J., and Snow, D. (2003). Detecting pedestrians using patterns of motion and appearance. In *null*, page 734. IEEE.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.