Grayscale And Event-Based Sensor Fusion for Robust Steering Prediction for Self-Driving Cars

Luca Pascarella
ETH Zürich
Zürich, Switzerland
luca.pascarella@pbl.ee.ethz.ch

Michele Magno
ETH Zürich
Zürich, Switzerland
michele.magno@pbl.ee.ethz.ch

2023 IEEE Sensors Applications Symposium (SAS) | 979-8-3503-2307-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/SAS58821.2023.10254055

***Abstract*—Event-based vision, led by a dynamic vision sensor (DVS), is a bio-inspired vision model that leverages timestamped pixel-level brightness changes of non-static scenes. Thus, DVS's architecture captures the dynamics of a scene and filters static information out. Although machine learning algorithms based on DVS inputs overcome active pixel sensors (APS), they still struggle in challenging conditions. For example, DVS-based models outperform APS-based ones in high-dynamic scenes but suffer in static landscapes. In this paper, we present GEFU (Grayscale and Event-based FUsor), an approach that opens to sensor fusion by combining grayscale and event-based inputs. In particular, we evaluate GEFU's performance on a practical task: predicting a vehicle's steering angle in a realistic driving condition. GEFU is built on top of a consolidated convolutional neural network and trained with realistic driving conditions. Our approach outperforms solo DVS- or APS-based models on non-trivial driving cases, such as the static scenes for the former and the suboptimal light exposure for the latter approach. Our results show that GEFU (i) reduces the root-mean-squared error to $\sim 2^{\circ}$ and (ii) although the magnitude of the steering angle does not always match the ground truth, the steering direction left/right is always predicted correctly.**

***Index Terms*—Sensor Vision, Machine Learning, Sensor Fusion**

I. Introduction

Event-based visions are biologically inspired vision algorithms driven by continuous scene changes. Dynamic vision sensor (DVS) [1]–[3] implements in hardware high-speed event generation acting as a high-pass filter that filters static and consequently redundant information out from a scene. Although both active pixel sensors (APS) and DVS share a similar construction architecture based on an active matrix of photodetectors, they have different output formats. Typically, APS returns a matrix of pixels' intensity at a constant rate. On the contrary, DVS generates an independent response to brightness changes for each pixel containing timing, intensity, and matrix-level position. Over frame-based sensors (*e.g.,* APS), DVS presents non-trivial advantages such as high temporal response (microseconds *vs.* milliseconds), high dynamic range (140dB *vs.* 60dB), reduced bandwidth, and low power [4]–[6].

Due to the intrinsic characteristics, event-based vision algorithms naturally behave as motion detectors by focusing on moving edges while ignoring static regions of the scene. Thus, event-based vision algorithms perfectly fulfill tasks in which the scene changes continuously, such as object tracking [3], [7], 3D reconstruction [8], [9], motion segementation [10], etc. In that regard, one of the first event-driven practical applications comes from Lee *et al.* [11], [12], which exploited the output of a DVS to feed a machine learning algorithm that recognizes non-stationary gestures in real-time.

By exploiting this approach, Maqueda *et al.* [13] used an enhanced commercial DVS camera to evaluate to what extent a consolidated frame-based vision algorithm fed with reconstructed frames (*e.g.,* the frames have been reconstructed by accumulating DVS events in a given time window), overcomes traditional grayscale inputs on self-driving cars. Specifically, the authors proposed a deep learning algorithm based on convolutional neural networks (CNN) [14] to exploit the natural response of DVS to scene motion and return a real number indicating the steering angle of a self-driving car. Maqueda *et al.* [13] reported that DVS-based deep learning models overcome grayscale vision algorithms in challenging scenarios where grayscale sensors fail due to adverse weather conditions, suboptimal illumination, and fast motion.

Although opening the potential of event cameras on a challenging motion-estimation task, the approach proposed by Maqueda *et al.* [13] overlooks a fully realistic usage. Indeed, none of the two variants (*i.e.,* solo grayscale or solo event-based) is immune to all road conditions. For example, the solo event-based approach overcomes the grayscale in adverse weather conditions due to a lack of brightness, but it hangs in stationary conditions due to the absence of events.

In this paper, we presented GEFU (**G**rayscale and **E**vent-based **FU**sor) an approach built on top of a consolidated frame-based CNN model [14]. The tool aimed to improve the frame-based model's robustness by exploiting both the advantages of grayscale and event-based sensors. In particular, we created two different network configurations to evaluate the robustness of GEFU in adverse ambient conditions such as suboptimal illumination and non-moving scenes.

Significance of research contribution. The proposed approach represents a step toward the design of robust self-moving vehicles. Our experiment shows that GEFU can extend previous results by integrating complementary inputs (*i.e.,* grayscale and event-based) into the continuous domain (*i.e.,* steering angle) with a limited performance impact. It is worth noticing that GEFU focuses on the robustness of frame-based algorithms when exploiting sensor fusion.

II. Related Work

We focus our discussion on (i) event-based sensors and (ii) deep-learning approaches for DVS. Due to space limitations, we omit to discuss the many applications of DVS, pointing the reader to an inclusive survey by Gallego *et al.* [15].

A. Event-Based Sensors

Frame-based sensors (*e.g.,* APS) are designed to return brightness information in a full square matrix of pixels at a constant interval known as frame rate (*e.g.,* 60 fps). On the contrary, event-based sensors, such as DVS [16]–[19], generate a stream of events in response to a brightness change in the observed scene. The events have the property of decoupling pixels' information by indicating only the pixel affected by the brightness variation. In other words, the generated events are independent of each other and assume the form of planer coordinates followed by a timestamp and brightness polarity (positive and negative). Thus the throughput is variable (up to 450 MEPS [20]) and is proportional to the amount of captured events. This encoding takes inspiration from the spiking nature of biological cells as described initially by Lichtsteiner [16] and acts as a filter that automatically suppresses static (*i.e.,* non-dynamic) information such as not moving scenes. Since the amount of data generated is dynamic, DVS can be optimized for low-power or high-throughput applications [5], [21], [22]. In short, DVS presents advantages regarding *high dynamic range* (HDR), up to 140dB, that allows for capturing bright stimuli even in dark environments [3], [23], *low-power* impact $<$10mW due to the absence of redundant information [4], and *low-latency* response, $<$10µs, because each pixel works independently [23]–[25].

B. Deep-Learning Approaches for DVS Input

For their intrinsic nature, event-based sensors produce asynchronous events (*e.g.,* timestamped pixel-wise brightness) as an irregular, over time, stream of coordinates. Due to this characteristic, two main strategies are commonly used to feed deep-learning models (i) rely on innovative models such as spike neural networks (SNN) [26] or (ii) perform a data preprocessing such as the frame-conversion based on fixed time-windows used by Lungu *et al.* [27]. By using the latter preprocessing approach, Lungu *et al.* [27] leveraged event-based inputs to train conventional frame-based CNN models for their rock-paper-scissors player, Orchard *et al.* [26] advanced an object recognition able to classify up to 36 characters, and Moeys *et al.* [28] combined both APS and DVS to produce a four-classes detector. As opposed to the previous classification models, Maqueda *et al.* [13] targeted a continuous estimation problem: Steering angle for self-driving cars. In other words, the downstream output is a continuous value rather than a predefined class. Our work continues on a regression problem as done by Maqueda *et al.* [13] by overlooking the limitations of single input (*i.e.,* APS or DVS) algorithms. In particular, our approach aims at evaluating sensor fusion as done by Moeys *et al.* [28] for their classification approach but applied to a continuous domain: Steering angle prediction.

III. Methodology

The *goal* of our study is to empirically assess the robustness of frame-based computer vision algorithms in realistic conditions, such as predicting the steering angle for autonomous vehicles, specifically for self-driving cars, when mixing multiple inputs. The *context* is represented by the data collected with a DAVIS [2] sensor mounted on the windshield of a vehicle driven by humans in realistic driving conditions. The study aims at answering the following research questions (RQs)

RQ_1: *What is the performance of* GEFU *when evaluated in a more realistic scenario?* With RQ_1 we aim at assessing the performance of GEFU in predicting the steering angle by considering a realistic driving condition, *i.e.,* we do not exclude challenging cases from the testing dataset.

RQ_2: *To what extent does sensor fusion overcome single-input approach?* With RQ_2 we aim at evaluating the robustness of GEFU against adverse driving conditions when fusing diverse data sensors *i.e.,* grayscale and DVS.

Figure 1 depicts the block diagram of the proposed approach. In particular, (A) illustrates the pre-processing steps adopted to create both training and testing datasets from the publically available DAVIS Driving Dataset 2017 (DDD17) [29]. (B) shows the differences between the two frame-based CNN models proposed in our approach to process single and multiple inputs.

A. Training and Testing Datasets Selection

To have a fair and repeatable evaluation, we train and validate our deep-learning model using the publicly available DAVIS Driving Dataset 2017 (DDD17) [29] as previously done by Maqueda *et al.* [13] that also represents our baseline. DDD17 contains approximately 12 hours of realistic vehicle driving ($\sim$400 GB) in different weather, road, and illumination conditions. The events and grayscale frames have been acquired with a DAVIS [2] sensor that includes both a grayscale camera and a DVS on the same pixel array (346×260 pixel resolution). To have a more realistic driving condition, the DAVIS has been mounted on the windscreen of a Ford Mondeo, framing the frontal street. Besides events and grayscale frames, the acquired dataset includes car data (*e.g.,* steering wheel angle, accelerator pedal position, brake pedal status, etc.) captured by connecting a custom recording system to the OBDII port of the vehicle. To have non-overlapping training and testing datasets, we implemented the same approach used by Maqueda *et al.* [13]. First, we split the DDD17 recording into one-minute sequences. Then, per each minute, we took the first 40 seconds for building the training dataset and the remaining 20 seconds for the testing dataset. This semi-random approach helps reduce the over-optimistic estimation [30] (*i.e.,* during normal driving, subsequent frames retain almost the same steering angle).

B. Event-to-Frame conversion

Consolidated vision algorithms are designed to process the input in the form of frames (*e.g.,* a 2D matrix of pixels intensity), such as the case for APS cameras. On the contrary,

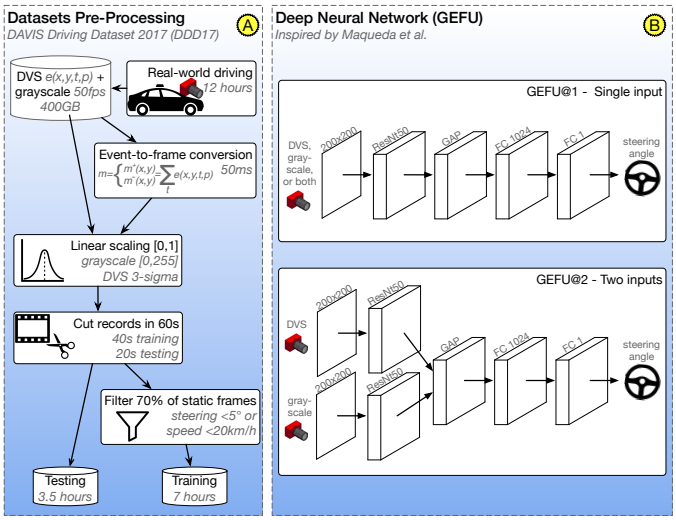


Fig. 1: Block diagram of the proposed approach. (A) describes the pre-processing steps used to create training and testing datasets from grayscale and DVS-based frames. (B) highlights the architectural variations of our frame-based models.

DVS cameras by generating decoupled events in response to a brightness change, cannot be used without applying a pre-processing step. Indeed, to take advantage of frame-based conventional algorithms, asynchronous and sparse inputs, such as the case of events for DVS, need to be converted into consecutive frames at a constant rate. To this aim, we adopted the demonstrative approach proposed in DDD17 [29] that accumulates the DVS events into pixel-wise time windows of a predefined length. The result is a stream (at a constant rate) of a 2D matrix of pixels' intensity. Specifically, for our approach, we accumulated in a 2D matrix m the events $e_k = (x_k, y_k, t_k, p_k)$ corresponding to all brightness changes happening in the constant interval time T. It is worth noticing that to prevent pixel-wise information loss due to the accumulation of both positive and negative brightness changes in the same time window, we accumulated positive and negative events into two different 2D matrixes, namely m^+ and m^-. In other words, for every accumulation interval T, we generate a 2-channel frame that results in the sum per pixel coordinates of all positive (or negative) events.

C. Datasets Pre-Processing

In a self-driving task, the challenge is to predict the correct steering angle of the wheels. Nonetheless, turns and overtakes are less frequent than straight lines. DDD17 shows a high unbalanced toward small, or even zero, steering angles. Moreover, there are conditions in which the vehicle is stationary due to, for example, a traffic light. To balance training data, we followed the approach of Maqueda *et al.* [13]. In particular, we removed from the training dataset $\sim$70% of frames corresponding to a steering angle $< 5°$ or a vehicle velocity $< 20km/h$. On the contrary, the testing dataset is left untouched for recreating as much realistic evaluation as possible. In addition, to accommodate a smooth learning process [31] and prevent unstable learning caused by weight values changing dramatically, the input values of both grayscale and event-based converted (see Section III-B) frames have been linearly scaled into the range $[0, 1]$. Specifically, for the grayscale frames, we linearly scaled the brightness from the original range $[0, 255]$ generated by the grayscale sensor to $[0, 1]$. For the DVS, we need to consider that each matrix's value represents the number of positive (or negative) brightness variations caught by the DVS per pixel in the given interval of time. Thus, a pixel-wise value does not have a predefined upper bound, such as in the grayscale sensors. Nonetheless, it remains proportional to the scene's dynamicity. For example, a fast-moving subject in the framed scene will generate more events. To scale the input in the range $[0, 1]$, we considered first the distribution of all positive and negative events across all frames, and then we applied a 3-sigma scaling. Finally, to diversify further the scenes seen by the model during the training (*e.g.,* low-speed moving with a constant steering angle), we applied a data augmentation technique that includes simple image transformations such as image scaling, rotation, translation, etc. [32].

D. Deep Neural Network Architecture

To achieve our goal, we started from the best-performing model proposed by Maqueda *et al.* [13]: A custom neural network composed of a single ResNet50, a global average pooling (GAP), and two fully connected (FC) layers. In particular, for single frame input, we created GEFU@1 by attaching to the ResNet50 a GAP layer followed by a fully connected 1024-dimensional layer, followed by a ReLU non-linear unit, and the final fully connected 1-dimensional layer. While to accommodate multiple frame-based inputs, we created GEFU@2 by including an additional ResNet50 before the GAP layer. In this way, GEFU@2 accepts two frame-based inputs fed with actual grayscale and DVS-derived frames (see Section III-B and Section III-C). GEFU@2 is specifically designed to merge the encoded image features of the two ResNet50 layers at the GAP stage rather than mixing grayscale and DVS-derived frames as a unified input of a single ResNet50 layer. The latter approach is justified because we already experimented with GEFU@1 which combines both grayscale and event-based information in a single 3-channel frame.

In both cases, the network output is a predicted real number scaled to match the actual wheels' steering angle. For both GEFU@1 and GEFU@2, we trained the models to optimize the loss function designed to minimize the error between the predicted output and the referring steering angle. Although a so-defined network is no longer state-of-the-art (*e.g.,* Vision Transformer may overcome ResNet50 at the cost of a complex architecture [33]), this network still represents a valid compromise in terms of architecture simplicity, training resources, and overall performance. Moreover, the Keras [34] implementation in the form of a simplified baseline for grayscale and event-based frames [35] allowed us to not start from scratch and focus more on our ultimate goal.

E. Data Collection and Analysis

We assess the accuracy of the predictions generated by each model as a regression problem. In other words, we could not count the number of instances correctly predicted; rather, we rely on the difference between the ground-truth and the prediction to estimate the model's goodness. Thus, for each model, we calculated two common metrics used to evaluate the performance of regression tasks: the root-mean-squared error (RMSE) [36] and the proportion of variation in the predicted values concerning the observed values (EVA) [13]. On top of this quantitative analysis, we also performed a qualitative analysis to better understand the strengths and weaknesses of GEFU. We manually inspected a set of "wrong predictions" *e.g.,* predictions in which the model is steering in the exact opposite direction. This means that the model may be correct in magnitude but wrong with the sign (*i.e.,* the steering angle is opposite to the ground-truth).

IV. Results

To answer RQ_1 and RQ_2, we run GEFU@1 and GEFU@2 using the datasets described in Section III. Thus, we created three single-input experiments (namely *APS*, *DVS*, and *CMB*) and one double-input experiment (namely *DBL*) as described in Section IV-A. Then, we performed quantitative and qualitative analyses to evaluate the characteristics of the best-performing model.

A. Setup of Experiments

The input of a CNN is usually a dense matrix that coincides with the shape of the first layer of the network, *i.e.,* 200×200 for GEFU. This approach work for frame-based inputs, such as the case of frames generated at a constant rate by APS. In our approach, the input generated by the DAVIS [2] sensor is both a grayscale matrix of 346×260 pixels generated at fix rate and a stream of timestamped events generated by the event-base camera. Thus, we needed to adapt these sources to our model. For the grayscale case, we centrally cropped the image to a square size of 200×200 pixels. For the event-based case, instead, we followed the demonstrative approach of DDD17 [29], accumulating the events in regular time windows (50ms for our best-performing model). This process allowed us to create frames at a fixed rate from the stream of timestamped events (see Section III-B). Moreover, the grayscale output has only a single value per pixel (*e.g.,* the corresponding grayscale brightness value), resulting in a single-channel matrix of $1 \times 200 \times 200$ pixels. Instead, the event-based frames created by accumulating positive and negative events falling in constant time windows resulted in a two-channel matrix of $2 \times 200 \times 200$ pixels. With these inputs, we defined a total of four experiments, three of which are based on a single-frame input and one uses a double-frame input as summarized in Table I.

Nonetheless, to comply with the ResNet50 input shape, we always adapt our inputs to recreate a three-channel image ($3 \times 200 \times 200$). To this aim, we applied three different strategies to the inputs of GEFU@1. For the solo grayscale case (*APS*), we triplicated the single-channel matrix to create a three-channel image. For the event-based case (*DVS*), we added an empty (zero-filled) channel next to the positive and negative channels derived from the event accumulation. And, for the combined grayscale plus DVS case (*CMB*), we used a full three-channel image where two channels come from event-based frames (positive and negative) and the third channel comes from the grayscale input. Finally, for GEFU@2 that expects two times $3 \times 200 \times 200$ images, we combined the previously discussed single input strategies. In other words, for the grayscale input, we triplicated the single-channel matrix and for the event-based input, we filled the third not used image channel with zeros. This resulted in two images of $3 \times 200 \times 200$ each. By combining these inputs with the two CNN architectures we obtained four different experiments, namely GEFU@1 *(APS)*, GEFU@1 *(DVS)*, GEFU@1 *(DVS)*, and GEFU@2 *(DBL)*.

TABLE I: Combination of GEFU and DDD17 derived dataset to define our four different experiments.

Model	Input	Description
GEFU@1	(*APS*)	Uses only grayscale frames
GEFU@1	(*DVS*)	Uses only event-based frames
GEFU@1	(*CMB*)	Combines grayscale and event-based frames in a single 3-channel frame
GEFU@2	(*DBL*)	Divide grayscale and event-based in separated input frames

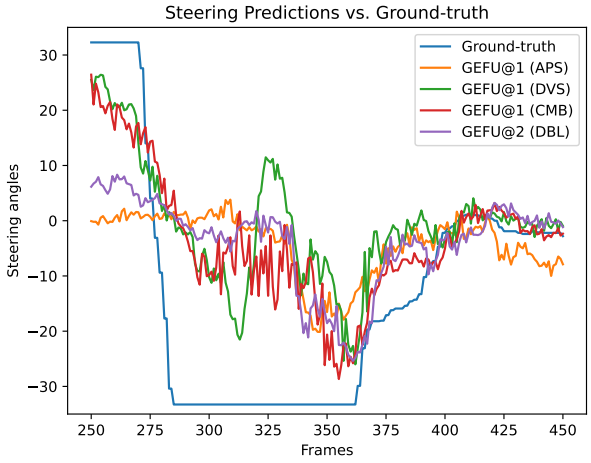
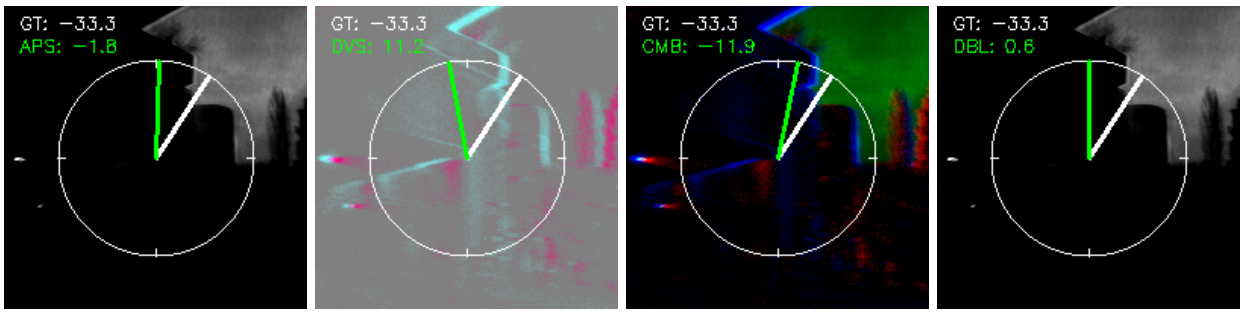


Fig. 2: Predictions of GEFU for a representative sample of 200 consecutive frames starting after 12.5s of the recording.

B. Quantitative and Qualitative Analysis

Figure 2 shows a portion of the steering prediction when GEFU is evaluated on the test dataset. For the sake of space limitation, this interval highlights only 200 consecutive frames starting at frame 250. By comparing the predictions to the ground-truth (blue line) it is clear how the steering angle, as well as the direction is not always correctly predicted, as confirmed by the RMSE value in Table II. Indeed, Figure 2 helps to clarify to what extent the model is missing the



(a) GEFU@1 (*APS*) (b) GEFU@1 (*DVS*) (c) GEFU@1 (*CMB*) (d) GEFU@2 (*DBL*)

Fig. 3: Example of steering angle prediction in suboptimal light conditions (frame 327). (a) shows a right turn misprediction when GEFU@1 relies only on grayscale inputs. Although the high dynamic of DVS allows for capturing more details about the foreground building, GEFU@1 in (b) still struggles to predict the right turn. (c) is the optimal case where the prediction error is minimal while still keeping the same architecture complexity. Finally, (d) shows the case where neither a more complex architecture outperforms GEFU@1 when mixing different sources of input.

TABLE II: Comparison of the best-performing models as previously reported by Maqueda *et al.* [13] and our best performing approaches.

Approach	Input	EVA	RMSE
Bojarski *et al.* [37]	grayscale	0.16	9.02°
Xu *et al.* [38]	grayscale	0.30	8.19°
Maqueda *et al.* [13]	event-based	0.82	4.10°
GEFU@1 (*APS*)	grayscale	0.47	3.66°
GEFU@1 (*DVS*)	event-based	0.79	2.30°
GEFU@1 (*CMB*)	grayscale + event-based	0.83	2.08°
GEFU@2 (*DBL*)	grayscale + event-based	0.76	2.45°

right angle. For example, by following GEFU@1 (*APS*) (Orange line), it is clear that the *APS*-based model outperforms GEFU@1 (*DVS*) (green line) around frame 325 (the left/right direction is respected) but fails almost for every initial frame (up to 300th frame) by predicting mainly a straight direction as also shown Figure 2. However, GEFU@1 (*DVS*) is not immune to errors; while there are frames where it outperforms all other models (*i.e.,* 315th frame). By looking at Figure 3b it is clear that GEFU@1 (*DVS*) at frame 327 predicts an opposite steering direction (*i.e.,* inverting left with right turns).

In contrast to single input models (*i.e.,*GEFU@1 *APS* and *DVS*), GEFU@2 (*DBL*) (violet line in Figure 2) is more accurate by preventing wrong turns. However, this benefit comes at the cost of more complex architecture that doubles the number of network weights. Consequently, it demands higher computational resources and a larger memory footprint.

Nonetheless, the best results are achieved with GEFU@1 (*CMB*) (red line in Figure 2) which shares the same architecture of GEFU@1, but it lies on a different input organization. In this case, we mixed grayscale, positive, and negative events in a three-channel matrix of $3 \times 200 \times 200$ pixels. This approach allows approximating or even outperforming the performance of GEFU@2 (*DBL*) but with zero impact in terms of additional computational power or memory footprint.

V. Conclusion

In this work, we presented GEFU a frame-based CNN model designed to overcome the limitations of two independent vision-based approaches by converging into the same network grayscale and event-based frames. With the proposed approach, we aim to evaluate two sensor fusion approaches while comparing the results again single input baselines. Indeed, our intent is to push sensor fusion rather than advancing as the best-performing model; thus, this study defines a lower bound in terms of performance achievable by a ResNet50-based architecture. We showed, through manual inspection of a sample of "wrong" predictions, that both grayscale and event-based solo-approaches suffer in their respective domains. Indeed, GEFU@1 (*APS*) confirmed a lower accuracy in sub-optimal light conditions while GEFU@1 (*DVS*) demonstrated weak robustness in static or low-dynamic scenes. Finally, we show how a straightforward model GEFU@1 (*CMB*) can outperform a more complex model GEFU@2 (*DBL*) by only manipulating data input with zero impact on performance.

This observation opens to future investigations and includes:

Investigate stat-of-the-art vision algorithms. This could be achieved by exploiting off-the-shelf architectures that appear natively robust [39]. For example, researchers may train vision algorithms based on Transformers [33]. However, contrary to CNN-based, Transformers need to be trained with massive data, which poses two challenges. First, handling such a huge network requires non-trivial computational resources and a very long training phase not always available at an academic level. Second, it requires extended recording sessions that include as many ambient conditions as possible.

Dig more into sensor fusion. While GEFU is a preliminary attempt to open to sensor fusion, researchers could investigate more on it. For example, researchers can investigate the effect of an early input merging that leaves the downstream network unmodified or a late merging that requires an *ad hoc* architecture.

REFERENCES

[1] T. Serrano-Gotarredona and B. Linares-Barranco, "A 128 ×128 1.5contrast sensitivity 0.9dynamic vision sensor using transimpedance preamplifiers," *IEEE Journal of Solid-State Circuits*, pp. 827–838, 2013.

[2] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240 × 180 130 db 3 µs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[3] 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*, vol. 40, no. 10, pp. 2402–2412, 2018.

[4] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza *et al.*, "A low power, fully event-based gesture recognition system," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7243–7252.

[5] A. Yousefzadeh, G. Orchard, E. Stromatias, T. Serrano-Gotarredona, and B. Linares-Barranco, "Hybrid neural network, an efficient low-power digital hardware implementation of event-based artificial neural network," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2018, pp. 1–5.

[6] A. Gruel, A. Vitale, J. Martinet, and M. Magno, "Neuromorphic event-based spatio-temporal attention using adaptive mechanisms," in *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2022, pp. 379–382.

[7] H. Rebecq, T. Horstschäfer, 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*, vol. 2, no. 2, pp. 593–600, 2016.

[8] 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*. Springer, 2016, pp. 349–364.

[9] H. Rebecq, G. Gallego, E. Mueggler, and D. Scaramuzza, "Emvs: Event-based multi-view stereo—3d reconstruction with an event camera in real-time," *International Journal of Computer Vision*, vol. 126, no. 12, pp. 1394–1414, 2018.

[10] A. Mitrokhin, C. Fermüller, C. Parameshwara, and Y. Aloimonos, "Event-based moving object detection and tracking," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[11] J. H. Lee, T. Delbruck, M. Pfeiffer, P. K. Park, C.-W. Shin, H. Ryu, and B. C. Kang, "Real-time gesture interface based on event-driven processing from stereo silicon retinas," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 12, pp. 2250–2263, 2014.

[12] J. Lee, T. Delbruck, P. K. Park, M. Pfeiffer, C.-W. Shin, H. Ryu, and B. C. Kang, "Live demonstration: Gesture-based remote control using stereo pair of dynamic vision sensors," in *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2012, pp. 741–745.

[13] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, 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*, 2018, pp. 5419–5427.

[14] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, "Cnn variants for computer vision: History, architecture, application, challenges and future scope," *Electronics*, vol. 10, no. 20, p. 2470, 2021.

[15] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis *et al.*, "Event-based vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 154–180, 2020.

[16] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[17] P. Lichtsteiner, "64x64 event-driven logarithmic temporal derivative silicon retina," in *Program 2003 IEEE Workshop on CCD and AIS*, 2003.

[18] P. Lichtsteiner and T. Delbruck, "A 64x64 aer logarithmic temporal derivative silicon retina," in *Research in Microelectronics and Electronics, 2005 PhD*, vol. 2. IEEE, 2005, pp. 202–205.

[19] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change," in *2006 IEEE International Solid State Circuits Conference-Digest of Technical Papers*. IEEE, 2006, pp. 2060–2069.

[20] "inivation," https://inivation.com/, accessed: 2023-01-26.

[21] A. Andreopoulos, H. J. Kashyap, T. K. Nayak, A. Amir, and M. D. Flickner, "A low power, high throughput, fully event-based stereo system," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7532–7542.

[22] L. de Souza Rosa, A. Dinale, S. Bamford, C. Bartolozzi, and A. Glover, "High-throughput asynchronous convolutions for high-resolution event-cameras," in *2022 8th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP)*. IEEE, 2022, pp. 1–8.

[23] C. Brandli, L. Muller, and T. Delbruck, "Real-time, high-speed video decompression using a frame-and event-based davis sensor," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2014, pp. 686–689.

[24] A. Censi and D. Scaramuzza, "Low-latency event-based visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 703–710.

[25] L. Everding and J. Conradt, "Low-latency line tracking using event-based dynamic vision sensors," *Frontiers in neurorobotics*, vol. 12, p. 4, 2018.

[26] 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*, vol. 37, no. 10, pp. 2028–2040, 2015.

[27] I.-A. Lungu, F. Corradi, and T. Delbrück, "Live demonstration: Convolutional neural network driven by dynamic vision sensor playing roshambo," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2017, pp. 1–1.

[28] D. P. Moeys, F. Corradi, E. Kerr, P. Vance, G. Das, D. Neil, D. Kerr, and T. Delbrück, "Steering a predator robot using a mixed frame/event-driven convolutional neural network," in *2016 Second international conference on event-based control, communication, and signal processing (EBCCSP)*. IEEE, 2016, pp. 1–8.

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

[30] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.

[31] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

[32] R. Poojary, R. Raina, and A. K. Mondal, "Effect of data-augmentation on fine-tuned cnn model performance," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 1, p. 84, 2021.

[33] M. N. Islam, M. Hasan, M. K. Hossain, M. G. R. Alam, M. Z. Uddin, and A. Soylu, "Vision transformer and explainable transfer learning models for auto detection of kidney cyst, stone and tumor from ct-radiography," *Scientific Reports*, vol. 12, no. 1, p. 11440, 2022.

[34] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[35] "Event-based vision meets deep learning on steering prediction for self-driving cars: Appendix," https://github.com/uzh-rpg/cvpr18_event_steering_angle, accessed: 2023-01-26.

[36] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature," *Geoscientific model development*, vol. 7, no. 3, pp. 1247–1250, 2014.

[37] Z. Chen and X. Huang, "End-to-end learning for lane keeping of self-driving cars," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1856–1860.

[38] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[39] A. Mastropaolo, L. Pascarella, E. Guglielmi, M. Ciniselli, S. Scalabrino, R. Oliveto, and G. Bavota, "On the robustness of code generation techniques: An empirical study on github copilot," in *Proceedings of the 45th International Conference on Software Engineering*, 2023.