Conference Paper

Cross-Modal Learning Filters for RGB-Neuromorphic Wormhole Learning

Author(s):
Zanardi, Alessandro; Aumiller, Andreas; Zilly, Julian; Censi, Andrea; Frazzoli, Emilio

Publication Date:
2019-06-24

Permanent Link:

https://doi.org/10.3929/ethz-b-000349414 →

Originally published in:

http://doi.org/10.15607/RSS.2019.XV.045 →

Cross-Modal Learning Filters for RGB-Neuromorphic Wormhole Learning

Alessandro Zanardi Andreas Aumiller Julian Zilly Andrea Censi Emilio Frazzoli
Dept. of Mechanical and Process Engineering
ETH Zürich
Zürich, Switzerland
{azanardi,andya,jzilly,acensi,emilio.frazzoli}@ethz.ch

***Abstract*—Robots that need to act in an uncertain, populated, and varied world need heterogeneous sensors to be able to perceive and act robustly. For example, self-driving cars currently on the road are equipped with dozens of sensors of several types (lidar, radar, sonar, cameras, ...). All of this existing and emerging complexity opens up many interesting questions regarding how to deal with multi-modal perception and learning.**

The recently developed technique of "wormhole learning" shows that even *temporary* access to a different sensor with complementary invariance characteristics can be used to enlarge the operating domain of an existing object detector without the use of additional training data. For example, an RGB object detector trained with daytime data can be updated to function at night time by using a "wormhole" jump through a different modality that is more illumination invariant, such as an IR camera. It turns out that having an additional sensor improves performance, even if you subsequently lose it.

In this work we extend wormhole learning to allow it to cope with sensors that are radically different, such as RGB cameras and event-based neuromorphic sensors. Their profound differences imply that we need a more careful selection of which samples to transfer, thus we design "cross-modal learning filters". We will walk in a relatively unexplored territory of multi-modal observability that is not usually considered in machine learning. We show that wormhole learning increases performance even though the intermediate neuromorphic modality is on average much *worse* at the task. These results suggest that multi-modal learning for perception is still an early field and there might be many opportunities to improve the perception performance by accessing a rich set of heterogeneous sensors (even if some are not actually deployed on the robot).

I. Introduction

Wormhole learning [1] is a technique that can be used in an object detection scenario to enlarge the effective operating domain of a detector, if one has an auxiliary sensor available *temporarily*. One can let sensor 1 "teach" sensor 2; then, the student becomes the teacher, and sensor 2 "teaches" sensor 1 as illustrated in Fig. 1. Surprisingly, under certain complementarity conditions of the two sensors, one can find that the teacher has something to learn from the student, even though the student has learned everything it knows from the teacher.

For instance, in the scenario where the two sensors are an RGB camera and an infrared (IR) camera, starting from an RGB detector trained at daytime, one can first learn an infrared detector using the RGB detector as teacher; then exploit the invariance of the IR sensor to illumination to extrapolate to low-light conditions; then use the detection from the IR domain to re-learn an RGB detector, which eventually has better performance / larger operating domain than the original

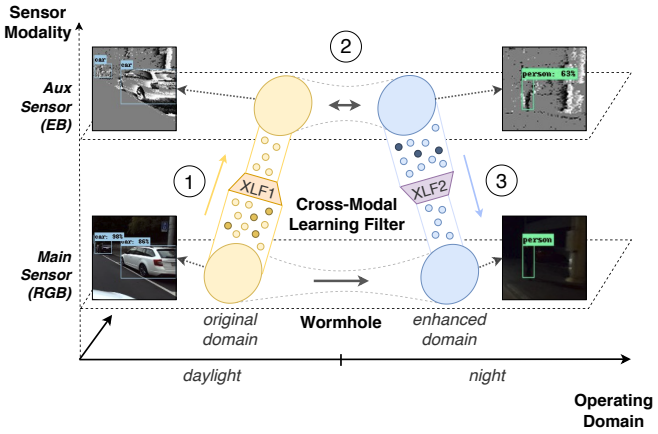
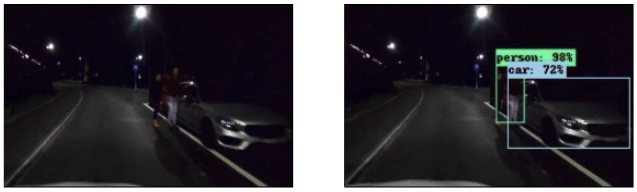


Fig. 1. The principle of wormhole learning is illustrated. Starting from the bottom left, in a first step, transfer learning is applied to learn a detector in the auxiliary domain. From there, the inherent invariance to a specific nuisance for the task allows us to travel across the operating domain. In the final step, the loop is closed performing another domain transfer: back into the main sensor space, but in a new operating envelope. We validate this concept adopting an event-based camera (EB) as auxiliary sensor and exploiting its inherent invariance to illumination, we can enhance the main RGB detector. With the due carefulness, the principle still applies even though the two sensors are radically different. Indeed, we have to introduce two cross-modal learning filters (XLFs) to cope with the, otherwise "noisy", domain transfer steps.



(a) Before *wormhole learning* (b) After *wormhole learning*

Fig. 2. Despite the availability of labeled data only for daytime, *wormhole learning* successfully improved RGB object detection performance in insufficient illuminated environments. Take, for instance, a parked car on the curb without headlights. At night it has a different appearance than at day (e.g. no reflections on the windows), yet is detected after wormhole learning.

one. Where does the additional information come from? The "trick" in wormhole learning is that the invariance properties of the second sensor act as an additional prior. So, informally, the second sensor has "learned" some invariance from the first sensor. The expression "wormhole" refers to the idea that transferring to a different modality allows to create a "portal" that connects one part of the operating domain to another.

This paper presents an advance to wormhole learning which

we name *cross-modal learning filters* (XLF). The basic idea is that, if the sensors are very different, not all detections from one sensor are useful examples for learning from data of the other sensor. Therefore, what is needed is a decision procedure that chooses whether to use one detection of a sensor for training the other, or to simply just ignore the sample. We call this decision procedure *cross-modal learning filter* (XLF). In Fig. 1 the XLFs are shown as funnels during the two transfer steps.

This paper applies the wormhole learning principle to the pair RGB camera and event-based neuromorphic sensor, for the task of object detection, and the scenario of urban driving. We show that the XLFs for this application, in addition to being necessary, are also intuitive and easy to compute, and so they are a computationally-negligible addition that improves the performance of the wormhole learning.

The event-based detector learned in the first wormhole step is also the first event-based object detector designed and tested for realistic driving scenarios (previous attempts considered much easier settings [2]–[7]).

We will see that the performance of this event-based detector obtained with the first transfer step is much worse than the RGB detector. This is explained by a combination of factors, including: 1) limitations of the resolution of existing event-based sensor prototypes; 2) the fact that event-based vision is a very early field, and our learning methods are biased towards frame-based representations; 3) an unavoidable difference in sensor aptitude for the task at hand.

The very curious finding is that, while the event-based detector performs much worse, we nevertheless observe that, when we transfer back to the original RGB domain, we can still improve the performance of the RGB-based detector.

Thus we find that a perception system can improve its performance on a task by *temporary* access to a sensor *that does not even work that well for the task*. These results suggest that our understanding of perception and learning in multi-modal setting is still fairly naive, and there might be many opportunities to improve robot perception performance by considering innovative combinations of complementary sensors.

II. Related work

Multi-modal perception, fusion, learning: Our work is fundamentally different from works in sensor fusion, as we never consider the data from the two sensors at the same time. In wormhole learning, we show that *having had* another sensor is valuable, even if you do not use it afterwards.

Transfer learning: Transfer learning is the ability to transfer "knowledge" across changing data distributions [8]. A large body of research has been developed to solve tasks such as learning to play computer games [9], translation of previously unseen language pairs using generalizations learned on known language pairs [10], executing several different robotic tasks by employing modular neural networks [11], and employing CycleGANs [12] to transfer one image domain to another.

The difference of our work with respect to transfer learning is that we can show that transferring *back* to the original domain might improve the performance on the task.

Beyond simple label noise: The problem of creating cross-modal learning filters is superficially similar to the problem of reducing "label noise" [13]. The assumption that labeled data is correct has remained unquestioned until recently [14]. Nowadays, two main forms of label noise are distinguished:
1) Feature noise mainly affecting the observable part of a given feature or class, for instance background noise in an image or suboptimal bounding box position.
2) Class noise describing incorrect labeling of an instance or object (e.g. labeling of a car where no car is observable).

The challenges encountered in our setup cannot be reduced to dealing with simple "label noise" (thought it still exists, naturally). Here, we deal with the case that the labels are "correct" for one sensor but not for the other. We would like to point the reader's attention to a small but significant epistemic divergence of our setup compared to the usual supervised learning setting. If there is more than one sensor, it is important to distinguish between the *ideal task* Y and the task conditioned on the sensor data Z_a, which we define as $Y_a \doteq Y|Z_a$. The ideal task Y is to be considered an unobservable absolute ground-truth. It exists independently of the sensors we might use, or not, to peek at the world.

In a conventional supervised learning setting, this distinction is blurred; if there is only one sensor, and annotations are typically created from the output of that sensor, by definition *and* by construction the label distribution is the same as the label distribution given the data. In our setting, this is not the case. For example, a camera cannot see through fog, while an IR camera can. In these conditions, the distribution of the task variable conditioned on the first sensor is different than if conditioned on the second sensor; and consequently, different from the ground truth distribution.

In the derivation of cross-modal learning filters we encounter unusual notions of "multi-modal observability". In the deterministic case, if one only has one sensor, then a certain quantity of interest can be observable or not. But if there are n sensors, then there are 2^n possible observability outcomes. Further, one can ask the question of whether the quantity is observable from sensor a, given that it is observable for sensor b, leading to n^2 pairwise conditional observability conditions. In the learning setting, we have the added issue that, while something could be observable in principle from a sensor, there might not be sufficient training data to learn to detect them. Thus, if one tries to picture all possible regions of operating domains, there are at least 3^n combinations (Fig. 4).

To our knowledge, these extended intertwined notions of observability and data sufficiency for multiple sensors are not well explored, neither in robotics, nor in the broader learning/detection/filtering fields.

Fig. 3. Exemplary result from [1]. It has been shown that an RGB detector with *temporary* access to an infrared sensor can learn to recognize cars, persons, and objects at night in spite of the initial daytime only data.

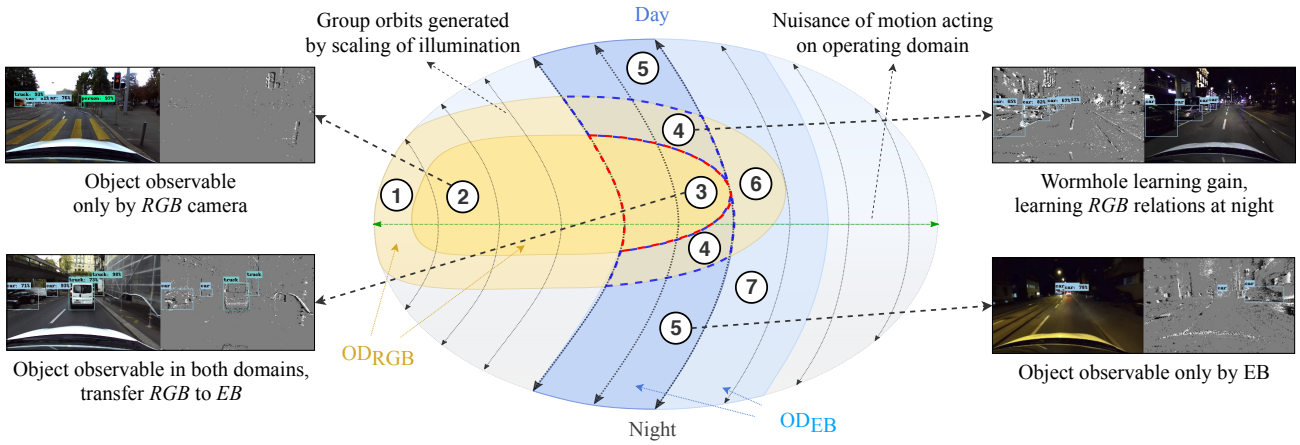


Fig. 4. A representation of partially overlapping operating domains (OD) of two sensors. The light color areas in yellow and blue represent the larger portion of the space where the two sensors can idealistically operate. Darker hues are the restriction to our data samples. Wormhole learning starts off from regions 2 and 3, applies the first domain transfer in region 3, expands to 4 and 5 due to the auxiliary invariance, and transfers back to the RGB domain gaining 4. While in [1], the transfer learning region 3 is taken for granted, in this work we look into how to **enforce** this region via cross-modal learning filters (XLFs). Symmetrically, during the first domain transfer we enforce the samples to be drawn from 3 rather than 2, during the backward transfer, we ensure samples are not taken from 5 but only from 4 or 3. It is in this context that "detection" takes over a shade of relativeness based on the sensor conditioning.

III. Wormhole Learning

In this section we recall the wormhole learning (WHL) setup.

We use common notation. A random variable is indicated by upper case letter such as V. The random variable is a function from the sample space Ω to the domain $\mathcal{V}$: $V : \Omega \to \mathcal{V}$. A particular sample is denoted using bold lower case, like $\boldsymbol{v} \in \mathcal{V}$.

A. Assumptions

The WHL setup assumes that:

1. There are two sensors available. The work [1] considered the case of an RGB camera and an IR camera; the present work considers the pair of an RGB camera with an event-based neuromorphic sensor. The data from sensor a is a random variable Z_a which takes values $\boldsymbol{z}_a$ in set $\mathcal{Z}_a$.

2. The sensors are "complementary". In an intuitive sense, the sensors must have operating domains that are distinct enough to not be redundant, but overlapping enough to allow transfer of some information. Refer to [1] for an information-theoretic formalization of this idea.

3. There exists a *scene* that generates the data. The scene is a random variable indicated as X. The data from the different sensors are independent given the scene.

4. There is a *task*, another random variable, called Y. In the object detection scenario, these are the labels. Given the scene, the task is independent of the sensing data.

5. There are available unannotated samples from the joint distribution $\langle Z_a, Z_b \rangle$.

6. There is a known approximation to the label distribution $p(Y|Z_a)$. We think of this as a pretrained object detector that uses only the first sensor and that is trained only in a limited operating domain.

The wormhole learning principle says that we can improve the performance of the initial detector on previously-unexperienced operating ranges by exploiting temporary access to the second sensor, without any additional labeled data.

B. Wormhole learning algorithm

Abstracting away from the particular sensor and learning methods, the WHL algorithm proceeds as follows (Fig. 1):

1) Obtain samples $\boldsymbol{z}^k$ in a domain where the initial detector $p(\boldsymbol{y}|\boldsymbol{z}_a^k)$ is sufficiently accurate.
2) Using $\boldsymbol{z}_a^k$, generate the label $\boldsymbol{y}^k$ from the initial detector.
3) Treat the pair $\langle \boldsymbol{z}_b^k, \boldsymbol{y}_a^k \rangle$ as a training sample to learn the distribution $p(\boldsymbol{y}|\boldsymbol{z}_b)$.
4) Once the distribution $p(\boldsymbol{y}|\boldsymbol{z}_b)$ is learned, we proceed in reverse. In a *different* domain, where the distribution $p(\boldsymbol{y}|\boldsymbol{z}_b)$ is proficient, sample $\langle \boldsymbol{z}_a^k, \boldsymbol{z}_b^k \rangle$, compute the label $\boldsymbol{y}^k$ from $p(\boldsymbol{y}|\boldsymbol{z}_b^k)$, and treat $\langle \boldsymbol{y}_b^k, \boldsymbol{z}_a^k \rangle$ as a sample for learning $p(\boldsymbol{y}|\boldsymbol{z}_a)$.

C. Example applied to RGB and IR cameras

For the particular case reported in [1], the two sensors were an RGB camera and an infrared (IR) camera, the learning methods were convolutional neural networks (CNNs) [15], and the initial approximation was a pre-trained neural network that was trained only on daytime data. The algorithm above was specified as follows:

1) Obtain a joint dataset for RGB and IR camera.
2) Use the pre-trained RGB network to generate detections.
3) Use these detections to train a second neural network, which will create detections from the IR camera data only.
4) Once the IR camera-based detector has been learned, do the process in reverse, by generating samples to re-learn (or re-tune) the initial detector.

The enhanced RGB detector has improved its performance outside the original operating domain. Fig. 3 provides an example, in which the RGB detector has been enhanced to better detect objects at night. In particular, the detector has learned that cars have bright lights at night; something that was not possible to extrapolate given only daytime data.

IV. Cross-modal Learning Filters (XLF) for Wormhole Learning

Algorithm modification: We generalize the WHL algorithm by introducing cross-modal learning filters (XLFs), which are functions of the type

$$\text{xlf}_{a\to b} : \mathcal{Z}_a \times \mathcal{Z}_b \times \mathcal{Y} \to \textsf{Bool}. \tag{1}$$

Suppose that for sample k, we know the sensor data $\langle \boldsymbol{z}_a^k, \boldsymbol{z}_b^k \rangle$ and we know that this is a detection of class i for the first sensor: $\boldsymbol{y}_a^k = i$. We want $\text{xlf}_{a\to b}(\boldsymbol{z}_a^k, \boldsymbol{z}_b^k, \boldsymbol{y}_a^k)$ to answer the question of whether the detection $\boldsymbol{y}_a^k$ is useful for training the detector for the second sensor or not. If it is, we are going to proceed with the wormhole learning step; otherwise we are going to discard the sample. Symmetrically, we want $\text{xlf}_{b\to a}(\boldsymbol{z}_a^k, \boldsymbol{z}_b^k, \boldsymbol{y}_b^k)$ to tell us whether a detection $\boldsymbol{y}_b^k$ is a good example to use to train the detector using the first sensor.

Basic considerations for the design of the XLFs: We start by noting that the XLFs are not trivial, as there are certainly detections that we do not want to transfer. For instance, suppose that an RGB sensor is at rest and there is no motion in the scene. An event-based sensor taped to the RGB sensor would produce no events, except for spurious events due to measurement noise. In these conditions, if the RGB sensor sees an object, we would *not* want to use it as a training example for learning an event-based object detector, because, with there not being any events, we would just increase the variance of the weights by overfitting to noise. Vice versa, an event-based camera can see at very low light conditions, while an RGB camera cannot, so the XLF is not trivial in the other direction either. (The authors were initially divided on this prediction: shouldn't the networks just ignore detections that do not correlate with anything? We did find experimentally that, in fact, the networks performance is negatively impacted by examples that are unobservable, at least for the common training policies that we used.)

One can expect that non-trivial cross-modal learning filters are required for wormhole learning for any pairs of sensors in which one does not fully "dominate" the other. (Here, "dominate" should be intended in the meaning given by [16], extrapolating that meaning to the probabilistic case.)

Specification for the XLFs: To get to a formalization of what an XLF should be, we must first formalize what it means for something to count as a detection. This can be formalized in a handful of different ways, depending on exactly which decision problem one is solving. One could say that a detection for a class happens when the likelihood of the data given the class is higher than a certain threshold. So, we have a detection for a class if $p(\text{data}|\text{class}) > \text{threshold}$. Using our notation, we detect class i with data sample $\boldsymbol{z}$ if $p(\boldsymbol{z}|\boldsymbol{y} = i) > c$. Alternatively, one could instead consider for detection the likelihood ratio of one class w.r.t. another exceeding a threshold, or testing posterior class probabilities. The ensuing formalization would not change, as long as we have defined some scalar score

$$\gamma_a : \mathcal{Z}_a \times \mathcal{Y} \to \mathbb{R} \tag{2}$$

that we are going to compare against a threshold c. We assume for simplicity of notation that the threshold c is the same for all sensors and all classes, though this can be easily relaxed.

Suppose then that we have defined these boundaries γ_a and something counts as a detection if $\gamma_a > c$. The previous discussion referencing RGB and event-based cameras should be sufficient to argue that, in the absence of any other assumptions about the sensor models, the functions γ_a and γ_b can be completely unrelated. Thus, in the general case we can picture as in Fig. 5 (left) that there are up to four regions of the data space $\mathcal{Z}$, depending on whether a particular sample would count as a detection for sensor a ("B"), sensor b ("D"), both ("C"), or neither ("A").

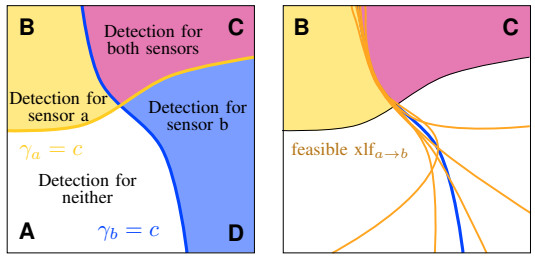


Fig. 5. *Left*: Depicted is the operating domain of two sensors with regions indicating where objects can be detected by both (C), sensor A (B), sensor B (D) or neither (A). *Right:* Boundaries laid out by possible cross-modal learning filters (XLF) to find whether conditioned on detection by one sensor, detection is also possible by the other. Note that it does not to fully mimic the boundary representing the main detection in the first sensor domain.

At this point, there might be suspicion in the reader that we have embarked on a circular argument. Is creating a cross-modal learning filter not equivalent to defining the detector itself?

The answer is: no—it is actually much easier. Notice that in the wormhole learning algorithm, we are asked to decide whether a sample should be considered a detection for the second sensor *only in the case that it is a detection for the first sensor*. This conditioning operation makes all the difference. Again in a graphical form (Fig. 5, right), we can see that, conditioned on knowing that the sample is a detection for the first sensor, we know that we are in regions "B" or "C"; we can already exclude regions "A" and "D". Graphically, we are not asked to reconstruct the entire boundary line between A ∪ B and C ∪ D, which is equivalent to knowing the detector for the second sensor, but only to reconstruct the boundary between "B" and "C". Our specification for an ideal learning filter $\text{xlf}_{a\to b}$ should only constrain what values it takes on the regions "B" and "C". In formulas, we have

$$\text{xlf}_{a\to b} : \langle \boldsymbol{z}_a, \boldsymbol{z}_b, i \rangle \mapsto \begin{cases} \gamma_b(\boldsymbol{z}_b, i) > c, & \text{for } \gamma_a(\boldsymbol{z}_a, i) > c, \\ \text{anything}, & \text{elsewhere.} \end{cases} \tag{3}$$

Finally, we need to adapt the specification above to the noisy probabilistic setting. We can restate the formulas above by saying that what we want is to define a random variable $\text{xlf}_{a\to b}$ that is maximally predictive of the random variable $\gamma_b > c$ when $\gamma_a > c$. This can be written formally by using mutual information as the scoring function, conditioned to the event $\gamma_a > c$. We finally reach the definition that the performance metric for a candidate XLF $\text{xlf}_{a\to b}$ is

$$R(\text{xlf}_{a\to b}) \doteq \mathcal{I}(\text{xlf}_{a\to b}; \gamma_b > c \mid \gamma_a > c). \tag{4}$$

V. Application to wormhole learning across RGB and event data

A. Event-based cameras

Event-based cameras have been introduced fairly recently [17]. In an event-based camera each pixel works asynchronously both in space and in time. The output from the sensor is a stream of events, where each event is generated if the local intensity of the light changes more than a threshold. More formally, we have that the k-th event is a tuple $\langle t_k, x_k, y_k, p_k \rangle$ containing the timestamp $t_k \in \mathbb{R}_{\geq 0}$, the pixel position $x_k, y_k \in \mathbb{N}$, and the polarity of the event $p_k \in \{-1, +1\}$, which indicates whether the brightness increased or decreased. Let Δ be the triggering threshold, I the light intensity hitting the photo-diode, and t_{k-1} the timestamp of the last event. Then the next spiking time can be characterized as

$$t_k = \arg\min_{t > t_{k-1}} \{ t \; s.t. \; |\log(I_t) - \log(I_{t_{k-1}})| > \Delta \}. \quad (5)$$

This expression neglects the refractory period, asymmetric thresholds, and noise. If the signal of the light intensity is "smooth enough" we recall from [17] that the event rate is

$$\text{event rate} \propto \frac{1}{I(x,y,t)} \frac{dI(x,y,t)}{dt}. \quad (6)$$

Simply expanding the total time derivative of the light intensity according to optical flow equation give us necessary conditions for a non-trivial output from the sensor:

$$\frac{dI(x,y,t)}{dt} = \langle \nabla I(x,y), \vec{v} \rangle + \frac{\partial I(x,y)}{\partial t}. \quad (7)$$

Either the scalar product of the spatial gradient and the velocity vector is non-zero, or the light intensity changes in magnitude. Except for those cases where there are pulsing lights in the field of view, the latter is unlikely. Thus, an object needs to have a velocity component on the image plane in order to produce events.

An established upside of event-based cameras is that they can operate in very high dynamic range environments. Indeed, while machine-vision cameras typically achieve dynamic ranges of 60 dB, event-based cameras reach from 120 dB up to 143 dB [17]–[19].

B. Using CNNs with event-based data

We use a frame-based representation of event-based data, and use a CNN architecture for learning a detector. To generate frames from events, we use the *surface of active events* (SAE) representation [20]. The SAE can be thought as a buffer that, for each pixel, keeps in memory the last event fired at the specific location. In order to favor strong activations on the object location, the SAE is weighted with a Gaussian-like profile over the last 100 ms. This value works well for the typical stimuli experienced by the sensor in an urban driving scenario. Note that precise synchronization of object location can happen only in correspondence of the frames' timestamps. Thus, given t_k as the timestamp of the RGB we extract the corresponding frame in the event domain in the interval $[t_k - 100\,\text{ms}, t_k]$. We denote this frame image as I_{EB}.

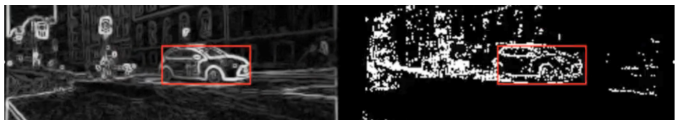
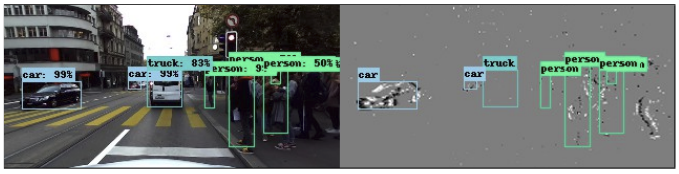
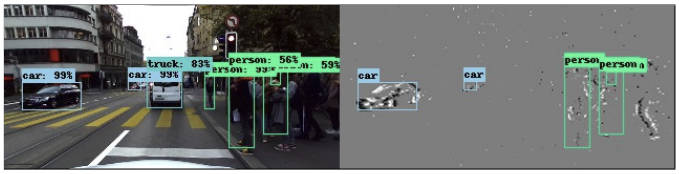


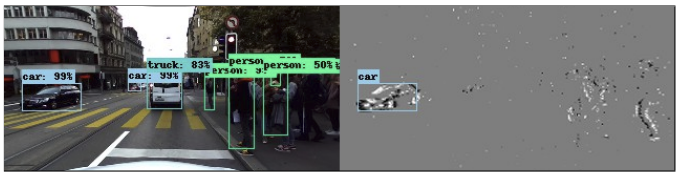
Fig. 6. On the left the gradient of the RGB image, on the right the events formatted into a frame. Note that polarity of the events is neglected. The score is computed only on the detection box in red.



(a) **No** XLF: $\underline{S^{\text{eo}}} = 0$; all labels are transferred.



(b) **Low threshold** XLF: $\underline{S^{\text{eo}}} = 0.35$; cleaning "noisy" labels.



(c) **High threshold** XLF: $\underline{S^{\text{eo}}} = 0.65$; only the clearly visible car is kept.

Fig. 7. The effect of XLF is illustrated from top to bottom with three increasing filter threshold levels.

C. Cross modal filters for RGB and EB

To obtain the XLF $\text{xlf}_{\text{RGB}\to\text{EB}}$, we should answer the following question: assuming that a target is detected in an RGB frame, under what conditions is it also detectable from event data? The answer is easy: it depends on the apparent motion. If there is no relative motion between sensor and object, no event will be generated; and if there is sufficient motion, then the events contain sufficient information to detect the object. So, all we need to compute the XLF $\text{xlf}_{\text{RGB}\to\text{EB}}$ is finding a proxy for the intensity of the apparent relative motion between sensor and objects. Note that we do not need to solve an optical flow problem and estimate the apparent motion at each point; we just need a single scalar for each detection: the average of the norm of the apparent motion in the detection box.

Given the detection box, for the pair of data $\langle I_{\text{RGB}}, I_{\text{EB}} \rangle$ we compute an *edge overlap score* S^{eo} as follows:

$$S^{\text{eo}} = \frac{\sum_{x,y \in \text{Box}} \sqrt{|I_{\text{EB}}(x,y)| \, \|\nabla I(x,y)\|_2}}{\sum_{x,y \in \text{Box}} \|\nabla I(x,y)\|_2}. \quad (8)$$

The detection taking place in the RGB domain gives us the confidence that ∇I_{RGB} contains a signal with enough discriminant power for the detected class. Since the velocity

is unknown for the RGB domain, we simply compute a score proportional to events being at gradient location, which is therefore proportional to the velocity component parallel to the spatial gradient of the light field. The square root is not predicted by the theory; it serves as a concave nonlinearity to mitigate the effects of unmodeled heavy tails of noise. Without the square root, the score can be seen as an approximated matched filter paired to the linear observation model given by (7).

To obtain the decision policy $\text{xlf}_{\text{RGB}\to\text{EB}}$ we check whether the score S^{eo} in (8) exceeds a threshold $\underline{S^{\text{eo}}}$, and thus define the cross-modal learning filter as

$$\text{xlf}_{\text{RGB}\to\text{EB}}(\boldsymbol{z}_{\text{EB}}, \boldsymbol{z}_{\text{RGB}}) \doteq S^{\text{eo}} > \underline{S^{\text{eo}}}. \tag{9}$$

The backward $\text{xlf}_{\text{EB}\to\text{RGB}}$ is computed as the forward, but normalizing by the conditioning signal. This corresponds to the same expression as in (8) with $\sum_{x,y\in\text{Box}}|I_{\text{EB}}(x,y)|$ as denominator.

Choice of thresholds: In principle, we could choose a different threshold $\underline{S^{\text{eo}}}$ for each class; for now, we only explored the method properties for a fixed threshold.

We can predict that there is an optimal value of the threshold from simple considerations. As we increase the threshold, it is increasingly likely that the examples are good examples to use for learning the detector for the second sensor. However, as we increase the threshold, we use fewer and fewer examples, so the performance of the detector will suffer. Thus we can predict that there is a point of diminishing returns, and we observed this empirically.

At this point, we have not found a way to obtain the best value for the threshold analytically; because it is a scalar quantity, we just sweep a plausible range to find the best choice.

VI. Experimental results

A. Dataset

We recorded our own multi-modal dataset, using a device that contains in the same physical package RGB cameras, neuromorphic sensors, and IR camera. The data is collected by driving in an urban environment. The dataset comprises approximately 500k camera frames and is much more extensive and realistic compared to the existing datasets that have simultaneous event-based and RGB cameras streams [2], [21]–[23]. The dataset will be subsequently released. The details for the two sensors that we adopted in this work are provided in Table I.

To perform WHL, an initial separation between daytime (D) and night (N) is required. Subsequently, in order to evaluate the detectors reliably, we split the dataset in three: training, validation, and testing. The test set has been generated by uniform sub-sampling — at a rate of 1 out of 4 frames — of few recordings that have been kept aside throughout the whole WHL process. This data has been hand-labeled adding up to a test set size of $2.2k$ frames at day and another $2.2k$ samples of night. The second partition of recordings was employed to create a validation set for monitoring the training. We hand-labeled a validation set of $1.1k$ frames at day, as well as at night, also sub-sampling uniformly 1 out of every 3 frames from separate recordings kept for this purpose. Finally, the training partition accounts for 98.3k frames at daytime and 42k at night time. Note that these three partitions belong to completely different recordings.

TABLE I
HARDWARE SPECIFICATIONS

Feature	Color Camera	Event-based Camera
Brand	Stereolabs	Insightness
Type	ZED	Silicon Eye
Sensor Size	1/3"	1/3.2"
Pixel Size	2 μm	13 μm
Spatial Resolution	1280×720 px	326×260 px
Temporal Resolution	30 Hz	~10 KHz
Dynamic Range	12 - 14 dB	> 98 dB
Lens	Built-in	M12
Aperture	$f2$	$f2.1$
Field of View	90°	85°

B. Training details

We adopt the pretrained Faster-RCNN NASnet [24] checkpoint, which is publicly available through the Tensorflow model zoo project [25] to generate the labels of the initial dataset for daytime training of the first RGB detector. The WHL process requires the training of three detectors. The first originates from the initial dataset, the second corresponds to the detector in the auxiliary sensor domain, the last is, again, in the main sensor domain expanding the operating domain of the first by leveraging semi-supervised learning from the second.

Each step of the training sessions share the same initial network checkpoint of the Single Shot Detector (SSD) with Inception V2 modules [26]. This is done for ease of convergence and keep the architecture of the three detectors as a constant.

We chose a batch size of 32 per step for our learning configuration. Furthermore, an RMSprop optimizer with initial learning rate of $6 \cdot 10^{-3}$, momentum of $\beta = 0.9$, and an exponential decay factor of 0.95 after 60k steps were applied. Additionally we used standard data augmentation techniques [27] to increase robustness and prevent overfitting; these include: random horizontal flips and random crops. In the RGB domain we additionally applied random scaling in brightness and contrast. The latter techniques have to be considered as having only a small and local impact with respect to the general changes in illumination (such as day and night).

C. Wormhole Learning Process

1) Step 1: Training of a daytime only RGB *detector:* We denote a detector by its parameters θ. An initial network $\theta^{\text{D}}_{\text{RGB}}$ is trained using small amount of ground-truth data $Y^{\text{D}}_{\text{train}}$ for day-time only RGB images, as depicted in Fig. 8. Then $\theta^{\text{D}}_{\text{RGB}}$ is deployed to generate inferred labels $Y^{\text{D}}_{\text{RGB}}$ that can be transferred to Y^{D}_{EB} thanks to paired images in the RGB/EB setup. Before being passed to EB domain, each detection pair is parsed by the cross-modal filters.

2) Step 2: Training of an event-based detector: An auxiliary detector $\theta^{\text{D}}_{\text{EB}}$ is trained in the event-based domain by exploiting Y^{D}_{EB} and Z^{D}_{EB}. Thanks to the approximate domain invariance to D/N, this particular network θ^{D}_{EB} is also effective on event-based data at night Z^{N}_{EB}. Hence, it represents a proxy also for Y^{N}_{EB}, which in turn can be transferred back to $Y^{\text{N}}_{\text{RGB}}$. Symmetrically,

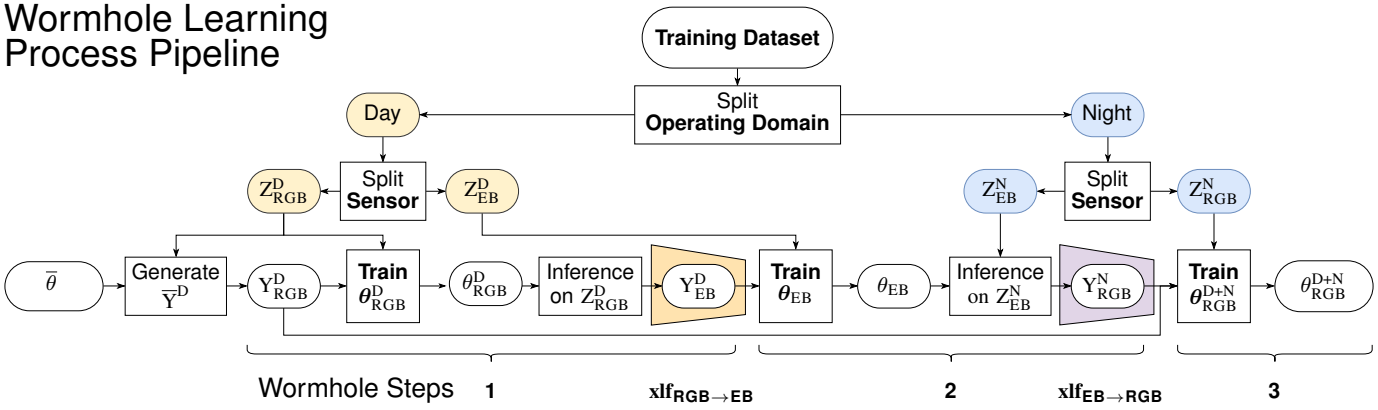


Fig. 8. The process can be generally divided into three sub-tasks: 1) Training of the network using day-time only annotations and transferring generated labels to $\mathrm{OD_{EB}}$, after it has been passed through the cross-modal learning filter ($\mathrm{xlf_{RGB\to EB}}$), depicted as orange funnel. 2) Training of the event-based network using labels generated by the object detector trained in last step and exploiting invariance of the DVS camera to generate labels at night, which are transferred back to $\mathrm{OD_{RGB}}$. Here, the violet funnel represents the cross-modal learning filter ($\mathrm{xlf_{EB\to RGB}}$) from EB to RGB. 3) Wormhole loop is closed by retraining the original network with day-time and night-time data from the event-based network.

we apply the backward cross-modal filter ($\mathrm{xlf_{EB\to RGB}}$) to enrich the future dataset.

3) Step 3: Training of a day & night RGB *detector:* In the last step the wormhole is completed. The original network is re-trained with the initial labels at day and newly generated labels at night $Y^{\mathrm{D}}_{\mathrm{RGB}} + Y^{\mathrm{N}}_{\mathrm{RGB}}$ to yield $\theta^{\mathrm{D+N}}_{\mathrm{RGB}}$. As a result, the new detector *"exits the wormhole"* in a new operating environment.

D. Results

In this section, we present the results of the *wormhole learning* experiment summarized in Tab. II, III and Fig. 9. The performance assessment of the networks follow the PASCAL VOC best practice guidelines [28]. We use the standard 0.5 IoU score for a positive match and multiple detection of the same object are classified as false positives[1].

In Tab. II, the first two rows display the performance of the initial RGB object detector, as evaluated against our test-set at day and night, respectively. It performs well at day-time but suffers from a drop in prediction performance at night-time, as is evident by the drop from 59.07% to 32.19% in mean average precision (mAP). The next two rows contain average precision scores of the event-based detector. We can clearly see, that it underperforms at day and night compared to the initial RGB object detector exhibiting a mAP of 26.20% and 16.22%, respectively. Notably, the performance at night is also inferior to the one of the initial RGB detector.

The last two rows showcase the RGB detector results after WHL, revealing that even with the meager result of the event-based network we are able to elevate the initial RGB detection performance at night by inheriting part of the invariance of the EB sensor. Although the day-time mAP scores decreases by 1.73% to 58.05%, we can see a relative increase of 28.8% to 41.46% in mAP at night. The single class average precision rises at night frequently measured classes such as *car*, *person*, *truck* or *train*. On the other hand, due to natural imbalance of objects in the recordings (indicated by the class count in the parentheses in Tab. II), the detection performance of *bicycle* and *motorcycle*, respectively drops or is non-existent. The *truck* class experienced the highest relative increase in detection performance of 147.7%, followed by *truck* with a rise of 133.1% and 36.7% for *car*.

More interestingly, in Tab. III we see the performance of the event-based detector depending on the learning filter threshold. Compared to not applying the cross-modal learning filter, the detection performance at day can be increased by 27.5% when choosing a learning filter with low threshold but only increases by 9.4% when choosing one with a high threshold. At night, the detection performance of the network trained using a learning filter with low threshold shows a 97.1% increase while using a high threshold filter setting the performance shows a 126.2% increase, compared to the network trained without applying any cross-modal filter. Note that if the filter is chosen to discard labels too freely, the dataset is shrunk significantly and thus decreases in performance.

Eventually we observe here, that although the performance at night using the high threshold learning filter is slightly superior, we chose to conduct the experiment using the network trained on the low threshold learning filter as it performed best during training on the validation set in overall performance.

VII. Discussion

These results on wormhole learning show that there are many creative ways to combine the data from heterogeneous sensors, and an additional sensor can be useful, even if you only have it during training, and even if it is not particularly good at the task at hand, or, equivalently, even if we do not know how to use it well for the task at hand.

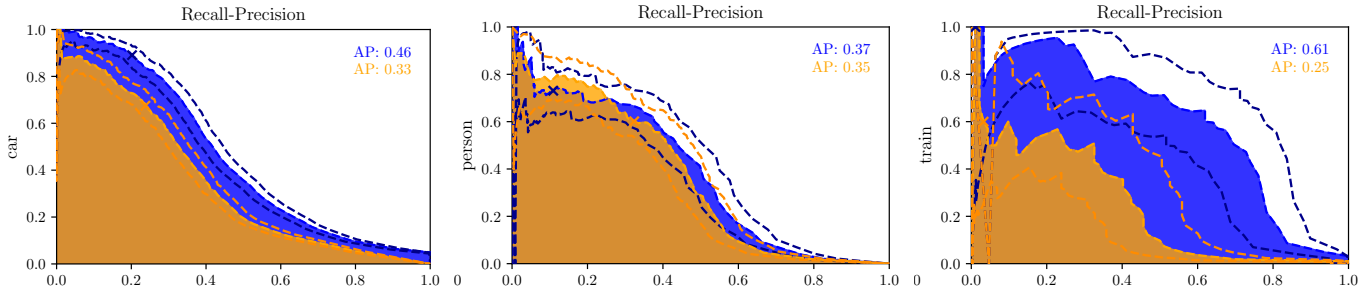
For the particular case of event-based neuromorphic sensors, we remark that we do not consider the reported performance to be representative of the possible performance when the technology is more mature. The hardware is still developing and nowhere near the projected future performance. There exists a handful of event-based sensors being developed independently

[1]We note in passing that these thresholds are not entirely reasonable for the self-driving car application, where the cost of a false negative is very high.

TABLE II
DETECTION PERFORMANCE COMPARISON FOR EACH CLASS. IMPROVED VALUES ARE MARKED IN BOLD. NUMBER OF TRAINING SAMPLES IN ().

Network	Testset	Car	Person	Truck	Bus	Train	Bicycle	Moto.	mAP
1 RGB [a]	Day	60.00 (404.0k)	52.99 (144.9k)	61.98 (35.1k)	0.26 (16.3k)	89.74 (9.5k)	41.80 (13.2k)	6.06 (4.9k)	59.07
	Night	33.31 (0k)	35.15 (0k)	6.40 (0k)	27.56 (0k)	24.61 (0k)	29.46 (0k)	0.00 (0k)	32.19
2 EB [b]	Day	29.35 (141.7k)	31.80 (55.2k)	17.54 (16.2k)	0.00 (4.1k)	46.14 (4.7k)	8.06 (2.8k)	0.49 (1.6k)	26.20
	Night	19.61 (0k)	8.26 (0k)	4.53 (0k)	3.87 (0k)	19.43 (0k)	0.43 (0k)	0.03 (0k)	16.22
3 RGB [c]	Day	56.01 (404.0k)	49.29 (144.9k)	**68.06** (35.1k)	**0.44** (16.3k)	88.76 (9.5k)	39.59 (13.2k)	2.75 (4.9k)	58.05
	Night	**45.53** (15.9k)	**37.18** (6.8k)	**14.92** (0.4k)	15.90 (0.1k)	**60.97** (0.1k)	5.08 (0k)	0.00 (0k)	**41.46**

[a] Network trained on day only data
[b] Network trained on data generated automatically by first network in conjunction with a learning filter with low threshold
[c] Network re-trained on day + night data



(a) The new detector dominates "strictly" the initial for all values of recall.
(b) The changes are within the confidence bounds. The two detectors are almost equivalent.
(c) Despite low sample size — thus wide confidence bounds — a significant improvement can be appreciated.

Fig. 9. Precision-Recall plots comparing the single class performance of the RGB detector before wormhole learning (orange) and after (blue). The 95% upper and lower confidence bounds are visualized using dashed lines, showing significant increase in performance for class *car* and *train*, but not for *person*.

TABLE III
DETECTION PERFORMANCE IN MAP DEPENDING ON XLF SETTINGS

Network	Testset	None	Low Threshold	High Threshold
2 EB	Day	20.55%	**26.20%** (+27.5%)	22.49% (+9.4%)
	Night	8.23%	16.22% (+97.1%)	**18.63%** (+126.2%)
3 RGB	Day	58.05%	53.53% (-7.8%)	**58.90%** (+1.5%)
	Night	**41.46%**	36.43% (-12.1%)	32.93% (-20.6%)

by different groups, and these vary considerable in their output and the tuning parameters that they expose. These are also very early days for event-based vision. In fact, this work is, to the authors' knowledge, the first to consider the problem of object detection in an urban driving scenario under the same conditions used for RGB detection. There is certainly considerable room for improvement, as we used techniques very similar to their frame-based counterparts (partly because here we were interested in the gap in performance allowed by wormhole learning, rather than in the absolute precision of event-based detectors.) Nevertheless, it looks like that we have proved that neuromorphic sensors are already useful complements to RGB sensors; even if they do not have the same performance by themselves, they can help cover the blind spots of RGB cameras.

To the best of our knowledge, the idea that a sensor can "donate" its invariance to another sensor has never been discovered before, and no available theory explains it. For the particular pair of sensors used in this work, it was relatively obvious how they complemented each other, by analyzing their invariance properties to motion and illumination. And it was intuitive what was the best dataset split to consider: day and night. One can ask, more generally, given two sensor models, what are the statistics that maximally segment the dataset to increase the wormhole gain. We do not see any easy answer even in the case where the invariance is represented by a group action for both sensors.

We also see the cross-modal learning filters as a first approximation to a process that can be more nuanced. For example, the natural generalization is to allow the filters to output a scalar importance value rather than a binary decision, and use these values to weigh the importance of the samples during learning. In that case, the simple geometric argument described in Fig. 5 does not apply anymore.

The other generalization is to the case of more than two sensors; for three or more sensors there are many possible orders in which the wormhole jumps can be ordered. A more technical question is what happens if we iterate the jumps. Is it allowed to use the same data multiple times? Does the process converge to a fixed point? We smile contemplating all the interesting things that we do not know yet.

REFERENCES

[1] A. Zanardi, J. Zilly, A. Aumiller, A. Censi, and E. Frazzoli, "Wormhole learning," *IEEE International Conference on Robotics and Automation*, vol. (to appear), 2019.

[2] M. Iacono, S. Weber, A. Glover, and C. Bartolozzi, "Towards Event-Driven Object Detection with Off-the-Shelf Deep Learning." IEEE, 2018, pp. 1–9.

[3] M. Macanovic, F. Chersi, F. Rutard, S.-H. Ieng, and R. Benosman, "When conventional machine learning meets neuromorphic engineering: Deep Temporal Networks (DTNets) a machine learning frawmework allowing to operate on Events and Frames and implantable on Tensor Flow Like Hardware," *preprint arXiv:1811.07672*, 2018.

[4] G. K. Cohen, "Event-based feature detection, recognition and classification," Ph.D. dissertation, Paris 6, 2016.

[5] B. Ramesh, H. Yang, G. Orchard, N. A. L. Thi, and C. Xiang, "DART: Distribution Aware Retinal Transform for Event-based Cameras," *arXiv preprint arXiv:1710.10800*, 2017.

[6] S. B. Shrestha and G. Orchard, "SLAYER: Spike Layer Error Reassignment in Time," in *Advances in Neural Information Processing Systems*, 2018, pp. 1419–1428.

[7] P. O'Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer, "Real-time classification and sensor fusion with a spiking deep belief network," *Frontiers in neuroscience*, vol. 7, p. 178, 2013.

[8] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. [Online]. Available: http://ieeexplore.ieee.org/document/5288526/

[9] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, S. Legg, and K. Kavukcuoglu, "IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures," *arXiv preprint arXiv:1802.01561*, 2018. [Online]. Available: http://arxiv.org/abs/1802.01561

[10] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation," *preprint arXiv:1611.04558*, 2016. [Online]. Available: http://arxiv.org/abs/1611.04558

[11] C. Devin, A. Gupta, T. Darrell, P. Abbeel, and S. Levine, "Learning modular neural network policies for multi-task and multi-robot transfer," 5 2017, pp. 2169–2176. [Online]. Available: http://ieeexplore.ieee.org/document/7989250/

[12] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2242–2251, October 2017. [Online]. Available: http://arxiv.org/abs/1703.10593v3

[13] B. Frénay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 845–869, 2014.

[14] J. Bootkrajang and A. Kabán, "Label-noise robust logistic regression and its applications," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2012, pp. 143–158.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[16] S. M. LaValle, "Sensing and Filtering: A Fresh Perspective Based on Preimages and Information Spaces," *Foundations and Trends in Robotics*, vol. 1, no. 4, pp. 253–372, 2010. [Online]. Available: http://www.nowpublishers.com/article/Details/ROB-004

[17] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 x 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.

[18] C. Brandli, R. Berner, M. Yang, S. C. Liu, and T. Delbruck, "A 240 x 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.

[19] C. Posch, D. Matolin, and R. Wohlgenannt, "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*, vol. 46, no. 1, pp. 259–275, 2011.

[20] E. Mueggler, C. Forster, N. Baumli, G. Gallego, and D. Scaramuzza, "Lifetime estimation of events from Dynamic Vision Sensors," in *Proceedings - IEEE International Conference on Robotics and Automation*, June 2015, pp. 4874–4881.

[21] J. Binas, D. Neil, S.-C. Liu, and T. Delbruck, "DDD17: End-To-End DAVIS Driving Dataset," *ArXiv e-prints*, 11 2017.

[22] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman, "HATS: Histograms of Averaged Time Surfaces for Robust Event-based Object Classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1731–1740.

[23] A. Z. Zhu, D. Thakur, T. Ozaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The Multi Vehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2032–2039, 2018.

[24] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," *arXiv preprint arXiv:1707.07012*, vol. 2, no. 6, 2017.

[25] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and others, "Tensorflow: a system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[26] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," *ArXiv e-prints*, 11 2016.

[27] D. A. Van Dyk and X.-L. Meng, "The art of data augmentation," *Journal of Computational and Graphical Statistics*, vol. 10, no. 1, pp. 1–50, 2001.

[28] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.