

Low-power, Continuous Remote Behavioral Localization with Event Cameras

Friedhelm Hamann^{1,5}, Suman Ghosh¹, Ignacio Juárez Martínez²,
 Tom Hart³, Alex Kacelnik^{2,5} and Guillermo Gallego^{1,4,5}.

¹ Technische Universität Berlin, ² Oxford University, ³ Oxford Brookes University,
⁴ Einstein Center for Digital Future, ⁵ Science of Intelligence Excellence Cluster.

Abstract

Researchers in natural science need reliable methods for quantifying animal behavior. Recently, numerous computer vision methods emerged to automate the process. However, observing wild species at remote locations remains a challenging task due to difficult lighting conditions and constraints on power supply and data storage. Event cameras offer unique advantages for battery-dependent remote monitoring due to their low power consumption and high dynamic range capabilities. We use this novel sensor to quantify a behavior in Chinstrap penguins called ecstatic display. We formulate the problem as a temporal action detection task, determining the start and end times of the behavior. For this purpose, we recorded a colony of breeding penguins in Antarctica for several weeks and labeled event data on 16 nests. The developed method consists of a generator of candidate time intervals (proposals) and a classifier of the actions within them. The experiments show that the event cameras' natural response to motion is effective for continuous behavior monitoring and detection, reaching a mean average precision (mAP) of 58% (which increases to 63% in good weather conditions). The results also demonstrate the robustness against various lighting conditions contained in the challenging dataset. The low-power capabilities of the event camera allow it to record significantly longer than with a conventional camera. This work pioneers the use of event cameras for remote wildlife observation, opening new interdisciplinary opportunities. <https://tub-rip.github.io/eventpenguins/>

1. Introduction

Quantification of wildlife animal behavior is key to several disciplines from ecology [1] to conservation [2] and neuroscience [3]. Behavioral studies need to know when, where, and for how long a particular behavior occurs to understand it or to draw conclusions about animal relationships [4], feeding habits [5], welfare [6] or culture [7].

Advances in computer vision have recently led to the

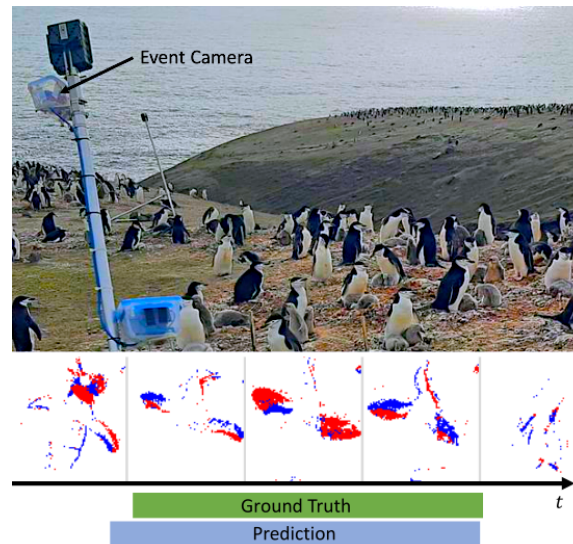


Figure 1. Remote wildlife monitoring using an event camera. Top: We recorded a colony of breeding Chinstrap penguins in Antarctica using a DAVIS346 camera. Bottom: Penguins show a behavior called “ecstatic display” (ED), visualized here using event data (positive events in blue, negative events in red). We provide the dataset with labels (Ground Truth) and investigate methods to determine the start and end times of this behavior (Prediction).

development of various systems (data acquisition and processing) for automated quantification of animal behavior [3, 8, 9]. It is a challenging topic because: (i) systems for wildlife monitoring are expected to work under varying lighting and weather conditions (snow, rain, fog), (ii) long-term observation of animals at remote places puts constraints on power usage and storage of the recording system, (iii) extracting biologically relevant information from large amounts of data is difficult and time-consuming, (iv) there is a lack of curated datasets with ground truth labels for developing robust machine learning systems.

We tackle the problem of quantifying the Ecstatic Display (ED) in Chinstrap penguins (*Pygoscelis antarctica*) during a month of their breeding season in Antarctica. The ED is a unique behavior where nesting penguins stand up-

right, point their heads upwards, beat their wings back and forth, and emit a loud call (see Fig. 1 and the accompanying video). The reasons behind this behavior are not well understood yet [4]. Thus, we aim to better understand this behavior through its large-scale automatic detection. We formulate the problem as a temporal action detection (TAD) task, which consists of localizing action instances (e.g., ED in each nest) from raw data streams (usually videos). TAD is a challenging computer vision task because it should estimate precise start and end times of the action. Moreover, TAD for ED is challenging because: (i) the ED length of different instances varies widely, in our data it ranges from approximately 1 to 40 s, (ii) the birds show similar wing-flapping behaviors that are not EDs, and are difficult to tell apart using a single snapshot (see Fig. 2). Only the availability of continuous temporal information allows us to reliably differentiate instances of ED from other wing flaps.

Current observation methods for animal behavior in the wild mostly rely on camera traps that produce a series of motion-triggered images. Long-term observation (weeks) of penguin behavior is provided by systems acquiring one image per minute. Such battery-powered systems suffer from energy constraints, which are increased by the need for illuminating with IR flashes in dim-light conditions. Neither of the above methods is suitable for analyzing fine-scale behaviors like EDs because the time resolution is insufficient or the covered observation time is too short.

We propose using event cameras for the quantification of behaviors like the ED. Event cameras [10] are novel bio-inspired sensors that output pixel-wise intensity changes instead of frames at a fixed rate. By design, they offer several advantages, such as high speed, high dynamic range (HDR), low power consumption, and data efficiency [11]. Wildlife behavior monitoring is a perfect scenario for event cameras and their natural property of highlighting movement. To this end, we used a DAVIS346 camera [12] to record a colony of Chinstrap penguins in Antarctica. The annotated data consists of 24 sequences, of ten minutes each, observing 16 penguin pairs breeding in nests. Besides providing this unique and difficult-to-acquire data, we introduce a method to detect instances of ED in two stages: generation of candidate pairs of start and end times of the behavior, and classification of the data within the pairs into ED or background.

The experiments show that the event data provides a distinct signal for ED quantification in terms of naturally capturing motion and being robust to challenging outdoor conditions. The results evidence a solid performance across a variety of tests, with a mean average precision (mAP) of 58%. A performance comparison during the night and with snow, together with a qualitative comparison with grayscale frames, show robustness to varying illumination conditions.

Our contributions are summarized as follows:

1. The first-ever use of event cameras for wild animal be-

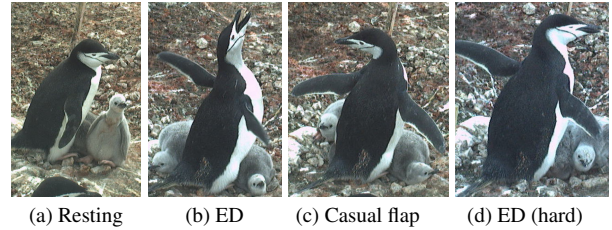


Figure 2. *Challenges in detecting Ecstatic Display (ED):* a) Penguin in rest position at its nest. b) Penguin showing an ED, a behavior where it points its head upwards and beats its wings while emitting a loud call. c) Not an ED. d) The start of an ED. Comparing c) and d) shows why it is difficult to distinguish EDs from other wing flaps using single snapshots.

havior observation. The system for energy-efficient continuous recording overcomes the limitations of previous frame-based solutions.

2. A method for the task of temporal action detection, consisting of proposal and classification stages. Both stages rely on efficient algorithms that make use of the characteristics of event data (Sec. 3).
3. An extensive (weeks long) event camera dataset of a Chinstrap penguin colony in Antarctica during the breeding season (Sec. 4). Twenty-four sequences of 10 minutes each are annotated with instances of ED on 16 nests, constituting a benchmark to foster research in this important wildlife monitoring field (Sec. 5).

To the best of our knowledge, this is the first publicly available dataset for event-based TAD, and it is also the first time for an event camera in Antarctica, which is a milestone demonstrating the viability of this technology for remote monitoring in extreme conditions. Our dataset and system contribute to advancing the knowledge about aspects of penguin behavior that would have been otherwise too costly (in both time and resources) to study.

2. Related Work

2.1. Event Cameras

Event cameras are a relatively new technology. Since the seminal work [10] they have gained increasing interest due to their appealing properties, which allow them to perform well in challenging scenarios for standard cameras, such as high speed, high dynamic range (HDR), and low power consumption. They have been studied for various applications in computer vision, like optical flow estimation, depth estimation, and SLAM. See [11] for a recent survey.

Research problems studied in the context of pattern recognition mainly target classification (e.g., recognition) [13–16] and detection [17, 18]. Due to the novelty of this imaging technology, efforts to use event cameras for pattern recognition tasks are limited by the availability of datasets.

To this end, [13] use a pan-tilt camera platform to convert the conventional frame-based dataset into an event dataset. Others provide data for gesture recognition [14] or sign language recognition [15]. Larger datasets are available in the context of automotive object classification and detection tasks [17–19]. Despite recent efforts by the research community to provide datasets for various pattern recognition tasks, there is still a lack of datasets for many applications involving a variety of environmental conditions as well as data streams longer than a few seconds.

2.2. Wildlife Observation using Computer Vision

Automated data acquisition for wildlife monitoring currently relies on recording color (i.e., RGB) images mostly through the use of camera traps and drones. Data is then hand-annotated [20], generally aided by various computer programs. Only recently has computer vision entered the field [8] with applications in counting [9, 21], species classification [22–25], detection [26, 27], individual identification [28–30] and tracking [9, 26]. Of all these, only [21] has used computer vision to count penguins, a task performed on single static images, which is simpler than our behavior monitoring problem. Other current approaches for automated behavior monitoring involve mounting and retrieval of tri-axial accelerometers to learn behavior classification [31, 32]. This invasive, non-vision approach can also monitor at-sea behavior, but it incurs a much greater effort, cost, and disruption to the animals for monitoring a similar amount of individuals for an equivalent amount of time.

2.3. Temporal Action Detection

The goal of Temporal Action Detection (TAD) is to determine the label and time interval of every action instance in an untrimmed video. It has been extensively studied for conventional RGB data. Previous methods can be roughly categorized into bottom-up and top-down approaches. Bottom-up approaches first perform frame-level classification and then merge the results into detections [33, 34]. Top-down approaches can have one [35, 36] or multiple stages [37], and usually follow a scheme where proposals are generated and fed to a classifier [38–40]. Recently, end-to-end learned methods were introduced [41–43]. TAD for conventional RGB data is a mature computer vision task, aided by large-scale datasets, such as [44, 45].

In the context of animal observation, [46] performed spatiotemporal action recognition to detect tail-biting behavior in pigs in a controlled indoor environment. The work in [47] used event-based TAD for human fall detection indoors; the method consisted of proposal and classification stages. For the proposals, they adopted an algorithm called temporal actionness grouping (TAG), introduced in [38], and used the event rate as the actionness score.

To the best of our knowledge, there is no publicly avail-

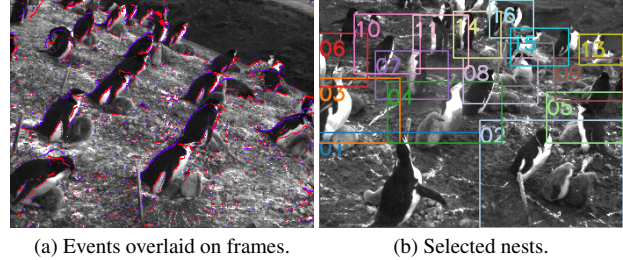


Figure 3. (a) Raw data acquired by the DAVIS346: events (in red and blue according to polarity, with μ s accuracy) overlaid on grayscale frames (1 every 8s). (b) Bounding boxes around the nests. A heat map of events over a whole sequence served as a base to hand-annotate the bounding boxes, each capturing one nest.

able dataset for animals in the wild using event cameras, a scenario with more disturbances than in indoor conditions. With this work, we demonstrate the possibilities that event cameras offer for detailed wildlife monitoring and show how event data makes it easy to determine intervals of interest for TAD with computationally efficient methods.

3. Temporal Action Detection from Event Data

This section formalizes the problem of temporal action detection (TAD) from event camera data (Sec. 3.1) and introduces our developed method (Sec. 3.2).

3.1. Problem Formulation

Let $\mathcal{E} = \{e_i\}$ be an input sequence of events corresponding to a penguin’s nest. Each event $e_i = (x_i, y_i, t_i, p_i)$ contains the pixel coordinates (x_i, y_i) , a timestamp t_i and the polarity p_i (i.e., sign) of a brightness change. Let the pixel coordinates (x_i, y_i) be already relative within the range defined by the bounding box coordinates of the nest (Fig. 3b). Let the time interval when the penguin exhibits an ecstatic display (ED) be $I = (t^a, t^b)$ (also called *action instance*), with start time t^a and end time t^b . The collection of all ground-truth action instances, $\mathcal{I}_{\text{gt}} = \{I_n\}_{n=1}^{N_{\text{gt}}}$, is the temporal annotation set of \mathcal{E} .

The goal of the temporal action detector is to generate predictions $\mathcal{I}_{\text{pred}}$ that cover \mathcal{I}_{gt} accurately and completely. The inputs of the detector are the per-nest events, and only temporal boundaries need to be detected since spatial boundaries of the nests are given. This procedure allows us to assign each ED detection reliably to a specific nest, thus enabling us to quantify the behavior of penguin couples in the nest. This step is feasible because the penguins remain in their respective nests while breeding.

3.2. Method

We follow a two-step approach (Fig. 4). First, proposals are created based on an actionness score (Sec. 3.2.1). Then, they

are classified as ED or “background” by our Augmented Temporal Segment Network (ATSN) (Sec. 3.2.2).

3.2.1 Generation of Temporal Region Proposals

The natural response of event cameras to motion is used to efficiently produce relevant proposals. Letting the event rate of \mathcal{E} be $r(t)$ (measured in events/s), in the first step, an actionness score is calculated as a normalized event rate by

$$\tilde{r}(t) = \frac{r(t) - \min(r(t))}{\max(r(t)) - \min(r(t))}. \quad (1)$$

Note that we use the robust minimum and maximum, defined by the p -th and $(100 - p)$ -th percentile, clipping $r(t)$ outside these values, and that polarity is not considered.

In the second step (ii), the intervals of high event rate (i.e., activity) are used as proposals: using the watershed algorithm [48], $\tilde{r}(t)$ can be viewed as a terrain which is “flooded” up to a water level (threshold) λ . The regions $\tilde{r}(t) > \lambda$ are used as proposals. In the third step (iii), the proposals are merged to avoid short interruptions. Starting from a proposal $\hat{I}_1 = (t_1^a, t_1^b)$, it is iteratively merged with a subsequent one $\hat{I}_2 = (t_2^a, t_2^b)$ if the ratio of the sum of individual durations over the duration of the merged proposal is above the merging threshold μ .

Lastly, steps (ii) and (iii) are performed for all combinations of thresholds $\lambda \in \{0.05, 0.1, \dots, 0.95\}$ and $\mu \in \{0.05, 0.1, \dots, 0.95\}$. The output set of interval proposals $\hat{\mathcal{I}}$ is the union of the proposals obtained from all threshold combinations. As the scheme often produces equal proposals for close threshold levels we perform non-maximum suppression (NMS) with a threshold of 0.95.

The above method, which we call **robust event-rate TAG** (reTAG), is an extension of the video-based TAG algorithm [38]. The main differences are: (i) the actionness score is given by a high-temporal resolution robustly-normalized event rate that guides the proposal search while it mitigates the influence of peak rates, and (ii) the combination of thresholds (λ, μ, NMS) is adapted to the event rate signal.

3.2.2 Proposal Classification

Overview. The input to this step are the proposed time intervals \hat{I} and the events within them, $\mathcal{E}_{\hat{I}}$. Classifying proposals brings two challenges: the varying length of the proposals and the asynchronous nature of the event data. To tackle the first challenge we leverage a common idea in action recognition and localization [49], sparsely sampling time snapshots from a proposal. For the second challenge, we convert events into a tensor-like representation, which allows us to use CNN-based learning approaches.

The main component of the classification stage is a model that we term augmented temporal-spatial network

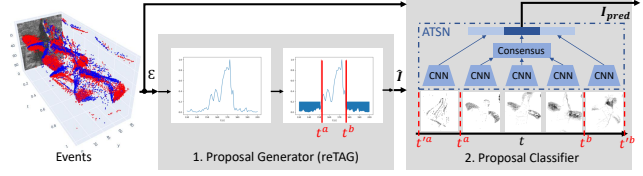


Figure 4. *Overview of the proposed method.* The input is the event data from one penguin nest (grayscale frame is not used and is only shown here for visualization). In a first step the normalized event rate is used as an actionness score to generate relevant temporal action proposals (reTAG algorithm, Sec. 3.2.1). In the second step the proposals are classified (Sec. 3.2.2). The proposals (t^a, t^b) are augmented to (t'^a, t'^b) and 2D event histograms are built at sampled timestamps within the augmented proposal duration. These are fed to the augmented temporal segment network (ATSN).

(ATSN). It works on proposals that are temporally augmented, similarly to [38], by adding “start” and “end” stages. The ATSN’s input is the sparsely sampled tensor-like representations and its output is the proposal prediction.

Augmentation. Formally, for a given proposal $\hat{I} = (t^a, t^b)$ and its duration $d = t^b - t^a$ we define two additional intervals, the start stage $\hat{I}_{\text{start}} = (t'^a, t^a)$ and the end stage $\hat{I}_{\text{end}} = (t^b, t'^b)$, where $t'^a = t^a - d/W$ and $t'^b = t^b + d/W$ (see Fig. 4), with $W = 3$ for a 33% augmentation width. The augmentation of the proposals is necessary to give the classifier information about the completeness of a proposal [38]: short proposals lying completely within an interval of an ED can be rejected based on information from the augmented intervals. Each augmented proposal consists of three consecutive intervals: $\{\hat{I}_{\text{start}}, \hat{I}, \hat{I}_{\text{end}}\}$.

Sparse Sampling. For each augmented proposal, we uniformly sample $N_{\hat{I}} = 3$ timestamps in \hat{I} and $N_{\hat{I}_{\text{start}}} = N_{\hat{I}_{\text{end}}} = 1$ timestamps in \hat{I}_{start} and \hat{I}_{end} . This step is independent of the actual duration of the proposal. Hence, the time between the timestamps can vary for different proposals.

Tensor-like representation. We convert events into tensor-like representations for compatibility with CNN networks. At each sampled timestamp t_i , a histogram [50] or a time map [51] of events \mathbf{H} is computed using a window $[t_i - \Delta t/2, t_i + \Delta t/2]$. These snapshots are fed to the ATSN.

ATSN. The ATSN receives the $N_{\hat{I}} + N_{\hat{I}_{\text{start}}} + N_{\hat{I}_{\text{end}}} = 5$ snapshots, encodes them with a backbone ResNet [52] and combines the resulting feature vectors (Fig. 4). The features from the central interval \hat{I} are aggregated via a consensus function (e.g., the mean), followed by a concatenation of the feature vectors corresponding to $\{\hat{I}_{\text{start}}, \hat{I}, \hat{I}_{\text{end}}\}$. The resulting vector is the input to the fully connected classifier. The final step is a non-maximum suppression (NMS) with a temporal Intersection over union (tIoU) threshold of 0.6.

Day	#seqs	#night	#precipitation	Mean rate [$\times 10^3$ ev/s]	Max rate [$\times 10^3$ ev/s]	Avg. #ED per seq.
Jan 5th	1	0	0	4.82	98.52	2.00
Jan 6th	2	1	0	22.68	98.52	32.00
Jan 7th	3	1	1	10.85	154.14	31.00
Jan 9th	1	0	0	11.29	28.17	28.00
Jan 11th	1	0	0	9.73	135.72	9.00
Jan 12th	5	1	3	9.01	390.75	27.00
Jan 13th	4	1	1	17.23	203.64	38.25
Jan 14th	2	1	0	16.86	154.86	4.50
Jan 15th	3	0	1	6.40	121.05	14.33
Jan 17th	1	0	0	5.76	121.83	29.00
Jan 18th	1	1	0	1.86	4.05	0.00
All	24	6	6	11.24	390.75	23.54

Table 1. Statistics of the annotated data.

4. Dataset

4.1. Data Collection

Data was collected during a scientific expedition to Deception Island, Antarctica. We recorded Chinstrap Penguins nesting at the Vapour Col/Punta Descubierta colony from Jan 5th to 30th, 2022. This covered most of the chick guarding stage and the early crèche period of the breeding season.

Available event cameras vary in resolution and provided features [11]. We chose a DAVIS346 [12], which can simultaneously record aligned events and grayscale frames (on the same array of 346×260 px). The event camera was connected to a Raspberry Pi CPU and a hard drive. To waterproof these we modified a couple of Polypropylene (PP) containers by fitting a Nikon L37 46mm glass UV filter and sealing the openings with silicone (Fig. 1). The camera was powered using a lithium-ion portable battery that needed to be changed daily, which resulted in data gaps on days when fieldwork could not be carried out due to inclement weather.

We used the in-hardware event noise reduction filter of the DAVIS and a low contrast sensitivity setting to limit the number of events produced and therefore enable long-term monitoring. We set the DAVIS to record 1 grayscale frame every 8 seconds. In total, we acquired 238 hours of data, consisting of events and frames from the DAVIS (Fig. 3a). Twenty-four 10-minute sequences were selected for annotation of ED, and their statistics are summarized in Tab. 1.

4.2. Data Annotation

We labeled 16 nests in all sequences using hand-annotated bounding boxes (Fig. 3b). Behaviors displayed outside nests or far away back were ignored. The label ID refers to the nest and not the penguin, as one nest is guarded in turns by two indistinguishable individuals. The camera shifted four times during deployment; we moved the bounding boxes accordingly for constant nest ID throughout. This is important for within-study consistency and to match results to other samples like blood or GPS tracks.

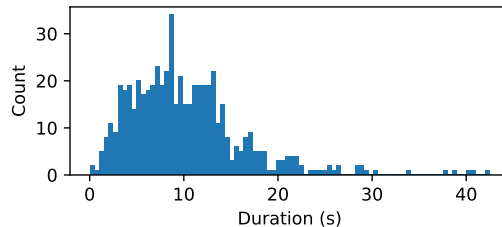


Figure 5. Histogram of the duration of the annotated EDs. Most EDs cannot be captured with previous long-term observation systems (1 frame/min). Events allow for continuous monitoring.

We then annotated the occurrence of ED behavior in twenty-four 10-minute tracks (Tab. 1). These were selected so that all hours of the day and all days in the study period were covered. An ED behavior was considered to last from the moment a penguin lifts its wings before flapping until the moment it stops and the wings are lowered, with a time resolution of 30 Hz. To keep labeling consistent, all annotations were carried out by a single penguin expert and verified by two other researchers. Figure 5 shows a histogram of the duration of the annotations. As Fig. 5 and Tab. 1 show, the number and duration of ED vary widely among the sequences, which contributes to making this a challenging detection problem.

5. Experiments

Several tests are carried out to assess the performance of the proposed method on the newly introduced dataset. This section presents the experimental settings (Sec. 5.1) as well as the experiments for the proposal generator (Sec. 5.2) and the whole pipeline (Sec. 5.3). We also evaluate the HDR capabilities of the system (Sec. 5.4), present sensitivity studies (Sec. 5.5), report the power consumption (Section 5.6) and discuss the limitations of our work (Sec. 5.7).

5.1. Experimental Settings

Implementation Details. We use a fixed data split into a training, validation, and test set (70%:10%:20%). All metrics are reported on the test data, never seen during training or validation. The test set contains five of the 24 sequences, including one night and one precipitation sequence (see supplementary for further details). The ATSN is trained on a dataset consisting of all proposals generated by the first stage (Fig. 4). Proposals with an IoU > 0.7 are labeled as positive samples, the rest as negative.

The ATSN is trained using Stochastic Gradient Descent with a momentum of 0.9. The batch size is 128, and the initial learning rate is 0.001. The backbone of the classifier is a ResNet18, which is initialized with weights pre-trained on ImageNet [53] and afterwards on the per-sample classification task. We used a weighted cross-entropy loss. The

Proposal Method	Top 20	Top 30	Top 50
Sliding Window	0.06	0.07	0.08
Watershed	<u>0.26</u>	0.27	0.27
event TAG [47]	0.24	<u>0.33</u>	<u>0.49</u>
reTAG (Ours)	0.43	0.53	0.66

Table 2. The average recall (AR), in percentage (%), for different proposal methods at the same number of proposals.

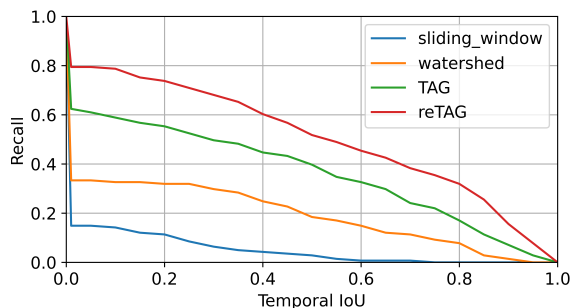


Figure 6. The Recall over IOU rates for the top-50 proposals per recording and nest.

bounding boxes of the penguin nests have different sizes. Therefore, the input event representations (histograms or time maps) are resized to 224×224 px to be passed as input to the CNN. To allow reusing the ImageNet weights, the histograms are replicated to a 3-channel input. Experiments are conducted on hosts with an Nvidia Tesla V100S GPU and an Intel Xeon 4215R CPU. The network is implemented in PyTorch 1.13.0.

Incorporating domain knowledge, we omit intervals shorter than 2s, which account for only 2.8% of the annotations. This also reduces the processing time since otherwise there are many noisy proposals.

Evaluation Metrics. We perform independent experiments for the proposal stage and the whole two-stage pipeline and evaluate using standard metrics. For the proposal stage, we report Average Recall (AR) for $tIoU = \{0.1, 0.3, 0.5, 0.7\}$ using the best N_p proposals per recording and penguin nest, with $N_p \in \{20, 30, 50\}$. For the whole pipeline, we report temporal mean average precision (mAP) at the same $tIoU$ values as for the AR metric.

5.2. Evaluation of the Proposal Generator

We compare our reTAG method to a re-implementation of the proposal method in [47] (since it is not publicly available) and two additional methods. The first additional method is a bare “watershed” algorithm corresponding to steps (i) and (ii) in Sec. 3.2.1, without an additional merging step or several threshold values. The threshold for the watershed algorithm is set to $\lambda = 0.2$, found by fitting to the training and validation sets. The second method is a sliding-

Method	0.1	0.3	0.5	0.7	Average
Bottom-up (w/o MF)	0.55	0.48	0.30	0.20	0.38
Bottom-up (w/ MF)	0.58	0.54	0.39	0.28	0.45
R3D + ActionFormer	0.72	0.65	0.52	0.32	0.55
Ours + histogram	0.63	0.61	<u>0.56</u>	0.45	<u>0.56</u>
Ours + time-map	<u>0.66</u>	<u>0.64</u>	0.58	<u>0.43</u>	0.58
<i>Perfect Classifier</i>	1.00	0.97	0.94	0.83	0.93

Table 3. Full pipeline evaluation. Mean Average Precision at several IoU levels (mAP@IoU). Best in bold. Runner-up underlined.

window approach [54] that builds proposals using all combinations of a set of start times (with stride 0.1s) and a set of window widths. As the range of durations varies greatly, we use a geometric progression to sample 30 window widths between 2 and 40s. The event rate (1) is computed with 33ms bins (30Hz equivalent) for all methods requiring it.

Table 2 shows the AR results of the different methods and Fig. 6 depicts the corresponding recall rates. Both TAG-based methods outperform the simpler methods “watershed” and “sliding window”. A direct comparison of our reTAG with the second TAG-based method shows that adding a robust minimum and maximum (percentiles $p = 1\%$) leads to significant performance gains: more than 25% for all considered numbers of proposals. The result indicates that peaks in the event rate severely affect the algorithm and removing outliers boosts performance.

5.3. Evaluation of the Full Pipeline

To evaluate the performance of the whole two-step approach in Fig. 4, we test various settings and compare the results against several baselines. We adapt the state-of-the-art frame-based method ActionFormer [42] to be used with event data and also compare against a self-designed bottom-up approach. Lastly, we provide results for a “perfect classifier” partially using ground truth information to provide an upper performance bound.

Top-down Approach (Ours). Since the method in Sec. 3 works by first analyzing a large time interval with an actionness score and then processing finer time intervals with a classifier, it is also referred to as “top-down”. The method is tested with two different tensor-like representations using the events in a window of $\Delta t = 1s$. The first representation is a 2D histogram, counting the number of events at each pixel position, and the second is an exponentially decaying time map [51] with a time constant of $\tau = 0.2s$.

Event-based ActionFormer baseline. We adapt ActionFormer [42] for use with event data. The latter works on offline-extracted video features specific to common datasets like THUMOS14 [44] and ActivityNet 1.3 [45], preventing direct adoption to custom data. Therefore we extract features in our dataset as follows: (i) batches of events are

converted to spatiotemporal voxel grids of shape (C, H, W) (height H , width W , and C time bins). Voxels are created at equidistant timesteps of 0.5s using events from a 5s time window; (ii) each voxel grid is encoded into a 1D feature vector of size 512 using a 3D-CNN (R3D [55]). The resulting sequence of feature vectors is the input to the model, which outputs the set of action instances.

Bottom-up Approach. In addition to our top-down approach of Sec. 3, we build a bottom-up approach for comparison. The event stream \mathcal{E} is converted into a sequence of synchronous frames (snapshots) similar to a video. Afterwards, per-snapshot classification is performed using a conventional CNN (e.g., ResNet). Lastly, the classification results are post-processed using morphological operators along the temporal axis. Regions of consecutive positive results are extracted as ED predictions.

Specifically, tensor-like representations $\{\mathbf{H}_1, \dots, \mathbf{H}_k\}$ are created by counting events in batches of $\Delta t = 5s$ and with a stride of 33ms (30 Hz). The synchronous classification predictions are then used for finding the temporal boundaries. This method is termed “Bottom-up (w/o MF)”. If we further apply a 1D closing morphological filter (MF) (with a kernel size of 15) we can join sparse predicted intervals; a method referred to as “Bottom-up (w/ MF).”

Perfect Classifier. This model follows a top-down approach using proposals provided by the reTAG generator. However, the classifier utilizes ground-truth information and classifies proposals as true if the tIoU is bigger than the respective metric value (0.1, 0.3, 0.5, 0.7). The classification is followed by an NMS. The values provide an upper bound for the classifier performance with given proposals.

Results. Table 3 reports the results of the abovementioned approaches. Our method using time maps (and the ResNet18 backbone) performs best (mAP= 58%), closely followed by the variant that uses histograms (mAP= 56%).

The top-down method outperforms the bottom-up method by 23%. We found $\Delta t = 5s$ works best for the bottom-up approach. While it may seem long, it is short for the simple (per-sample) classification task, indicating that the bottom-up method does not aggregate temporal information meaningfully (simply increasing Δt does not improve performance). The ATSN has information from multiple sampled timestamps and therefore yields better results.

Our method outperforms ActionFormer by 3% on average (Tab. 3). This could be due to the amount of labeled data. State-of-the-art frame-based methods in action detection (e.g., ActionFormer) rely on large amounts of data (e.g., ActivityNet $\approx 648h$). However, this is often unfeasible for use cases in conservation. Our approach relies on a non-learned first stage and a second stage with only 11M parameters (compared to 27M parameters of ActionFormer). Methods needing less data are more prone to be adopted by researchers in biology. Finally, the results show

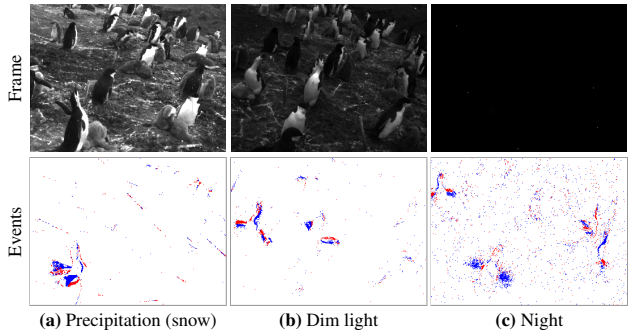


Figure 7. Examples of EDs on events and grayscale images for different lighting conditions.

Category	0.1	0.3	0.5	0.7	Average
Undisturbed	0.70	0.69	0.64	0.48	0.63
Night	0.72	0.69	0.63	0.44	0.62
Precipitation	0.49	0.41	0.34	0.25	0.37

Table 4. *Robustness to visual conditions.* Mean Average Precision at several IoU levels (mAP@IoU). Undisturbed means day time with good lighting and no precipitation.

that our method performs better for higher IoU thresholds, yielding more accurate interval boundaries. Per-nest results are given in the supplementary.

5.4. High Dynamic Range Experiment

The dataset contains samples from various lighting and weather conditions, which we use to assess the robustness of our method to impaired vision. As frame data is not available at a sufficient rate, it is not possible to directly compare a frame-based method to our event-based method. However, we perform two examinations: a qualitative comparison with the grayscale frames of the DAVIS and a quantitative comparison of our method for different conditions.

Figure 7 shows a comparison of frame data and event data at different visual conditions. The event camera signal degrades in low-light conditions and becomes significantly noisier. In comparison, the grayscale frame does not capture any information at the lowest lighting level. Table 4 shows the results on different subsets of the test set. The category “Undisturbed” shows results for the three sequences at full daylight without precipitation (the mAP increases to 63%), “Night” is the result of the night sequence and “Precipitation” is the result of the sequence containing snow. In comparison to the undisturbed case, the results show a comparable performance in night conditions (mAP= 62%) and a decrease in the snow sequence. This is remarkable, as it is apparent from Fig. 7 that grayscale frames would be unusable (i.e., not just a 1% drop) in night sequences. In summary, both examinations demonstrate the advantages

$N_{\hat{t}}$	$N_{\hat{t}_{\text{aug}}}$	Augmented	0.1	0.3	0.5	0.7	Average
3	0	✗	0.29	0.22	0.15	0.05	0.17
1	1	✓	0.53	0.47	0.40	0.19	0.40
3	1	✓	0.64	0.59	0.48	0.31	0.50
5	2	✓	0.63	0.61	0.54	0.43	0.55
7	3	✓ (Tab. 3)	0.66	0.64	0.58	<u>0.43</u>	0.58
9	3	✓	<u>0.65</u>	<u>0.62</u>	<u>0.55</u>	0.40	<u>0.56</u>

Table 5. Ablation and sensitivity study for the number of samples in the augmentation of the classification network. Mean Average Precision at several IoU levels (mAP@IoU).

W	Aug. width	0.1	0.3	0.5	0.7	Average
5	20%	0.62	0.58	0.52	<u>0.37</u>	0.52
3	33% (Tab. 3)	0.66	0.64	0.58	0.43	0.58
2	50%	<u>0.65</u>	<u>0.61</u>	<u>0.54</u>	0.37	<u>0.54</u>

Table 6. Sensitivity of the system (Ours + histogram) with respect to the augmentation width W . Mean average precision at several IoU levels (mAP@IoU).

Pretraining strategy	0.1	0.3	0.5	0.7	Average
None	0.45	0.43	0.35	0.26	0.37
Imagenet	0.52	0.50	0.45	0.34	0.45
Imagenet + Time-map	0.66	0.64	0.58	0.43	0.58

Table 7. Sensitivity to pretraining of the backbone. Mean Average Precision at several IoU levels (mAP@IoU).

of event cameras in tackling difficult illumination scenarios and the system’s robustness (novel sensor and algorithm) to natural changes in visual conditions.

5.5. Sensitivity and Ablation Studies

We assess the sensitivity of our method to multiple design choices: the amount of data fed to the classification network (Tab. 5) and the size of the augmentation width (Tab. 6). Additional sensitivities to the choice of network backbone and input representation are given in the supplementary.

Table 5 reports results for varying numbers of timestamp samples $N_{\hat{t}}$ in the ATSN. With increasing $N_{\hat{t}}$ we also increase $N_{\hat{t}_{\text{aug}}}$. The augmentation duration is fixed at 33% of the central interval duration d . The sensitivity analysis shows an increase in the performance with a higher number of samples up to $N_{\hat{t}} = 7$, which indicates that increasing the number of samples beyond that is ineffective.

Additionally, the first row of Tab. 5 reports the results for a raw TSN network without augmentation. Omitting augmentation severely limits the performance, thus highlighting the importance of augmentation, which provides information beyond the boundaries of the proposal and therefore allows the classifier to judge its “completeness”. Table 6 lists the mAP for different augmentation widths W . The results indicate an augmentation of 33% as optimal.

Table 7 shows the strong influence of pretraining.

5.6. Power Consumption

The DAVIS346 power usage is 0.7W in idle mode and 0.83W while recording. Data generation and streaming take up a small portion of the power spent by the overall device (chip, board, USB, etc.). In comparison, a frame-based camera (Basler acA1300-200u) draws 2.28W in idle mode and 2.44W while recording. The event camera allowed us to record significantly longer than the frame-based camera.

5.7. Limitations

Although our method is robust in different lighting conditions, it’s not possible to record during heavy weather. Strong wind leads to setup vibrations and noisy events.

Our detection method works under the assumption that a high event rate is linked to activity of interest. The results show that the assumption is reasonable for observing nesting penguins. However, it limits the generalization to problems where the assumption holds.

Our method relies on user-supplied regions of interest (e.g., coarse bounding boxes). The assumption is that EDs detected in one region can be associated with a specific nest. This has been carefully considered in the provided dataset. Application to unseen data requires the user to select new bounding boxes with penguins not relocating between them, although some overlap is allowed (Fig. 3b).

6. Conclusion

We have presented the first approach for wildlife monitoring using an event camera. The choice of sensor aims to drastically increase the ability to record continuously during long periods when relying on battery-based camera systems. We have introduced a unique dataset of breeding Chinstrap penguins in Antarctica acquired with a DAVIS camera, and have used it to quantify a penguin behavior called “ecstatic display” (ED). The problem has been formulated as a temporal action detection task, inferring instances of the start and end times of ED. Our two-stage detector consists of a lightweight generator of time-interval proposals and a subsequent classifier. It outperforms comparison methods and is robust against environmental changes. The results indicate that the event camera is a fit sensor for this problem, as it naturally captures motion and therefore simplifies the task, especially in difficult lighting conditions.

Acknowledgements

We thank Prof. Josabel Belliure for her help with camera deployment. The research was funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC 2002/1 “Science of Intelligence” – project number 390523135.

References

- [1] Emma Marks, Allen Rodrigo, and Dianne Brunton, “Ecstatic display calls of the adélie penguin honestly predict male condition and breeding success,” *Behaviour*, vol. 147, no. 2, pp. 165–184, 2010.
- [2] Anthony Caravaggi, Peter B Banks, A Cole Burton, Caroline MV Finlay, Peter M Haswell, Matt W Hayward, Marcus J Rowcliffe, and Mike D Wood, “A review of camera trapping for conservation behaviour research,” *Remote Sensing in Ecology and Conservation*, vol. 3, no. 3, pp. 109–122, 2017.
- [3] Talmo D Pereira, Joshua W Shaevitz, and Mala Murthy, “Quantifying behavior to understand the brain,” *Nature Neuroscience*, vol. 23, no. 12, pp. 1537–1549, 2020.
- [4] Pierre Jouventin and F Stephen Dobson, *Why penguins communicate: the evolution of visual and vocal signals*. Academic Press, 2017.
- [5] Thomas Doniol-Valcroze, Véronique Lesage, Janie Giard, and Robert Michaud, “Optimal foraging theory predicts diving and feeding strategies of the largest marine predator,” *Behavioral Ecology*, vol. 22, no. 4, pp. 880–888, 2011.
- [6] Marian S Dawkins, “Using behaviour to assess animal welfare,” *Animal welfare*, vol. 13, no. S1, pp. S3–S7, 2004.
- [7] Andrew Whiten, “The burgeoning reach of animal culture,” *Science*, vol. 372, no. 6537, p. eabe6514, 2021.
- [8] Ben G Weinstein, “A computer vision for animal ecology,” *J. Animal Ecology*, vol. 87, no. 3, pp. 533–545, 2018.
- [9] Justin Kay, Peter Kultits, Suzanne Stathatos, Siqi Deng, Erik Young, Sara Beery, Grant Van Horn, and Pietro Perona, “The Caltech fish counting dataset: A benchmark for multiple-object tracking and counting,” in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 290–311, 2022.
- [10] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck, “A 128×128 120 dB $15 \mu\text{s}$ latency asynchronous temporal contrast vision sensor,” *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.
- [11] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jörg Conrath, Kostas Daniilidis, and Davide Scaramuzza, “Event-based vision: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, 2022.
- [12] Gemma Taverni, Diederik Paul Moeys, Chenghan Li, Celso Cavaco, Vasyl Motsnyi, David San Segundo Bello, and Tobi Delbruck, “Front and back illuminated Dynamic and Active Pixel Vision Sensors comparison,” *IEEE Trans. Circuits Syst. II (TCSII)*, vol. 65, no. 5, pp. 677–681, 2018.
- [13] Garrick Orchard, Ajinkya Jayawant, Gregory K. Cohen, and Nitish Thakor, “Converting static image datasets to spiking neuromorphic datasets using saccades,” *Front. Neurosci.*, vol. 9, p. 437, 2015.
- [14] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, Jeff Kusnitz, Michael Debole, Steve Esser, Tobi Delbruck, Myron Flickner, and Dharmendra Modha, “A low power, fully event-based gesture recognition system,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 7388–7397, 2017.
- [15] Ajay Vasudevan, Pablo Negri, Camila Di Ielsi, Bernabe Linares-Barranco, and Teresa Serrano-Gotarredona, “SL-Animals-DVS: event-driven sign language animals dataset,” *Pattern Analysis and Applications*, vol. 25, no. 3, pp. 505–520, 2022.
- [16] Shu Miao, Guang Chen, Xiangyu Ning, Yang Zi, Kejia Ren, Zhenshan Bing, and Alois Knoll, “Neuromorphic vision datasets for pedestrian detection, action recognition, and fall detection,” *Front. Neurorobot.*, 2019.
- [17] Pierre de Tournemire, Davide Nitti, Etienne Perot, Davide Migliore, and Amos Sironi, “A large scale event-based detection dataset for automotive,” *arXiv preprint arXiv:2001.08499*, 2020.
- [18] Etienne Perot, Pierre de Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi, “Learning to detect objects with a 1 megapixel event camera,” *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 16639–16652, 2020.
- [19] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman, “HATS: Histograms of averaged time surfaces for robust event-based object classification,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 1731–1740, 2018.
- [20] Rahel Sollmann, “A gentle introduction to camera-trap data analysis,” *African J. Ecology*, vol. 56, no. 4, pp. 740–749, 2018.
- [21] Carlos Arteta, Victor Lempitsky, and Andrew Zisserman, “Counting in the wild,” in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 483–498, 2016.
- [22] Sara Beery, Grant Van Horn, Oisín Mac Aodha, and Pietro Perona, “The iWildCam 2018 challenge dataset,” *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, 2018.
- [23] Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L Alexander, David W Jacobs, and Peter N Belhumeur, “Birdsnap: Large-scale fine-grained visual categorization of birds,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 2011–2018, 2014.
- [24] M-E Nilsback and Andrew Zisserman, “A visual vocabulary for flower classification,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, vol. 2, pp. 1447–1454, IEEE, 2006.
- [25] Alexandra Swanson, Margaret Kosmala, Chris Lintott, Robert Simpson, Arfon Smith, and Craig Packer, “Snapshot serengeti, high-frequency annotated camera trap images of 40 mammalian species in an african savanna,” *Scientific data*, vol. 2, no. 1, pp. 1–14, 2015.
- [26] Elizabeth Bondi, Raghav Jain, Palash Aggrawal, Saket Anand, Robert Hannaford, Ashish Kapoor, Jim Piavis, Shital Shah, Lucas Joppa, Bistra Dilkina, *et al.*, “BIRDSAI: A dataset for detection and tracking in aerial thermal infrared videos,” in *IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, pp. 1747–1756, 2020.
- [27] Sara Beery, Grant Van Horn, and Pietro Perona, “Recognition in Terra Incognita,” in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 456–473, 2018.
- [28] Jason Parham, Jonathan Crall, Charles Stewart, Tanya Berger-Wolf, and Daniel I Rubenstein, “Animal population

- censusing at scale with citizen science and photographic identification,” in *AAAI Spring Symposium-Tech. Rep.*, 2017.
- [29] Jason Holmberg, Bradley Norman, and Zaven Arzoumanian, “Estimating population size, structure, and residency time for whale sharks rhincodon typus through collaborative photo-identification,” *Endangered Species Research*, vol. 7, no. 1, pp. 39–53, 2009.
- [30] Shuyuan Li, Jianguo Li, Hanlin Tang, Rui Qian, and Weiyao Lin, “ATRW: a benchmark for amur tiger re-identification in the wild,” *arXiv preprint arXiv:1906.05586*, 2019.
- [31] Stefano Chessa, Alessio Micheli, Rita Pucci, Jane Hunter, Gemma Carroll, and Rob Harcourt, “A comparative analysis of SVM and IDNN for identifying penguin activities,” *Applied Artificial Intell.*, vol. 31, no. 5-6, pp. 453–471, 2017.
- [32] GJ Sutton, Charles-André Bost, AZ Kouzani, SD Adams, K Mitchell, and JPY Arnould, “Fine-scale foraging effort and efficiency of macaroni penguins is influenced by prey type, patch density and temporal dynamics,” *Marine Biology*, vol. 168, pp. 1–16, 2021.
- [33] Zehuan Yuan, Jonathan C Stroud, Tong Lu, and Jia Deng, “Temporal action localization by structured maximal sums,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 3684–3692, 2017.
- [34] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu, “Learning salient boundary feature for anchor-free temporal action localization,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 3320–3329, 2021.
- [35] Tianwei Lin, Xu Zhao, and Zheng Shou, “Single shot temporal action detection,” in *Proc. 25th ACM Int. Conf. Multimedia*, pp. 988–996, 2017.
- [36] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei, “Gaussian temporal awareness networks for action localization,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 344–353, 2019.
- [37] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem, “G-tad: Sub-graph localization for temporal action detection,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 10156–10165, 2020.
- [38] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin, “Temporal action detection with structured segment networks,” *Int. J. Comput. Vis.*, vol. 128, no. 1, pp. 74–95, 2020.
- [39] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang, “BSN: Boundary sensitive network for temporal action proposal generation,” in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 3–19, 2018.
- [40] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen, “BMN: Boundary-matching network for temporal action proposal generation,” in *Int. Conf. Comput. Vis. (ICCV)*, pp. 3889–3898, 2019.
- [41] Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Shiwei Zhang, Song Bai, and Xiang Bai, “End-to-end temporal action detection with transformer,” *IEEE Trans. Image Process.*, vol. 31, pp. 5427–5441, 2022.
- [42] Chen-Lin Zhang, Jianxin Wu, and Yin Li, “Actionformer: Localizing moments of actions with transformers,” in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 492–510, 2022.
- [43] Feng Cheng and Gedas Bertasius, “TallFormer: Temporal action localization with a long-memory transformer,” in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 503–521, 2022.
- [44] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, “THUMOS challenge: Action recognition with a large number of classes.” <http://csrcv.ucf.edu/THUMOS14/>, 2014.
- [45] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles, “ActivityNet: A large-scale video benchmark for human activity understanding,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 961–970, 2015.
- [46] Dong Liu, Maciej Oczak, Kristina Maschat, Johannes Baumgartner, Bernadette Pletzer, Dongjian He, and Tomas Norton, “A computer vision-based method for spatial-temporal action recognition of tail-biting behaviour in group-housed pigs,” *Biosystems Engineering*, vol. 195, pp. 27–41, 2020.
- [47] Guang Chen, Sanqing Qu, Zhijun Li, Haitao Zhu, Jiaxuan Dong, Min Liu, and Jörg Conradt, “Neuromorphic vision-based fall localization in event streams with temporal-spatial attention weighted network,” *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9251–9262, 2022.
- [48] Jos BTM Roerdink and Arnold Meijster, “The watershed transform: Definitions, algorithms and parallelization strategies,” *Fundamenta informaticae*, vol. 41, no. 1-2, pp. 187–228, 2000.
- [49] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool, “Temporal segment networks: Towards good practices for deep action recognition,” in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 20–36, 2016.
- [50] Ana I. Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza, “Event-based vision meets deep learning on steering prediction for self-driving cars,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 5419–5427, 2018.
- [51] Xavier Lagorce, Garrick Orchard, Francesco Gallupi, Bertram E. Shi, and Ryad Benosman, “HOTS: A hierarchy of event-based time-surfaces for pattern recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, pp. 1346–1359, July 2017.
- [52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 770–778, 2016.
- [53] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 248–255, 2009.
- [54] Zheng Shou, Dongang Wang, and Shih-Fu Chang, “Temporal action localization in untrimmed videos via multi-stage CNNs,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.
- [55] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri, “A closer look at spatiotemporal convolutions for action recognition,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 6450–6459, 2018.