EventMASK: A Frame-Free Rapid Human Instance Segmentation with Event Camera Through Constrained Mask Propagation

Lakshmi Annamalai[1], Vignesh Ramanathan[2] and Chetan Singh Thakur[2]

***Abstract*—Human Instance Segmentation (HIS) is essential in robotics for applications such as autonomous driving and human-robot interaction, *etc*. Existing HIS solutions using conventional cameras are computationally expensive and slow. Benefits such as sparsity, high temporal resolution, *etc.* make the event camera a promising alternative. HIS with an event camera is not actively explored, though. Thus, we introduce EventMASK, a novel HIS solution that makes use of an event camera. EventMASK has been meticulously designed to process sparse raw events asynchronously, enabling low-latency processing. EventMASK employs simple statistical and probabilistic non-deep learning techniques for computational efficiency and adopts mask propagation for real-time performance. To curtail error accumulation, we present an innovative constrained likelihood-based mask updation method. EventMASK's semi-supervised approach circumvents the need for event-level instance labeling. The comprehensive analysis demonstrates EventMASK's robustness in a wide spectrum of scenarios, offering a low-cost and low-latent HIS solution for resource-constrained robotics.**

***Index Terms*—Surveillance Robotic Systems, Computer Vision for Automation, Recognition**

I. INTRODUCTION

HUMAN Instance Segmentation (HIS) has emerged as a foundational component in robotics perception in recent years. Its multi-disciplinary applications include enhancing pedestrian safety in autonomous vehicles and facilitating collaborative robots in industries, *etc*. HIS not only identifies humans but also accurately delineates their boundaries. Resource-constrained robotic systems, particularly mobile and autonomous robots, often operate with limited processing power and restricted battery energy. These limitations make computational efficiency a critical bottleneck for their smooth functioning. Furthermore, low latency is imperative to avert potential accidents in autonomous robots and enable real-time threat assessment in surveillance robots, to name a few.

Conventional cameras acquire huge volumes of data at constant intervals of time, resulting in computationally intensive and slow vision algorithms inappropriate for quick and resource-constrained robotics perception. On the other hand, a novel sensor known as an event camera captures sparse data asynchronously, triggered solely by changes in the scene. The sparseness of event data significantly reduces the computational overhead and minimizes latency, often in the order of milliseconds.

There has been considerable progress in HIS using conventional cameras. Nevertheless, extending these algorithms to event cameras poses several challenges, including the following: Current conventional camera approaches of HIS rely on synchronous, supervised deep-learning networks operating on dense $2D$ frames. Synchronous and dense processing is inefficient in handling sparse event data, resulting in unnecessary computations and increased power consumption. Additionally, deep learning-based HIS methods are time-consuming, resource-intensive, and rely on extensive labeled data. Instance-level labeling of huge amounts of data is time-consuming and expensive.

To address these challenges, we introduce EventMASK, an asynchronous, semi-supervised, non-deep learning-based method for human instance segmentation of the space-time event cloud. EventMASK operates on raw event data, which allows for reduced latency compared to frame-based systems due to the sparse and asynchronous nature of event data. This zero-latency processing is highly beneficial, especially for applications such as autonomous vehicles, which require quick decision-making.

Non-deep learning solutions are preferable in robotics due to their low computational complexity, mainly when data is scarce, as with event cameras. Therefore, we have designed EventMASK as a non-deep learning technique utilizing less-compute statistical and probabilistic methods. In the conventional camera domain, non-deep learning techniques lack the ability to capture the intricate features necessary for HIS. However, the unique ability of the event camera to sense scene changes has enabled the use of non-deep learning techniques to accomplish the complex task of HIS.

To further reduce the computation involved in mask estimation, we propose mask propagation with a novel mask parameter update module. The mask parameter update module solves the significant challenge of maintaining the predicted mask as close as possible to the true mask over time. By implementing a novel likelihood-based constrained optimization technique, Mask parameter update strikes the right balance between preserving the memory of past events and capturing the dynamics of current human instances. Additionally, by

Manuscript received: Nov, 22, 2023; Revised Feb, 14, 2024; Accepted Feb, 27, 2024.

This paper was recommended for publication by Editor Pascal Vasseur upon evaluation of the Associate Editor and Reviewers' comments

[1] Lakshmi Annamalai is with Defence Research and Development Organization and Indian Institute of Science, Bangalore, India lakshmia@iisc.ac.in

[2] Vignesh Ramanathan and Chetan Singh Thakur are with Department of Electronics Systems Engineering, Indian Institute of Science, Bangalore, India vigneshr@iisc.ac.in, csthakur@iisc.ac.in

Digital Object Identifier (DOI): see top of this page.

combining supervised human segmentation with unsupervised instance mask estimation, EventMASK eliminates the need for expensive event-level instance labeling.

II. Related Work

Human Instance Segmentation has received limited attention in the event camera domain. To present context to HIS in the event camera domain, we present a survey of the related tasks in event camera vision and instance segmentation using conventional camera [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16].

A. Event Camera Vision Tasks

Several similar domains, such as semantic segmentation, motion segmentation, human detection, and instance segmentation, have established the advantage of event cameras over frame-based cameras. Semantic Segmentation [17] [18] [19] [20] [21] [22] [23] labels each pixel with a corresponding class, without delineating the instances. Event-based human detection [24] [25] [26] accumulates events into 2D grids and detects humans as bounding boxes by the application of conventional vision networks. In contrast, the proposed EventMASK performs simultaneous semantic segmentation and detection of individual instances of humans at the event level.

Motion segmentation [27] [28] [29] [30] [31] [32] [33] [34] [35] focuses on distinguishing regions of event stream that are in motion from those that are static, without assigning semantic classes. Motion segmentation generally belongs to one of the three methods, i) Remove the background events and then analyze the remaining events [30] [29], ii) treating it as two sub-problems: event-object association and object model refinement [31] [36] [37] [35], iii) end-to-end learning-based pipeline, where events are converted into 2D format suitable for vision networks [32].

Fundamental differences between motion segmentation and EventMASK are as follows: EventMASK is a two-step process of event-class association followed by event-instance association. Although the latter process involves clustering techniques similar to motion segmentation tasks, the unique aspect of EventMASK lies in its mask model refinement tailored to human instance segmentation.

Recently, event camera-based instance segmentation has been attempted in [38] and [39]. In [38], events were converted into two-channel images, and the most popular Mask R-CNN, a deep learning-based approach, was applied for object instance segmentation. [39] employed transfer learning on event camera semantic segmentation deep learning models by unfreezing the last few layers of the encoder, full decoder, and classification layer. Unlike these dense and synchronous deep learning instance segmentation methods, EventMASK attempts HIS using sparse and asynchronous conventional processing.

B. Frame-based Instance Segmentation

Mask R-CNN [40] directly extends Faster R-CNN [41]. Authors of [42] implemented inside/outside score maps, which are utilized for detection and segmentation. Object Mask Network (OMN) [43] creates a mask by warping the features of each proposal. MaskLAB [44] proposed mask labeling built on top of Faster R-CNN, which generates semantic segmentation and instance center direction. The instance center direction is utilized to separate instances from the segmented objects. TensorMASK [45] introduced a dense sliding window technique. ShapeMASK [46] predicts a bounding box, which is then filtered and refined to produce instance masks using shape prior.

CenterMASK [16] achieves good accuracy with lesser computation with an anchor-free one-stage object detector and a novel spatial attention-guided mask. SOLO [47] introduced the concept of instant categories. SOLO was made more efficient in SOLOv2 [48]. YOLOv7 [49] has evolved into YOLOv8 [50], which offers significant advantages in terms of fast and accurate instance segmentation. Recently, transformer-based architectures have been proposed in Mask DINO [51] for instance segmentation. Mask DINO extends DINO [52] to instance segmentation task by adding a mask prediction branch.

III. proposed Solution

In this section, we will delve into the technical details of EventMASK. EventMASK is a three-step process: i) Events to Human segmentation, ii) Segmentation to Instance Mask, and iii) Instance Mask Parameter update.

A. Principles of Event Camera

Event camera generates events asynchronously and independently at each pixel (x_i, y_i) at time t_i once the intensity $I(x_i, y_i)$ changes by a pre-set threshold $\pm C$ in the logarithmic domain, with $C > 0$. The event is expressed as sequence of tuples $e_i = \{x_i, y_i, t_i, p_i\}$, where $p_i \in \{+1, -1\}$ is the polarity of brightness change.

B. Events to Human Segmentation

This is the initial step, where we identify and isolate the events of interest from the background and other objects. The architecture of our Human Segmentation is composed of two stages: spatio-temporal feature extraction and classification.

1) Spatio-Temporal Feature Extraction: Feature extraction is a fundamental step in vision tasks as it influences the accuracy of the downstream applications. In this section, we elucidate the methodology designed for extracting informative spatio-temporal features from raw event data. The stream of N event output by the event camera can be expressed as,

$$E = \{e_i \mid e_i = [x_i, y_i, t_i, p_i], i \in N\} \quad (1)$$

The event stream is a collection of spatio-temporal point clouds that contain information regarding the scene changes over time. To get a comprehensive understanding of these events, we employ a feature extraction technique that focuses on capturing spatial as well as temporal information. Specifically, we extract three types of temporal features and one spatio-temporal feature.

a) Temporal Features: The first category of temporal features concentrates on quantifying the magnitude of motion between events at a given pixel. The theoretical insight of the second and third types is given in the following sections. We define array of ∇N events that occurred at (u, v) as

$$E_{u,v} = \{e_i \mid x_i \in u, y_i \in v, i \in \nabla N\} \tag{2}$$

The temporal features are defined as follows,

$$\begin{aligned} S_{E_{u,v}} &= \frac{1}{\nabla N} \sum_{e_j \in E_{u,v}} exp^{-(t-t_j)} \\ P^{+}_{E_{u,v}} &= \frac{1}{\nabla N} \sum_{e_j \in E^{+}_{u,v}} 1 \\ P^{-}_{E_{u,v}} &= \frac{1}{\nabla N} \sum_{e_j \in E^{-}_{u,v}} 1 \end{aligned} \tag{3}$$

Where, t is the time of occurence of current event, $E^{+}_{u,v}$ and $E^{-}_{u,v}$ are the events $\{e_i \in E_{u,v} \mid p_i = +1\}$ and $\{e_i \in E_{u,v} \mid p_i = -1\}$ respectively.

b) Spatio-Temporal Feature: Our spatio-temporal feature offers a holistic view of the interplay between object and their movement. Towards spatio-temporal feature extraction, the spatio-temporal entropy of the event that occurred at (u, v) is given as,

$$H_{E_{u,v}} = - \sum_{x \in u \pm \nabla_u, y \in v \pm \nabla_v} \mathbb{P}(E_{x,y}) \log\left[\mathbb{P}(E_{x,y})\right] \tag{4}$$

Where, $\mathbb{P}(E_{x,y})$ is the probability of getting $E_{x,y}$ events at pixel location (x, y), which is estimated as the ratio between the number of events at (u, v) and total number of events generated in $(u \pm \nabla_u, v \pm \nabla_v)$.

c) Theoretical Justification for $P^{+}_{E_{u,v}}$ and $P^{-}_{E_{u,v}}$: This section brings out the theoretical foundation of the second and third types of temporal features. The polarity of events $\{\mathbf{p} : p_i | e_i \in E_{u,v}\}$ can be modeled as a Bernoulli random variable with parameter $\{pr_{+}\}$, where pr_{+} is the probability of getting polarity as $+1$ at pixel (u, v). Mapping $p_i = -1$ to 0, the Bernoulli probability of getting the sequence $\mathbf{p}$ (p_i are assumed to be independent) can be written as,

$$\mathbb{P}\left(\mathbf{p} \mid pr_{+}\right) = \prod_{n=1}^{\nabla N} pr_{+}^{p_n} (1 - pr_{+})^{1-p_n} \tag{5}$$

The log likelihood of the probability $\mathbb{P}(\mathbf{p} \mid pr_{+})$ is given as,

$$\sum_{n=1}^{\nabla N} \left[p_n \log(pr_{+}) + (1 - p_n) \log(1 - pr_{+})\right] \tag{6}$$

Differentiating and equating it to 0, it becomes,

$$pr_{+} = \frac{1}{\nabla N} \sum_{n=1}^{\nabla N} p_n \tag{7}$$

Hence, the feature $P^{+}_{E_{u,v}}$ is nothing but the probability of getting positive polarity events at the pixel (u, v)

2) Spatio-Temporal Event Classification: The extracted spatio-temporal features of each event form the basis for discerning between humans and non-humans. Toward this, we have employed a conventional binary classifier (human vs. non-human) known as Support Vector Machine (SVM). SVM is renowned for its efficiency even when presented with fewer samples.

This pipeline includes the training and inference phase. During the training phase, SVM is trained with $M \times 4$ input feature matrix, where $M \ll M_{ul}$ is the total number of labeled events and M_{ul} is the total number of unlabeled events. During the inference phase, our trained SVM model is utilized to classify each event as human or non-human. This less-compute classification enables us to identify human events as they unfold. Additionally, the ability to identify humans with a lesser number of labeled events makes our approach a valuable tool.

C. Segmentation to Instance Mask

The aim of this section is to be able to pick individual human instances from the segmented event cloud comprising humans. Given a set of segmented events E^h belonging to the human category, suppose there are C instances of humans; the aim is to split it into C set of events such that the events in E^c pertain exclusively to c^{th} human instance.

To achieve this task of localizing instance boundaries, we employ an unsupervised approach specifically designed for event data. The cost involved in instance labeling has been the motivation for selecting an unsupervised approach for this task.

The spatial characteristics of events encapsulate the necessary and sufficient information to facilitate the segregation of human events into individual instances. Hence, each event is modeled as two-dimensional features $\boldsymbol{e_i^{h_f}} = (u_i, v_i)$, which is effectively harnessed in the process of delineating the human instance masks.

Let us consider that the set of human events feature E^{h_f} are generated from C multi-variate Gaussian Mixture Model (GMM) with parameter, $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \ldots \boldsymbol{\theta}_C]$. Each $\boldsymbol{\theta}_i$ is characterized by $(\boldsymbol{\omega}_c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$, where $\boldsymbol{\omega}_c$ is the weight, $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ are the $d \times 1$ and $d \times d$ mean and covariance matrix of the c^{th} Gaussian $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$, where d is the dimension of the feature. The event features can be expressed as $\boldsymbol{e_i^{h_f}} \sim \sum_{c=1}^{C} \mathcal{N}(\boldsymbol{e_i^{h_f}} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$.

1) Estimation of Masks: The parameters $\boldsymbol{\theta}$ are estimated by maximizing the following log-likelihood using the Expectation Maximization (EM) algorithm.

$$\log\left[\mathbb{P}(E^{h_f} \mid \boldsymbol{\theta})\right] = \sum_{i=1}^{N} \log \sum_{c=1}^{C} \boldsymbol{\omega}_c \mathcal{N}(\boldsymbol{e_i^{h_f}} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \tag{8}$$

2) Propagation of Masks: The masks estimated in the previous section are propagated for future events. For every event that belongs to the human class, the proximity of the event to c masks with parameter $\boldsymbol{\theta}$ is evaluated. The event e_i is assigned to mask k if $\mathbb{P}(\boldsymbol{e_i^{h_f}} \mid \boldsymbol{\theta}_k) > \mathbb{P}(\boldsymbol{e_i^{h_f}} \mid \boldsymbol{\theta}_c)\ \forall c \in C$.

D. Instance Mask Parameter Update

This section brings out the novel method proposed to update the parameters of the instance mask. This critical step plays a pivotal role in ensuring that the masks remain aligned with the evolving characteristics of human instances as time progresses, thereby enhancing the accuracy and consistency of the instance masks. The updated parameters $\boldsymbol{\mu}_c^+$ and $\boldsymbol{\Sigma}_c^+$ of the cluster c are estimated with ∇N events $E^{{h_f}^+}$ by maximizing the following,

$$\max_{\boldsymbol{\mu}_c^+, \boldsymbol{\Sigma}_c^+} \log \left[\mathcal{N}\left(E^{{h_f}^+} | \boldsymbol{\mu}_c^+, \boldsymbol{\Sigma}_c^+ \right) \right] \tag{9}$$

with the following constraints,

$$\begin{aligned} {\boldsymbol{\mu}_c^+}^T \boldsymbol{\mu}_c &= 1 \\ trace\left[(\boldsymbol{\Sigma}_c^+)^{-1} \mathbf{B_1} \right] &= 0 \\ trace\left[(\boldsymbol{\Sigma}_c^+)^{-1} \mathbf{B_2} \right] &= 0 \end{aligned} \tag{10}$$

Where,

$$\mathbf{B_1} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{B_2} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \tag{11}$$

Where $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ are the mean and variance of the cluster c of the previous event set. The constraints enable parameter learning that injects the memory of previous events while avoiding the necessity of allocating memory to store all previous events. The first constraint updates the model such that the new $\boldsymbol{\mu}_c^+$ is as close as possible to $\boldsymbol{\mu}_c$. The second and third constraints play crucial roles in enforcing diagonal dominance in $\boldsymbol{\Sigma}_c^+$. Minimizing $trace\left[(\boldsymbol{\Sigma}_c^+)^{-1} \mathbf{B_1} \right]$ and $trace\left[(\boldsymbol{\Sigma}_c^+)^{-1} \mathbf{B_2} \right]$ encourages $\boldsymbol{\Sigma}_c^+$ to be as close as possible to diagonal matrix. This is particularly important as the features represent the spatial distribution of humans, where it is expected that the correlation between the variances of two axes will be minimal.

Combining with Langrangian multiplier, substituting $\mathcal{N}\left(E^{{h_f}^+} | \boldsymbol{\mu}_c^+, \boldsymbol{\Sigma}_c^+ \right)$ in Eq. 9, using the fact $|(\boldsymbol{\Sigma}_c^+)^{-1}| = \frac{1}{|\boldsymbol{\Sigma}_c^+|}$) and retaining only the terms which has $\boldsymbol{\mu}_c^+$ or $\boldsymbol{\Sigma}_c^+$, we get,

$$\begin{gathered} -\tfrac{1}{2} \textstyle\sum_{n=1}^{\nabla N} (\boldsymbol{e_n^{h_f}} - \boldsymbol{\mu}_c^+)^T \left(\boldsymbol{\Sigma}_c^+ \right)^{-1} \left(\boldsymbol{e_n^{h_f}} - \boldsymbol{\mu}_c^+ \right) \\ + \tfrac{\nabla N}{2} \log | \left(\boldsymbol{\Sigma}_c^+ \right)^{-1} | + \lambda_1 \left({\boldsymbol{\mu}_c^+}^T \boldsymbol{\mu}_c - 1 \right) \\ + \lambda_2 \left(trace\left[(\boldsymbol{\Sigma}_c^+)^{-1} \mathbf{B_1} \right] \right) + \lambda_3 \left(trace\left[(\boldsymbol{\Sigma}_c^+)^{-1} \mathbf{B_2} \right] \right) \end{gathered} \tag{12}$$

Differentiating with respect to $\boldsymbol{\mu}_c^+$ and equating to 0, we get,

$$\sum_{n=1}^{\nabla N} \left(\boldsymbol{\Sigma}_c^+ \right)^{-1} \left(\boldsymbol{e_n^{h_f}} - \boldsymbol{\mu}_c^+ \right) + \lambda_1 \boldsymbol{\mu}_c = 0 \tag{13}$$

The updated mean turns out to be,

$$\frac{\lambda_1 \left(\boldsymbol{\Sigma}_c^+ \right) \boldsymbol{\mu}_c}{\nabla N} + \frac{\sum_{n=1}^{\nabla N} \boldsymbol{e_n^{h_f}}}{\nabla N} \tag{14}$$

Metrics	KNN	SVM	DT	RF	MLP	NB	QDA
roc_auc	0.81	**0.86**	0.72	0.62	0.83	0.8	0.82
acc	0.88	0.96	0.9	0.9	0.96	0.88	0.91
AP	0.98	0.98	0.99	0.98	0.98	0.98	0.98
AR	0.88	0.96	0.99	0.9	0.96	0.88	0.91
F1	0.91	0.96	0.99	0.92	0.97	0.92	0.94

TABLE I: Comparison of classifiers: K nearest neighbor (KNN), SVM, Decision tree (DT), Random forest (RF), Multi-layer perceptron (MLP), Naive Baye's (NB) and Quadratic discriminant analysis (QDA) in terms of Area Under the Curve of ROC (roc_auc), accuracy (acc), average precision (AP), average recall (AR) and F1 score (F1). SVM outperforms other classifiers in terms of roc_auc.

The equation for $\boldsymbol{\mu}_c^+$ depends on $\boldsymbol{\Sigma}_c^+$. The term $\frac{(\boldsymbol{\Sigma}_c^+)}{\nabla N}$ is only a scaling factor which determines the weight given to the previous mean $\boldsymbol{\mu}_c$. Hence, it could be replaced with $\frac{(\boldsymbol{\Sigma}_c)}{\nabla N}$ or a constant.

Differentiating terms 1 and 2 of Eq. 12 with respect to $\left(\boldsymbol{\Sigma}_c^+ \right)^{-1}$ and using the properties $x^T A x = trace(\mathbf{x}\mathbf{x}^T \mathbf{A})$, $\frac{\partial}{\partial \mathbf{A}} trace(\mathbf{AB}) = \mathbf{B}^T$, $\frac{\partial}{\partial \mathbf{A}} \log |\mathbf{A}| = (\mathbf{A}^T)^{-1}$, we get,

$$\frac{\nabla N}{2} \boldsymbol{\Sigma}_c^+ - \frac{1}{2} \sum_{n=1}^{\nabla N} (\boldsymbol{e_n^{h_f}} - \boldsymbol{\mu}_c^+)(\boldsymbol{e_n^{h_f}} - \boldsymbol{\mu}_c^+)^T \tag{15}$$

Differentiating terms 4 and 5 of Eq. 12 with respect to $\left(\boldsymbol{\Sigma}_c^+ \right)^{-1}$, we get $\lambda_2 \mathbf{B_1} + \lambda_3 \mathbf{B_2}$. Combining Eq. 15 with this, equating it to 0 and rearranging the terms, we get,

$$\begin{gathered} \tfrac{\nabla N}{2} \boldsymbol{\Sigma}_c^+ + \lambda_2 \mathbf{B_1} + \lambda_3 \mathbf{B_2} - \\ \tfrac{1}{2} \textstyle\sum_{n=1}^{\nabla N} (\boldsymbol{e_n^{h_f}} - \boldsymbol{\mu}_c^+)(\boldsymbol{e_n^{h_f}} - \boldsymbol{\mu}_c^+)^T \end{gathered} \tag{16}$$

$\boldsymbol{\Sigma}_c^+$ is estimated from the above equation as follows,

$$\frac{1}{\nabla N} \sum_{n=1}^{\nabla N} (\boldsymbol{e_n^{h_f}} - \boldsymbol{\mu}_c^+)(\boldsymbol{e_n^{h_f}} - \boldsymbol{\mu}_c^+)^T - 2\lambda_2 \mathbf{B_1} - 2\lambda_3 \mathbf{B_2} \tag{17}$$

IV. Experiments and Results

This section presents the comprehensive results with relevant metrics, plots, and visualizations. The experiments performed are classified into four categories, each focusing on validating specific aspects of our approach: i) analysis of events to the human segmentation module, ii) evaluation providing insight into the effect of the number of instances, iii) performance of the proposed mask parameter update model and iv) comparison with conventional HIS to highlight the benefits of the proposed HIS pipeline.

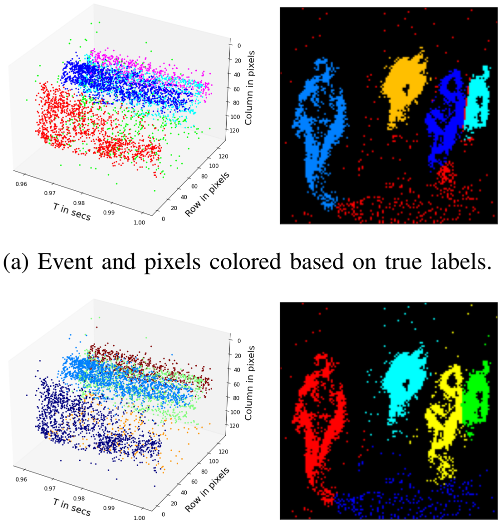
A. Analysis of Human Segmentation Module

This section provides an evaluation and analysis of the performance of the initial module, which is responsible for converting events into human segmentation. The classifier is trained using features extracted from the events that occurred between $0ms$ and $24ms$, and the remaining events in each sequence are utilized for testing. The performance of this module was rigorously evaluated on different sequences and classifiers, in terms of roc_auc (Area under the curve of Receiver Operating Characteristics) accuracy, recall, precision, and F1-Score (Table. I). Although most classifiers exhibit comparable performance, SVM stands out for its superior behavior with respect to roc_auc.

The results demonstrate the remarkable ability of the human segmentation module to work with an exceptionally minimal number of events. Furthermore, it confirms that binary segmentation tasks, such as human segmentation, can be effectively accomplished using a cost-effective, non-deep learning approach.

B. Qualitative Analysis

This section provides the visual results of EventMASK on the SPIDER dataset (Fig. 1). Key insights emerged from visual analysis (more results provided in supplementary) give an overview of the performance of EventMASK under variations in the number and size of humans, cluttered environment, etc.



(a) Event and pixels colored based on true labels.

(b) Event and pixels colored based on predicted labels.

Fig. 1: Visual results of EventMASK on SPIDER dataset [53]. Event data (left) and $2D$ grid created from events (right)

C. EventMASK vs. Number of Instances Parameter

The number of humans present in the given sequence of events is initially unknown, yet the number of instances is an input parameter to the algorithm. Consequently, the following scenario may arise: The specified number of instances may exceed the actual number of instances and vice versa. Therefore, it is mandatory to analyze the performance of the proposed method in relation to the number of initialized instances. To address this, we conducted a study on the curated N-MuPeTS [54] dataset, where the true number of instances was between 2 to 4. We investigated the performance of the proposed algorithm when the number of instances specified ranged from 4 to 20.

Top and bottom rows of Table. II respectively give an analysis of IS Module and EventMASK averaged across all the sequences in terms of recall, precision, and mIOU. Analysis of the IS module involved the estimation of metrics with respect to its input events, while the EventMASK analysis involved metrics estimation with respect to events emitted from the event camera. Upon visual inspection, it was observed that the noise in the data got split into multiple clusters when the specified number of instances exceeded the true number of instances. Consequently, there was no apparent deterioration of mIOU when the value of the specified number of instances was higher. However, it is important to note that higher values of specified instances will result in increased computation.

Another important observation from the Table is the trade-off between recall and precision achieved by varying the value of the number of instances parameter. With an increase in the number of specified instances, the proposed method tends to become more conservative in its predictions, leading to a steady decrease in recall with an increase in precision. The right balance between recall and precision, and thereby the number of instances, can be determined based on the specific requirements of the application.

D. Mask Parameter Update Model of EventMASK

To validate the mathematical update model proposed in section III-D, we thoroughly investigated the performance of EventMASK under two scenarios: i) proposed mask update model vs. mask re-estimation, ii) proposed mask update model vs. a simple mask update model.

1) Proposed Mask Update vs. Mask Re-estimation: The results presented in this section provide valuable insights into the performance of the IS Module and EventMASK in terms of the proposed mask update model vs. mask re-estimation. The mask re-estimation window determines how often the mask is estimated from scratch. During the intermediate duration, the mask is propagated along the time axis. In our analysis, we varied the mask re-estimation window from 24ms to $5 * 24$ms in increments of 24ms.

Key findings of the analysis are as follows: Table. III indicates that the mask estimation window has a significant impact on the performance of the method, with a more frequent estimation of masks yielding improved performance. However, this improvement comes at the expense of increased computational cost. For a set of ∇N events, the complexity involved in mask re-estimation is $O(m\nabla NCd^3)$ (only dominant computation given), where m is the number of iterations, while the complexity of mask propagation is notably lower (results provided in section IV-E1). Hence, the observed trade-off between performance and computational complexity serves as critical guidance for practical implementation. Applications with stricter computational constraints may opt for larger windows. This insight facilitates the adaptation of the proposed method to specific robotic applications.

2) Proposed Mask Update vs Simple Mask (SM) Update: Table. IV compares the proposed mask update model (Eq. 17 and 14) with a simple mask update model that updates the parameters of the masks as the average of the current and previous parameters. Experiments are conducted on the Hotel-Bar [55] dataset (details are provided in supplementary). Since

# Instances	AR	AP	mIOU
4	0.99	0.5	0.49
8	0.9	0.7	0.61
12	0.82	0.79	0.63
16	0.75	0.82	0.59
20	0.7	0.86	0.58

(a) Quality 1 (IS Module)

# Instances	AR	AP	mIOU
4	0.99	0.38	0.37
8	0.94	0.54	0.49
12	0.89	0.64	0.55
16	0.84	0.69	0.56
20	0.79	0.73	0.54

(b) Quality 2 (IS Module)

# Instances	AR	AP	mIOU
4	0.96	0.74	0.71
8	0.79	0.9	0.7
12	0.66	0.93	0.61
16	0.55	0.95	0.52
20	0.47	0.95	0.45

(c) Quality 3 (IS Module)

# Instances	AR	AP	mIOU
4	0.8	0.5	0.45
8	0.72	0.7	0.53
12	0.65	0.79	0.53
16	0.58	0.82	0.49
20	0.55	0.86	0.48

(d) Quality 1 (EventMASK)

# Instances	AR	AP	mIOU
4	0.72	0.38	0.32
8	0.68	0.54	0.4
12	0.64	0.64	0.43
16	0.6	0.69	0.43
20	0.55	0.73	0.42

(e) Quality 2 (EventMASK)

# Instances	AR	AP	mIOU
4	0.74	0.74	0.58
8	0.6	0.9	0.55
12	0.5	0.93	0.47
16	0.42	0.95	0.4
20	0.36	0.95	0.35

(f) Quality 3 (EventMASK)

TABLE II: Analysis of IS module (top row) and EventMASK (bottom row) vs. specified number of instances in terms of Average Recall (AR), Average Precision (AP), and mIOU. With an increase in the specified number of instances, there were no observed occurrences of human instance splitting. With an increase in the number of instance masks, recall dropped, whereas precision increased. Depending on the specific requirements of recall vs. precision, the number of instances could be chosen.

Time (ms)	AR	AP	mIOU
24	0.91	0.77	0.7
48	0.92	0.58	0.53
72	0.92	0.5	0.46
96	0.93	0.45	0.42
120	0.94	0.43	0.39

(a) Quality 1 (IS module)

Time (ms)	AR	AP	mIOU
24	0.87	0.69	0.58
48	0.85	0.64	0.53
72	0.88	0.49	0.41
96	0.92	0.4	0.33
120	0.93	0.34	0.3

(b) Quality 2 (IS module)

Time (ms)	AR	AP	mIOU
24	0.96	0.74	0.71
48	0.94	0.72	0.68
72	0.96	0.57	0.54
96	0.97	0.52	0.49
120	0.98	0.45	0.44

(c) Quality 3 (IS module)

Time (ms)	AR	AP	mIOU
24	0.75	0.76	0.6
48	0.76	0.57	0.47
72	0.76	0.49	0.4
96	0.76	0.44	0.37
120	0.76	0.43	0.35

(d) Quality 1 (EventMASK)

Time (ms)	AR	AP	mIOU
24	0.61	0.69	0.45
48	0.6	0.64	0.42
72	0.63	0.49	0.33
96	0.66	0.4	0.27
120	0.67	0.34	0.25

(e) Quality 2 (EventMASK)

Time (ms)	AR	AP	mIOU
24	0.74	0.74	0.58
48	0.72	0.72	0.55
72	0.74	0.57	0.45
96	0.74	0.52	0.42
120	0.75	0.45	0.38

(f) Quality 3 (EventMASK)

TABLE III: Performance of IS module (top row) and EventMASK (bottom row) with respect to mask parameter update model vs. mask re-estimation. The higher the mask re-estimation rate, the higher the mIOU. However, this comes at the cost of increased computation. Based on the trade-off between accuracy and complexity, either the mask re-estimation or mask parameter update can be adapted.

Update Method	AR	AP	mIOU
Proposed	0.79	0.90	0.71
SM	0.75	0.85	0.63

TABLE IV: analysis of proposed mask update model vs. simple mask update (SM) model reveals that the proposed model displays better performance in terms of recall, precision, and mIOU.

the dataset contained only humans, the human segmentation module was not included in the pipeline. It also highlights the flexibility to add or exclude modules as needed, thereby reducing computational overhead. The number of masks was initialized to 10. Masks were re-estimated every 200k event with a set of 10k events and propagated via mask parameter update model (proposed and simple models) during intermediate events. Note the increased recall, precision, and mIOU of the proposed mask update model compared to the simple update model.

E. Comparison with conventional HIS

This section brings out the low latency and low computing benefits offered by EventMASK compared to the conventional HIS pipeline. Latency and computation are influenced by the input mode used in the two pipelines and the algorithm employed. Experiments are carried out on SPIDER [53] dataset (details provided in supplementary), which consists of event recordings as well as conventional camera frames.

1) Frame-based Algorithm vs. EventMask: The state-of-the-art conventional camera HIS algorithms such as YOLOv8-V8 [50], Mask R-CNN [40], Mask-DINO [51], CenterMask [16], SOLOv2 [47] *etc.* are compared with EventMASK in terms of GFLOP, processing time and mIOU (which is estimated as explained in supplementary).

Frames are generated at the rate of 30 frames per second, i.e., once every 33 ms. The average number of events generated within 33 ms is 608 (estimated based on the total number of events that occurred during the entire sequence). Consequently, the GFLOP and running time have been estimated per frame for conventional HIS and every 608 event for EventMASK.

The number of instances of EventMASK is initially set to 2 and later re-initialized to 4 after $260 * 5$k events when there

Method	**Time (secs)**	**GFLOP**	**mIOU**
Proposed	0.0358	6.657×10^{-3}	0.36
YOLOv8 [50]	0.3021	110.2	0.32
Mask R-CNN [40] [40]	1.3141	177.589	0.35
Mask DINO [51]	4.2936	283.2882	0.34
CenterMask [16]	1.3297	435.1077	0.29
SOLOv2 [48]	1.0567	122.4566	0.26

TABLE V: Comparison of the state-of-the-art conventional HIS pipelines with the proposed event-based HIS on SPIDER dataset [53]. The table provides the run time and GFLOP for the mask update pipeline of EventMASK, whereas it turns out to be 0.054 secs and 9.55×10^{-3} GFLOP for mask re-estimation pipeline. Notably, EventMASK performs at par with conventional HIS pipelines while demonstrating lower computation and latency. Further reduction in run time could be achieved with optimized implementation.

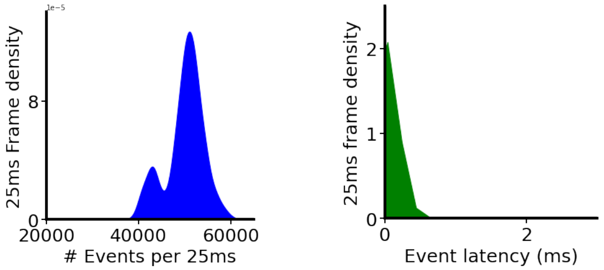


Fig. 2: Frame-based vs. event-driven computing. Left: Number of events occurring within 25ms interval vs. the density of the number of 25ms window. In event-driven computing, within an interval of 25ms, processing ranges from 0 to an average of 50k events, while frame-based computing processes 491520 pixels always, regardless of the scene dynamics. Right: Average time resolution of events vs. the density of the number of 25ms windows with the given resolution. Event-driven computing offers a time resolution of as low as 1ms, while frame-based computing introduces a constant latency of 25ms (frame rate)

was a significant increase in the number of events. Mask is re-estimated every 25k event while propagated through Eq. 14 and 17 during the intermediate events.

Experiments were performed on a Ubuntu 20.04.4 machine with an AMD EPYC 7742 64-Core Processor running at 1500 MHz and 64GB of RAM. GFLOP of conventional HIS is estimated through APIs provided by deep learning frameworks, whereas FLOP of EventMASK is calculated (with approximations) in the traditional way of counting the number of multiplications and additions. Table V confirms the superiority of the proposed EventMASK over the conventional HIS pipeline in terms of latency and compute.

2) Frame-based vs. Event-driven Computing: Conventional HIS includes frame-based processing, which is synchronous and dense processing. In contrast, EventMASK adapts event-driven processing, which is asynchronous and sparse, with operations triggered by events often involving smaller amounts of data. This section quantitatively validates the benefits of EventMASK achieved via event-driven processing.

a) Efficient Computing: This section demonstrates the efficient computation achieved by event-driven processing. Towards this, we analyzed the N-MuPeTS dataset in terms of the number of events $|E|$ occurring in the frame interval of 25ms vs. the density of the number of 25ms window with $|E|$ events (left plot of Fig. 2). Observe the significant variation in the number of events depending on the information content of the scene. Consequently, the number of events to be processed by EventMASK is directly influenced by the dynamics of the scene, while frame-based processing entails processing all 491520 pixels ($M \times N$, where M and N represent the number of rows and cols of frames, respectively) throughout, regardless of the scene dynamics. This leads to substantial power savings, particularly in situations where the scene has minimal dynamic objects.

b) Low Latency: This section substantiates the low-latency effect of event-driven processing, which processes events as they occur. Towards this, we estimated the latency of events $E_{\Delta t}$ vs. the density of the number of 25ms window with the given latency (right plot of Fig. 2). Event latency is estimated as the average of the time duration between the occurrence of events every 25ms. It is evident that most 25ms windows have a latency as low as 1ms, whereas frame-based processing has a latency of 25ms (frame rate). Event-driven computing of EventMASK, thus, enables real-time response capabilities.

V. Conclusion

In this paper, we introduced EventMASK, a low-cost and low-latent HIS designed for robotics applications utilizing an event camera. EventMASK operates on raw event data, ensuring minimal latency. Leveraging the unique sensing dynamics of the event camera, EventMASK achieved HIS through non-deep learning techniques, thereby significantly reducing the computational complexity. Extensive testing of individual modules and the complete EventMASK pipeline proved its ability to accurately segment human instances. Furthermore, the experiments also revealed the significance of the proposed parameter update model of EventMASK. Comparison with conventional HIS approaches highlights the low-compute and low-latency benefits of EventMASK. Notably, EventMASK exhibited robust performance without necessitating instance-level labeling of events. As part of future work, we aim to extend EventMASK to accommodate moving event cameras.

References

[1] Z. Huang, Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring r-cnn," *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[2] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "Blendmask: top-down meets bottom-up for instance segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[3] R. Zhang, Z. Tian, C. Shen, M. You, and Y. Yan, "Mask encoding for single shot instance segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[4] E. Xie, W. Wang, M. Ding, R. Zhang, and P. Luo, "Polarmask++: enhanced polar representation for single-shot instance segmentation and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[5] C. Fu, M. Shvets, and A. Berg, "Retinamask: learning to predict masks improves state-of-the-art single-shot detection for free," *arXiv preprint*, 2019.

[6] M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," *IEEE Conference on Computer Vision and Pattern Rcognition*, 2017.

[7] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother, "Instancecut: from edges to instances with multicut," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[8] B. RomeraParedes and P. Torr, "Recurrent instance segmentation," *European Conference on Computer Vision*, 2016.

[9] A. Arnab and P. Torr, "Pixelwise instance segmentation with a dynamically instantiated network," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[10] M. Ren and R. Zemel, "End-to-end instance segmentation with recurrent attention," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[11] S. Liu, J. Jia, and S. Fidler, "Sgn: Sequential grouping networks for instance segmentation." *IEEE International Conference on Computer Vision*, 2017.

[12] X. Liang, L. Lin, Y. Wei, X. Shen, J. Yang, and S. Yan, "Proposal-free network for instance-level object segmentation." *IEEE Transactions on Pattern Analysis and Maching Intelligence*, 2017.

[13] V. Iglovikov, S. Seferbekov, A. Buslaev, and A. Shvets, "Ternausnetv2: fully convolutional network for instance segmentation." *IEEE Conference on Computer Vision and Pattern Recognition.*, 2018.

[14] Y. Zhou, Y. Zhu, Q. Ye, Q. Qiu, and J. Jiao, "Weakly supervised instance segmentation using class peak response." *IEEE Conference on Computer Vision and Pattern Recognition.*, 2018.

[15] N. Gao, Y. Shan, Y. Wang, X. Zhao, Y. Yu, M. Yang, and K. Huang, "Ssap: single-shot instance segmentation with affinity pyramid." *IEEE International Conference on Computer Vision*, 2019.

[16] Y. Lee and J. Park, "Centermask: real-time anchor-free instance segmentation." *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[17] F. Barranco, Teo, C. Fermuller, and Y. Aloimonos, "Contour detection and characterization for asynchronous event sensors," *IEEE International Conference on Computer Vision*, pp. 486–494, 2015.

[18] A. Marcireau, Ieng, C. SimonChane, and R. Benosman, "Event-based color segmentation with a high dynamic range sensor," *Frontiers on Neuroscience*, 2018.

[19] F. Barranco, C. Fermuller, and E. Ros, "Real-time clustering and multi-target tracking using event-based sensors," *IEEE International Conference on Intelligent Robots and Systems*, 2018.

[20] I. Alonso and A. Murillo, "Ev-segnet: Semantic segmentation for event-based cameras," *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[21] L. Wang, Y. Chae, Yoon, Kim, and Yoon, "Evdistill: Asynchronous events to end-task learning via bidirectional reconstruction-guided cross-modal knowledge distillation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[22] Z. Sun, N. Messikommer, D. Gehrig, and D. Scaramuzza, "Ess: Learning event-based semantic segmentation from still images," *European Conference on Computer Vision (ECCV)*, 2022.

[23] Biswas, Kosta, Liyanagedera, Apolinario, and Roy, "Halsie: Hybrid approach to learning segmentation by simultaneously exploiting image and event modalities," *IEEE Winter Conference on Applications of Computer Vision*, 2024.

[24] Z. Jiang, P. Xia, K. Huang, W. Stechele, G. Chen, Z. Bing, and A. Knoll, "Mixed frame-/event-driven fast pedestrian detection," *IEEE International Conference on Robotics and Automation*, 2019.

[25] G. Chen, H. Cao, C. Ye, Z. Zhang, X. Liu, X. Mo, Z. Qu, J. Conradt, F. Röhrbein, and A. Knoll, "Multi-cue event information fusion for pedestrian detection with neuromorphic vision sensors," *Frontiers on Neurorobotics*, 2019.

[26] F. Ojeda, A. Bisulco, D. Kepple, V. Isler, and D. Lee, "On-device event filtering with binary neural networks for pedestrian detection using neuromorphic vision sensors," *IEEE International Conference on Image Processing*, 2020.

[27] A. Mishra, R. Ghosh, Principe, Thakor, and Kukreja, "A saccade based framework for real-time motion segmentation using event based vision sensors," *Frontiers on Neuroscience*, 2017.

[28] V. Vasco, A. Glover, E. Mueggler, D. Scaramuzza, L. Natale, and C. Bartolozzi, "Independent motion detection with event-driven cameras," *International Conference on Advanced Robotics*, pp. 530–536, 2017.

[29] Stoffregen, Timo, and K. Lindsay, "Simultaneous optical flow and segmentation (sofas) using dynamic vision sensor," *arXiv preprint arXiv:1805.12326*, 2018.

[30] Mitrokhin, Anton, F. Cornelia, P. Chethan, and A. Yiannis, "Event-based moving object detection and tracking," *IEEE International Conference on Intelligent Robots and Systems*, 2018.

[31] T. Stoffregen, G. Gallego, T. Drummond, L. Kleeman, and D. Scaramuzza, "Event-based motion segmentation by motion compensation," *IEEE International Conference on Computer Vision*, 2019.

[32] A. Mitrokhin, C. Ye, C. Fermüller, Y. Aloimonos, and T. Delbruck, "Ev-imo: Motion segmentation dataset and learning pipeline for event cameras," *IEEE International Conference on Intelligent Robots and Systems*, 2019.

[33] A. Mitrokhin, Z. Hua, C. Fermüller, and Y. Aloimonos, "Learning visual motion segmentation using event surfaces," *IEEE Computer Vision and Pattern Recognition*, 2020.

[34] D. Kepple, D. Lee, C. Prepsius, V. Isler, I. Park, and D. Lee, "Jointly learning visual motion and confidence from local patches in event cameras," *European Conference on Computer Vision*, 2020.

[35] Y. Zhou, G. Gallego, X. Lu, S. Liu, and S. Shen, "Emsgc: Event-based motion segmentation with spatio-temporal graph cuts," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[36] G. Gallego, H. Rebecq, and D. Scaramuzza, "A unifying contrast maximization framework for event cameras with applications to motion, depth and optical flow estimation," *IEEE Conference on Computer Vision and Pattern Recognition.*, 2018.

[37] C. Parameshwara, N. Sanket, A. Gupta, C. Fermuller, and Y. Aloimonos, "Moms with events: Multi-oject motion segmentation with monocular event cameras," *arXiv: 2006.06158*, 2020.

[38] A. Baltaretu, "Ev-mask-rcnn: Instance segmentation in event-based videos ." *Theses*, 2021.

[39] Huang, Xiaoqian, K. Sanket, A. Abdulla, N. Fariborz, Baghaei, M. Dimitrios, and Z. Yahya, "A neuromorphic dataset for tabletop object segmentation in indoor cluttered environment," *Scientific Data*, 2024.

[40] K. He, G. Gkioxari, P. Doll, and R. Girshick, "Mask r-cnn," *IEEE International Conference on Computer Vision*, pp. 2961–2969, 2017.

[41] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.

[42] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[43] Z. Hayder, X. He, and M. Salzmann, "Boundary-aware instance segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[44] L. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, and H. Adam, "Masklab: Instance segmentation by refining object detection with semantic and direction features," *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[45] X. Chen, R. Girshick, K. He, and P. Dollár, "Tensormask: a foundation for dense object segmentation," *IEEE International Conference on Computer Vision*, 2019.

[46] W. Kuo, A. Angelova, J. Malik, and T. Lin, "Shapemask: learning to segment novel objects by refining shape priors," *IEEE International Conference on Computer Vision*, 2019.

[47] X. Wang, K. Tao, S. Chunhua, J. Yuning, and L. Lei, "Solo: segmenting objects by locations." *European Conference on Computer Vision*, 2020.

[48] Wang, Xinlong, Z. Rufeng, K. Tao, L. Lei, and S. Chunhua, "Solov2: dynamic and fast instance segmentation." *arXiv preprint*, 2020.

[49] Wang, ChienYao, B. Alexey, and L. HongYuan, Mark, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

[50] "Yolov8," *https://github.com/ultralytics/ultralytics*, 2023.

[51] Li, Feng, Z. Hao, X. Huaizhe, L. Shilong, Z. Lei, N. Lionel, M, and S. HeungYeung, "Mask dino: Towards a unified transformer-based framework for object detection and segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

[52] H. Zhang, L. Feng, L. Shilong, Z. Lei, S. Hang, Z. Jun, N. Lionel, M, and S. HeungYeung, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," *arXiv:2203.03605*, 2022.

[53] Axenie and Cristian, "Event-based vision datasets for pedestrian detection in urban scenarios." *https://zenodo.org/records/10102416*, 2023.

[54] Bolten, Tobias, N. Christian, P. Regina, and Tönnies, "N-mupets: Event camera dataset for multi-person tracking and instance segmentation." *VISIGRAPP*, 2023.

[55] Guo, Shasha, and T. Delbruck, "Low cost and latency event camera background activity denoising." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 785–795, 2023.