Bringing Online Egocentric Action Recognition Into the Wild

Bringing Online Egocentric Action Recognition into the wild

Gabriele Goletto*,[1], Mirco Planamente*,[1,2,3], Barbara Caputo[1,3], and Giuseppe Averta[1]

***Abstract*—To enable a safe and effective human-robot cooperation, it is crucial to develop models for the identification of human activities. Egocentric vision seems to be a viable solution to solve this problem, and therefore many works provide deep learning solutions to infer human actions from first person videos. However, although very promising, most of these do not consider the major challenges that comes with a realistic deployment, such as the portability of the model, the need for real-time inference, and the robustness with respect to the novel domains (i.e., new spaces, users, tasks). With this paper, we set the boundaries that egocentric vision models should consider for realistic applications, defining a novel setting of egocentric action recognition in the wild, which encourages researchers to develop novel, applications-aware solutions. We also present a new model-agnostic technique that enables the rapid repurposing of existing architectures in this new context, demonstrating the feasibility to deploy a model on a tiny device (Jetson Nano) and to perform the task directly on the edge with very low energy consumption (2.4W on average at 50 fps). The code is publicly available at: https://github.com/EgocentricVision/EgoWild.**

***Index Terms*—Deep Learning for Visual Perception; Deep Learning Methods; Human-Robot Collaboration**

I. Introduction

Current robotics research demonstrated an increasing interest in the development of technologies to support the physical interaction between humans and machines, ranging from the planning and control [1], up to their social impact [2]. However, the deployment of this technology in the real world requires an extension of the human intention retrieval capabilities of robots, from a mere pose estimation and forecast, to an high level description of the action executed. To reach this goal, a very promising solution relies on the use of egocentric vision, in which the human activity is recorded by wearable cameras placed on the head of the user [3]. This setting comes with the benefit that source data are characterised by a rich multi-modal information, thanks to the proximity of audio/video sensors to the action scene, and by an intrinsic embedding of an attention mechanisms that stems from the human gaze direction itself.

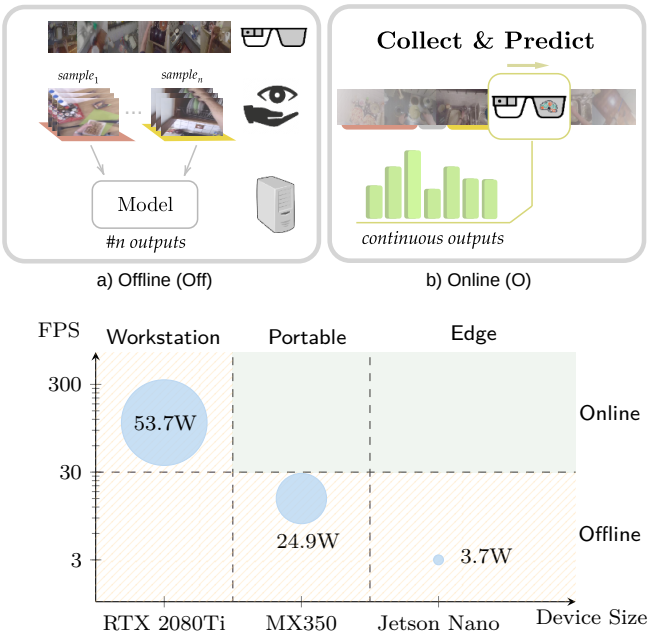

Fig. 1. **Top:** Comparison between offline (a) and online (b) inference protocol for first person action recognition (FPAR). **Bottom**: Frames per Second (FPS) processed with the I3D model [4] on different devices. The ▨ areas show traditional action recognition models' difficulty to run online inference on edge devices, either due to latency or hardware constraints. Our goal is to promote research toward models that can work in the ■ area, allowing egocentric models to run online inference and on tiny devices.

Although many works in the literature have provided solutions to infer knowledge about human activity from egocentric data (a.k.a. First Person Action Recognition, FPAR), this is frequently achieved through very large neural architectures without regard to their computational demand (see Fig. 1, bottom part). As a consequence, although very accurate, most of the models presented in literature are not suitable for realistic usecases, where real-time inference (Fig. 1, online scenario) should be performed on board of low-power hardware to enable wearability, avoid data transfer and preserve privacy. The goal of this paper is to encourage a new line of research based on realistic

Manuscript received: October 26, 2023; Revised December 13, 2022; Accepted February 8, 2023.

This paper was recommended for publication by Editor Cesar Cadena upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by the Italian Ministry of University and Research (DM1061), the IIT HPC infrastructure for the availability of high performance computing (Franklin) and CINECA award under the ISCRA initiative. This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

*The authors equally contributed to this work.

[1] Dipartimento di Automatica e Informatica, Politecnico di Torino, Corso Duca degli Abruzzi, 24, 10124 Torino, Italy `name.surname@polito.it`

[2] Italian Institute of Technology, Genova, Italy

[3] Consortium Cini, Italy

Fig. 2. Examples of two frames taken from different videos sources with the same label ("mix").

egocentric vision use cases. We propose a new FPAR benchmark with real-world constraints, which consists of altering current action recognition protocol to follow a set of realistic limitations that we add progressively (model size, cross domains, online, and untrimmed).

In addition, we propose a model-agnostic technique to enable a fast re-purposing of existing architectures in this new context. Our approach consists of two components: an anomaly detection-based solution for action boundary localization, followed by a two-fold aggregator strategy. The first solution is based on the assumption that *if I can recognize an action, I can also localize it*. Considering that traditional training of the action recognition framework is done with trimmed data containing single actions, the embedding that arises from multiple actions will be very different from the standard one, and as a consequence, the network will be able to detect it as an anomaly. The second solution is introduced to cope with the large proportion of overlapping segments in fine-grained action recognition that make it harder to localize concurrent actions.

To summarize, this paper contributes with:

- the definition of a new setting of FPAR in the wild, which encourages researchers to develop applications-aware solutions;
- a benchmark of popular action recognition models for real-world application in FPAR;
- a method to enable the use of existing features extractors to achieve efficient yet accurate action recognition under constraints, exploiting an anomaly detection strategy to localize the boundary of the actions and a two-fold aggregator solution to deal with concurrent actions in a continuous stream;
- an analysis of performance on an edge device, opening interesting perspectives for on-board intelligence.

II. Related Works

First Person Action Recognition (FPAR). The main architectures utilized in this context are generally inherited from third-person literature and may be classified into two broad categories: 2D convolution -based [5], [6], [7], [8] and 3D convolution -based [4], [9], [10], [11], [12]. The first group is generally complemented with other modules such as LSTM or its variations [13], [14], [15], Temporal Shift Module (TSM) - a parameter-free channel-wise temporal shift operator presented in [6], or the Temporal Relation Network module (TRN) [7]. The use of 3D convolutions was proposed as an alternative in [4], [10] to learn spatial and temporal relations simultaneously, even if they often introduce more parameters, requiring pre-training on large-scale video datasets [4].

The complex nature of egocentric videos raises a variety of challenges, such as ego-motion [16], partially visible or occluded objects, and environmental bias [17], [18], [19], which limit the performance of traditional approaches when used in FPAR [20], [21]. Those challenges attract the community's interest and motivate the design of novel and more complex architectures, often based on multi-stream approaches such as [13], [22], [23], [14].

Action segmentation and detection. Action segmentation [24], [25], [26], [27], [28], [29] and detection [30], [31], [32] can be intended as the extension of action recognition to the more complex scenario of untrimmed videos, where the task is to assign an action label to each frame, identifying non-overlapping (for segmentation) and overlapping (for detection) action segments. Most of these tasks require large and offline models, especially for the EPIC-KITCHENS challenge solutions*, in which the network uses the entire video as input to infer the action. This makes state-of-the-art models unsuitable for our purpose, where on-line processing is fundamental. Recently, [33] developed a novel unsupervised methodology for event boundary localization (detection) that outperforms current approaches while increasing inference time significantly, making it perfect for edge devices.

III. Bringing FPAR in the wild

To really enable the deployment of egocentric vision models, it is fundamental to consider a variety of constraints in terms of energetic, memory and temporal budget. The first (and foremost) of these is the amount of resources required to perform the task, namely the memory size to store model parameters and input data, and the number of operations (e.g. MACs) required to perform inference. The first is a constraint imposed by the flash memory of the device, while the second is related to the micro-controller velocity in inference, and to the frame-rate required by the task.

The input specification is another important feature to consider for real-world applications. In this regard, the goal is to find a good trade-off between: i) the amount of information needed as input to properly encode the temporal information; ii) the corresponding memory increase for storing input data on the device; and iii) the critical fact that, unlike the spatial dimension, the temporal dimension is presented as a continuous stream, which prevents an efficient sub-sampling and requires online processing. Another important aspect to consider when posing real-time constrains is that, in the context of egocentric vision, many techniques attain notable results only by leveraging non-real-time secondary modalities such as the optical flow. Although this modality is highly successful, it has a high computational cost [34], [35], which

*https://epic-kitchens.github.io/Reports/EPIC-KITCHENS-Challenges-2021-Report.pdf

prevents its use in real-time applications, and increases the size of the model.

It is also worth reporting that, because the sensor is worn by the user - usually at the head level - it records data with a high degree of variation produced by rapid changes in environment, perspective, and illumination as in Fig 2. Input variability can cause a difference in the distribution of data between the training and testing phases. This results in a problem known as *environmental bias* or *domain shift* that can negatively impact the performance of the model. Studying the network capability to generalize across domains provides clues on how the model will perform in a real scenario (where domain shifts are present).

The last point of interest for a real deployment of egocentric technologies in the wild lies in the intrinsic untrimmed nature of input data. Indeed, the vast majority of works of action recognition assume that the input clips are "trimmed" around the action of interest, which clearly represent an invasive form of supervision not available in realistic settings. Therefore, we argue that, despite recent progress in the area, trimmed action recognition has limited relevance in real-world scenarios, while continuous video flows with no previous knowledge on action location in time are the primary input source to be considered.

Model size, online recognition, robustness across domains, and untrimmed data source represent the constraints that realistic usecases pose, and - to the best of our knowledge - no work in the literature investigates general solutions appropriate for this setting. In this paper, beyond proposing a new line of research, we investigate a solution to bring existing FPAR models to perform online action recognition without introducing further training, thereby promoting the repurposing of existing models.

A. Benchmarking FPAR with real-world constraints.

As a first step, we tackle the model footprint issue and assess the impact of the model reduction by comparing it with popular action recognition networks, testing their generalization capabilities in seen and unseen settings (domain shift). Then, following an increasing complexity order, all real-world restrictions are added sequentially (Streaming, Online, and Untrimmed).

Backbone. To assess the effects of model footprint on task accuracy, we considered several 2D-CNN and 3D-CNN models for action recognition, which are often used in the context of egocentric vision, including I3D [4], TSN [36], TSM [6] and TRN [7]. These typical action recognition architectures are compared with two families of NAS-based models which optimize model efficiency: [37] and [38]. From these, we considered for our purposes the smallest versions, named X3D-XS and MoViNet-A0 (with and without buffer) respectively. We tested the backbones on both seen and unseen data distributions (e.g. different environments). Albeit this is very often omitted, testing across domains is fundamental to assess the generalization capabilities of models and can highlight overfitting occurrence on specific data distributions.

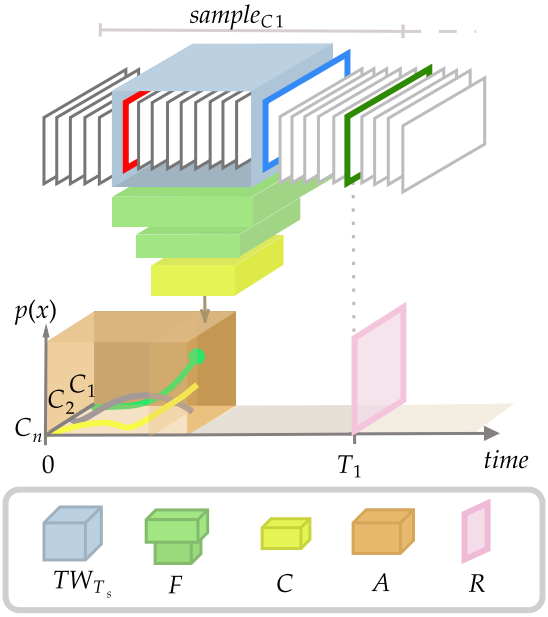

Fig. 3. Illustration of the Streaming inference scenario. TW_{T_s} represents a temporal window sliding along the video with stride 1. At each time step, a clip of T_s contiguous frames is fed into the network, which comprises a feature extractor F and a classifier C with n classes (C_1, C_2, ... ,C_n). A represents the aggregator that - at each step - updates the output of the network, taking into consideration the current output and the previous ones. R stands for aggregator cleaning, triggered by the sample's last frame.

Offline, Streaming, and Online. As anticipated before, the standard action recognition inference protocol usually works in an offline fashion, exploiting the supervision information on the edges of the action (start and end) to take the right input to process. To do this, specific sampling strategies are needed to reduce the amount of input data and avoid that a sample x of T frames cause a model to exceed its memory budget. Most works address this issue relying on uniform sampling, i.e picking T_s equidistant frames, with $T_s < T$. This method is the preferable solution for video understanding, but suffers of two major drawbacks: i) it assumes the knowledge of samples length in advance (which is not the case for continuous streams); and ii) uniform sampling completely filters out information related to the action velocity. Other works, instead, rely on dense sampling, selecting a set of T_s contiguous frames. In some cases, this choice penalizes the model due to the fact that its temporal receptive fields may see only a limited portion of the action. Indeed, the final prediction is usually obtained by averaging the predictions of different equidistant clips over the whole video, performing video level uniform sampling, i.e. requiring the sample's length information.

The artificial limits of offline inference approaches are alleviated in two novel cases. The first, hereinafter named *Streaming inference*, still assumes the knowledge of action boundaries but enables the processing of a continuous input stream (see Fig. 3). Intermediate outputs are continuously collected with an aggregator (A in Fig. 3) which is then used to obtain a final prediction. When the action is completed (i.e. at the final frame), the aggregator is flushed to reset the

model for novel predictions. Removing the supervision on action boundaries as well (i.e. no prior knowledge on when to reset the aggregator), we introduce the *Online inference* setting, where the model is asked to identify both actions and their (rough) temporal edges. The complexity of this setting requires to deal with untrimmed data when actions are alternated with "unknown" clips. In our experiments, we studied the online inference settings with and without "unknown" clips, to verify how their presence affects the final performance.

IV. Methodology

A. From single clips to continuous data streams

We extended the offline inference approach to deal with continuous streaming input by using a sliding window (TW_{T_s}) with a unitary stride that selects T_s dense frames progressively (see Fig. 3). For each time sample, the oldest frame (red in Fig. 3) is removed and replaced by a new one (blue in Fig. 3), and a new inference is performed and accumulated with the previous ones. Then, a continuous output is obtained with an aggregator strategy (aggregator(A) in Fig 3). MoViNet implements its aggregator by replacing 3D convolution with the (2+1)D operation and exposing a stream buffer mechanism to cache feature activations, allowing the temporal receptive field to expand without the need for recomputation. To support frame-by-frame output and exploit the buffer mechanism, it uses Causal Convolutions and Cumulative Global Average Pooling. The first one is used to make the convolutions unidirectional along the temporal dimension. The second one, instead, approximates any global average pooling involving the temporal dimension. For models lacking specific aggregator mechanisms, we implemented a continuous averaging of the corresponding temporal window's output. Each aggregator is empty at the beginning of each sample and is resetted (R) at the end.

B. Actions boundaries localization

As anticipated before, action recognition models are trained to classify well-separated actions taken as input. Transferring this capability to continuous video flows comes with the difficulty that the model may be asked to infer from clips that do not necessarily contain separate and complete actions. Therefore, the continual encoding of successive actions results in an overall increase in prediction uncertainty and instability in time. The anomaly detection literature [39] describes this behavior as a consequence of the fact that the network processes data with a pattern that does not conform to the defined notion of *normal* data learned during training. Therefore, the presence of concurrent or unknown action can be seen as an anomaly in time. Based on this consideration, we implemented a Dynamic Boundary Localization (DBL) strategy - with almost no overhead in terms of model size and latency - to localize the boundary of an action by examining the continuous stream of extracted features.

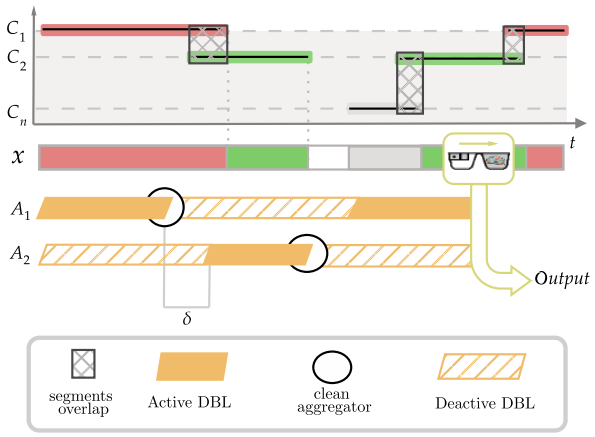

Fig. 4. Illustration of the proposed two-fold aggregator (A^2) method. The two aggregators work asyncronsly, δ is a parameter used to guarantee the asyncroncity of the two and indicates the frame-delay of the DBL activation of one aggregator when the other one detects an anomaly.

More specifically, since cross-entropy loss (de facto standard for FPAR) promotes class representations to be well separated in feature space [40], it is possible to use a distance metric (e.g. Mean Square Error) between the features extracted to measure their variations caused by action changes. Therefore, it is possible to identify action boundaries in a continuous data flow by looking for abnormalities in feature distribution over time while treating all frames of the same action as *normal*. This method could not only reveal differences between known actions but also detect the presence of "unknown" segments of video (e.g., background).

However, it is important to note that in case of overlapping segments (which e.g. in the EPIC-Kitchens dataset reaches up to 28.1% of the total clip), the detection of the new class could be delayed or anticipated with respect to the current action. Since the aggregator solution can encode only one action at a time, the network's inference will favor one of the two consecutive actions.

In light of the above considerations, we can state that for fine-grained action recognition, the standard aggregator may be ineffective. To solve this problem, we introduced a two-fold aggregator strategy (A^2). The two aggregators (A_1 and A_2 in Fig. 4) run asynchronously using a mixed boundary detection approach, allowing the encoding of the next action before the previous one finishes. When one aggregator detects an anomaly, it disables its DBL and activates the DBL of the second aggregator. To guarantee asynchrony in the moment of the anomaly's detection, we delay the activation of the second aggregator's DBL by an hyperparameter δ.

The final output is obtained as:

$$\mathcal{O} = n_1 A_1(x) + n_2 A_2(x) \tag{1}$$

where $A_i(x)$ is the output of the *i-th* aggregator for the input x and n_i corresponds on the quantity of frame processed by the *i-th* aggregator.

V. Implementation

Dataset. In our experiments, we utilize the top three kitchens with the most labeled samples from the EPIC-Kitchens-55 dataset [20]. These kitchens are referred to as D1, D2, and D3. We have chosen this specific setting as it is the standard and widely used dataset for cross-domain analysis in first-person perspective [12], and it also provides rich multi-modal information, including audio and event data [35], which can be beneficial for further analysis. Additionally, the difficulties in this dataset arise not only from the significant domain shift among different kitchens, but also from imbalanced class distribution both intra- and inter-domain.

Input. Experiments with I3D [4] and X3D [37] are conducted by sampling one random clip from the video during training and 5 equidistant clips spanning across all the video during test, as in [17]. The number of frames composing each clip is 16. For TSN [36], TSM [6] and TRN [7] architectures, uniform sampling is used, consisting of 5 frames uniformly sampled along the video. During testing, 5 clips per video are adopted, following the experimental protocol proposed in [6]. For MoVinet [38], dense sampling is adopted, with 4 consecutive clips composed by 8 frames, randomly taken from the video during training as in the original work. All the architectures follow the standard video data augmentation as in [5], the spatial input resolution has been kept consistent with the pretrained models (182 for X3D, 172 for MoViNet and 224 for the others) while the temporal resolution for all the models has been set to 30 fps.

Implementation Details. We adopted the original I3D network proposed in [4] with Inception-V1 as inflated backbone, while we chose to use X3D-XS and MoViNet-A0 to have the most efficient models from the two families. The optimizer is SGD with momentum of 0.9, weight decay 10^{-7} and a starting learning rate η of 0.01. I3D has been trained for a total of 5000 iterations, the learning rate decays by 0.1 at step 3000. Instead, MoViNet-A0 and X3D-XS have been trained for 1500 iterations without learning rate decay. For all the experiments we adopted a batch size of 128. For the two-fold aggregator implementation, we estimated the value of $\delta = 20$ directly from the dataset (a subset of kitchens from [21] not used in this paper) by calculating the average length of action overlaps (at 30 Hz).

Evaluation Protocols. In this part, we discuss the evaluation protocol we used for our benchmark.

Seen $\rightleftharpoons$ Unseen. For the seen results we train on kitchen D_i and test on the same ($D_i \rightarrow D_i$), $i \in \{1,2,3\}$. We evaluate performance on unseen test by training on D_i and testing on D_j, with $i \neq j$ and $i,j \in \{1,2,3\}$ ($D_i \rightarrow D_j$).

Offline, Streaming, and Online. We refer to *offline* to indicate the standard action recognition inference protocol, which typically uses as input a sub-sample of the input frames. We perform experiments using both uniform and dense sampling. The term *streaming inference* refers to experiments where the test is performed using all the frames (to simulate the continuous stream of the data that comes from a wearable device) with the supervision on the action's boundary (start and end) to properly clean the aggregator. The *online inference* setting, instead, assumes no supervision on action limits, and to effectively deal with this scenario the model should automatically detect both action and their boundaries.

Trimmed $\rightarrow$ Untrimmed. Moving from the trimmed to the untrimmed scenario, the lack of mutually exclusive temporal separation from the actions found in the original dataset makes it difficult to calculate an accuracy per frame. At the same time, the precise timestamp of start and end in fine-grained action is complex and extremely subjective. For this reason, we use accuracy as a metric to validate our experiments, putting the focus on the ability to recognize the action when it happens instead of the precise localization of its boundaries. For simplicity, the performance is still computed at the end of each action, and the "unknown" segments are not used in the evaluation of the performance but only in the event boundary localization part. In other words, we did not use the "unknown" class during the evaluation of the accuracy, but the network should be able to manage it during the clean phase of the buffer or the logits accumulation.

TABLE I
Top-1 *mean* accuracy (%) of different common-use architectures, over all $D_i \rightarrow D_j$ combinations on both seen and unseen test sets in *offline-trimmed* setting.

EPIC-Kitchens 55				
Network	Sampling	Params	Seen	Unseen
TSN	U 5x5	10.7M	60.88	31.55
TSN-TRN	U 5x5	18.3M	63.13	32.42
TSM	U 5x5	24.3M	**71.48**	35.97
TSM-TRN	U 5x5	-	69.52	36.05
I3D	U 16x5	12.4M	67.34	**43.89**
I3D	D 16x5	12.4M	67.08	42.42
X3D-XS	U 5x5	3.8M	51.46	36.39
X3D-XS	D 16x5	3.8M	48.45	32.66
MoViNet-A0	U 5x5	3.1M	62.17	39.25
MoViNet-A0	D 16x5	3.1M	64.17	40.68

TABLE II
Top-1 *mean* accuracy (%), over all $D_i \rightarrow D_j$ combinations on both seen and unseen test sets in both *offline-trimmed* setting and *streaming-trimmed* setting

EPIC-Kitchens 55				
Network	Mode	Sampling	Seen	Unseen
I3D	Offline	D 16x5	67.08	42.42
X3D-XS	Offline	D 16x5	48.45	32.66
MoViNet-A0	Offline	D 16x5	64.17	40.68
I3D	Streaming	All Stream	**63.38**	**40.57**
X3D-XS	Streaming	All Stream	43.37	32.31
MoViNet-A0	Streaming	All Stream	62.24	39.59

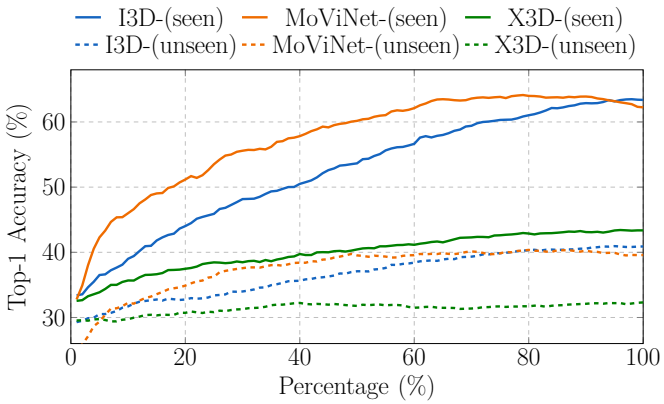

Fig. 5. MoViNet, X3D and I3D performance with respect to the percentage of video observed

VI. Experiments

In Table I, we compare two families of recently designed tiny-networks with popular architectures used in action recognition, examining various factors such as different pretrains, sampling methodologies, and amount of params. Then, in Table II, we analyze the performance in the streaming scenario, displaying a plot of the models' accuracy vs the portion of the video observed (Fig 5). Fig 6 shows the effects of the action detection algorithms used to move from the streaming to the online inference scenario. Finally, in Fig 7, we test the performance with untrimmed data, demonstrating how our two-fold aggregator (A^2) method grants a more robust solution with a little increase in parameters. The Table III illustrates the impact of performance in terms of latency, power consumption, and other critical characteristics for a designed device.

Impact of footprint on model accuracy. MoViNet and X3D are the two tiny architecture included in our benchmark to compare smaller models w.r.t standard action recognition networks. Interestingly, for X3D the tiny model size appears to have a negative impact on the final results, showing the lowest accuracy. It also suffers significantly from the transition from uniform to dense sampling (U→D). MoViNet, on the other hand, appears to be the preferable alternative, showing more notable results in both seen and unseen settings. Noteworthy, we also observed higher robustness to the shift in sampling from uniform to dense (U→D). All those considerations motivate our focus on MoViNet in this work.

The importance of seen-unseen accuracy. In contrast with the standard benchmark in action recognition, in our analysis we conduct experiments considering two different scenarios: seen and unseen. Indeed, looking only at the performances in the seen scenario, it seems that MoViNet obtains lower results compared to the TSM (62.45% and 71.48% respectively). Instead, when tested on unseen data distribution, we have a significant gain in performance of MoViNet w.r.t. TSM and I3D. In particular, the MoViNet results with uniform or dense strategy are quite similar (39.25% and 40.68% respectively). It is also worth noticing that, with more frames, MoViNet results in unseen scenarios improve considerably (see Table II).

Offline → Streaming. Table II shows the results in these two distinct settings. It is interesting to observe that MoViNet is the model that better exploits the continuous stream of data, obtaining the smallest deterioration in performance equal to 2% and 1% in seen and unseen scenarios respectively, whereas the other two networks show a much bigger decrease in performance. This behavior is caused by the buffer implementation used in the MoViNet streaming version, which enables the simulation of a receptive field as large as the entire input video, while processing frames one-by-one ($T_s = 1$). On the contrary, I3D and X3D take as input block of 16 frames ($T_s = 16$), which requires the recomputation of overlapping frames activations and may limit the total efficiency of the models.
Streaming → Online. As discussed before, the standard action recognition protocol assumes available the knowledge of the action boundary as a prior-knowledge for the correct restart of the averaging output, to obtain video level prediction for architectures such as I3D or X3D, or to properly reset the buffer mechanism for MoViNet. In other words, "cleaning" the prior encoding for the new one is necessary to produce an accurate prediction for the current action. At this stage of our investigation, we assess how much the typical action recognition architectures rely on the action's boundary and how their performance is affected by the absence of this supervision knowledge.

Dependency from the actions boundary. In Fig. 5 we plot the accuracy of the models as a function of the percentage of the video observed. From this chart, we notice that the use of the last portion of the video does not provide a gain in accuracy, and after the 85% of the video, no substantial improvement is obtained. Similar observations can be made for the initial part of the video. Interestingly, the performances of the tiny model X3D in the initial part of the observed video are very close to the final one, revealing a tendency to privilege appearance information with respect to motion information. Instead, the performance gap of MoViNet and I3D from the first portion of the video observed and after viewing 60%–80% of the data, confirms that their prediction is based more on the motion. This behavior is consistent with the more robust results in unknown conditions (unseen), where the appearance-based solution suffers more due to the fact that the appearance characteristics of the scene (texture, light condition, etc.) changes more among the environments with respect to the motion.

Effects of no supervision on actions boundaries. The loss of knowledge on actions boundaries requires a solution to automatically identify action changes. In this section, we discuss the performance of the strategy presented in section IV-B, comparing the results with a static solution (Fig. 6). The latter is based on the "naive" assumption that all sample lengths are nearly equivalent, and as a result, it assumes that a new action is "discovered" at each k frame. For both the solutions, we report a sensibility analysis on the number of frames for the static solution (SBL), and on the threshold value for the dynamic one

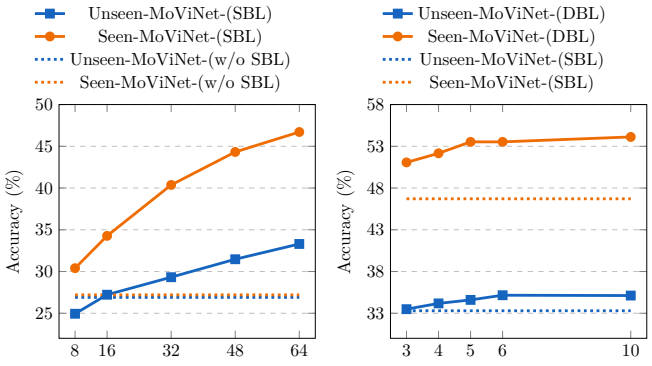

Fig. 6. Top-1 *mean* accuracy (%), over all $D_i \rightarrow D_j$ combinations on both seen and unseen test sets in *online-trimmed* setting. Left) **Static boundary localization (SBL)** with different values for the clean buffer. Right) **Dynamic boundary localization (DBL)** with different threshold-values.

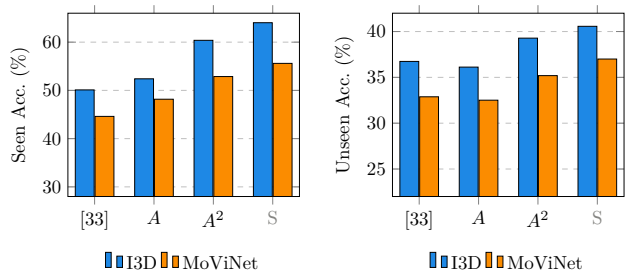

Fig. 7. We present the results in an *online-untrimmed* scenario using three different approaches: ABD [33] as a secondary stream to identify the boundaries, our DBL technique with a single aggregator (A), and our DBL technique with a two-fold aggregator (A^2). We also report the results obtained in a streaming scenario (S) as an upper bound reference.

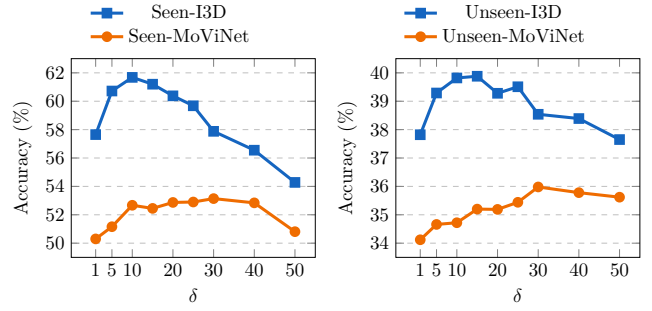

Fig. 8. Analysis of the effects of the delay parameter δ. We report Top-1 *mean* accuracy (%) in *online-untrimmed* setting.

(DBL), in both seen and unseen settings. According to Fig. 6, MoViNet with a low-loaded aggregator is unreliable; indeed, the results are lower than those without a clean one. Furthermore, by raising the buffer load, i.e., forwarding more frames, it increases its performance. A significant improvement of the dynamic strategy over the static one is also noticeable in Fig 6. Moreover, MoViNet performance appears to be not sensitive to proper threshold values; indeed, the improvements of the DBL solution are always better compared to the best results of the SBL solution.

Trimmed → Untrimmed. In Fig 7 we show the results in an untrimmed online scenario. We compare the performance of our DBL approach with single (A) and two-fold (A^2) aggregator, to the recently proposed technique ABD [33], exploiting it as a secondary stream to identify the boundary and provide the action boundary to the primary model of classification. For ABD, we used the original online implementation, with both NMS and filter windows size equal to 50. Furthermore, we report, as a reference, untrimmed streaming (S) results, i.e., experiments in which the real boundary of the action is used as prior knowledge. We present the performance of the DBL technique and two-fold aggregator using I3D to demonstrate that the proposed approach is scalable and model agnostic. Indeed the improvement of A^2 is remarkable and the results obtained for both the architecture are comparable with the streaming scenario. Moreover, the solution with a single aggregator performs similarly to the competitor ABD, without using a secondary stream for the boundary localization. Finally, the improvements of our solution A^2 with respect to the ABD are consistent across scenarios and models. To provide a comprehensive analysis, we conducted an ablation study on the delay hyperparameter δ. The results are presented in Fig. 8 and confirm that estimating δ as the average overlap of actions at the desired frame rate is a reliable approach.

TABLE III
MACs, FPS (Hz), Latency (ms, inference time) and Energy(watt) on different devices.

On Device					
Network	Device	MACs	FPS	Lat.(ms)	Power(watt)
I3D	2080 Ti	$270e^8$	110	9.1	53.7
MoViNet	2080 Ti	$0.47e^8$	781	1.3	52
I3D	MX350	$270e^8$	**15**	65.7	24.9
MoViNet	MX350	$0.47e^8$	169	5.9	11.5
I3D	Jetson Nano	$270e^8$	**3**	393.7	3.7
MoViNet	Jetson Nano	$0.47e^8$	**56**	17.9	**2.4**

Edge Deployment. In Table III we show MACs, FPS Latency and Energy on different devices. These metrics are obtained from models deployed on each different hardware through the usage of TensorRT. Power is measured with a power meter, subtracting the static power. This analysis focuses on how hardware constraints affect the applicability of the existing model for action recognition on real device. Indeed, when the I3D model moved from a high-performance GPU (2080 Ti) to a laptop GPU (MX350) and, to an edge device (NVIDIA Jetson Nano), it used more energy, falling short of the required FPS threshold for identifying human motion (up to 20-30 FPS [41]). Instead, in the case of MoViNet, the minimal number of model parameters ensures appropriate FPS (twice as needed), allowing the use of two-fold aggregator technique in online inference scenario.

VII. Conclusions

The purpose of this work is to investigate and highlight the limitations that mainstream egocentric vision models show in realistic usecases, where computational time and

power are limited. We promote a new line of research for FPAR, which considers real-world application limits such as hardware restrictions, cross-domain scenarios, and online inference on untrimmed data.

We provide: i) a new benchmark to assess the challenges of real world deployment, and ii) a novel approach capable to bring FPAR models on low-power devices (edge computing), tackling the presence of overlapping actions and the absence of supervision on action boundaries for real world usage. In light of the challenges discussed in this work, we encourage future researchers to devote attention to designing innovative approaches that allow real-time adaptation of the model on the edge during the processing of untrimmed videos, particularly in the presence of changes in environmental conditions.

References

[1] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, 2018.

[2] A. Henschel, R. Hortensius, and E. S. Cross, "Social cognition in the age of human–robot interaction," *Trends in Neurosciences*, 2020.

[3] I. Rodin, A. Furnari, D. Mavroeidis, and G. M. Farinella, "Predicting the future from first person (egocentric) vision: A survey," *Computer Vision and Image Understanding*, 2021.

[4] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.

[5] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016.

[6] J. Lin, C. Gan, and S. Han, "Tsm: Temporal shift module for efficient video understanding," in *Proc. Int. Conf. Comput. Vis.*, 2019.

[7] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, "Temporal relational reasoning in videos," in *Proc. Eur. Conf. Comput. Vis.*, 2018.

[8] A. Cartas, J. Luque, P. Radeva, C. Segura, and M. Dimiccoli, "Seeing and hearing egocentric actions: How much can we learn?," in *Proc. Int. Conf. Comput. Vis. Workshops*, 2019.

[9] S. Singh, C. Arora, and C. Jawahar, "First person action recognition using deep learned descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[10] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015.

[11] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Int. Conf. Pattern Recog.*, 2019.

[12] J. Munro and D. Damen, "Multi-modal domain adaptation for fine-grained action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020.

[13] S. Sudhakaran, S. Escalera, and O. Lanz, "Lsta: Long short-term attention for egocentric action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019.

[14] A. Furnari and G. Farinella, "Rolling-unrolling lstms for action anticipation from first-person video," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[15] M. Planamente, A. Bottino, and B. Caputo, "Self-supervised joint encoding of motion and appearance for first person action recognition," in *Int. Conf. Pattern Recog.*, IEEE, 2021.

[16] Y. Li, Z. Ye, and J. M. Rehg, "Delving into egocentric actions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015.

[17] J. Munro and D. Damen, "Multi-modal domain adaptation for fine-grained action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020.

[18] M. Planamente, C. Plizzari, E. Alberti, and B. Caputo, "Domain generalization through audio-visual relative norm alignment in first person action recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022.

[19] M. Planamente, G. Goletto, G. Trivigno, G. Averta, and B. Caputo, "Toward human-robot cooperation: Unsupervised domain adaptation for egocentric action recognition," in *Human-Friendly Robotics 2022: HFR: 15th International Workshop on Human-Friendly Robotics*, pp. 218–232, Springer, 2023.

[20] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, *et al.*, "Scaling egocentric vision: The epic-kitchens dataset," in *Proceedings of the Proc. Eur. Conf. Comput. Vis.*, 2018.

[21] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price, *et al.*, "Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100," *Int. J. Comput. Vis.*, 2021.

[22] X. Wang, L. Zhu, H. Wang, and Y. Yang, "Interactive prototype learning for egocentric action recognition," in *Proc. Int. Conf. Comput. Vis.*, 2021.

[23] E. Kazakos, J. Huh, A. Nagrani, A. Zisserman, and D. Damen, "With a little help from my temporal context: Multimodal egocentric action recognition," *arXiv preprint arXiv:2111.01024*, 2021.

[24] D.-A. Huang, L. Fei-Fei, and J. C. Niebles, "Connectionist temporal modeling for weakly supervised action labeling," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016.

[25] Z. Wang, Z. Gao, L. Wang, Z. Li, and G. Wu, "Boundary-aware cascade networks for temporal action segmentation," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020.

[26] Z. Li, Y. Abu Farha, and J. Gall, "Temporal action segmentation from timestamp supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021.

[27] H. Khan, S. Haresh, A. Ahmed, S. Siddiqui, A. Konin, M. Z. Zia, and Q.-H. Tran, "Timestamp-supervised action segmentation with graph convolutional networks," *arXiv preprint arXiv:2206.15031*, 2022.

[28] S. N. Aakur and S. Sarkar, "A perceptual prediction framework for self supervised event segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1197–1206, 2019.

[29] R. Mounir, R. Gula, J. Theuerkauf, and S. Sarkar, "Spatio-temporal event segmentation for wildlife extended videos," in *International Conference on Computer Vision and Image Processing*, pp. 48–59, Springer, 2022.

[30] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[31] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.

[32] A. Piergiovanni and M. Ryoo, "Temporal gaussian mixture layer for videos," in *Int. Conf. on Mach. Learn.*, 2019.

[33] Z. Du, X. Wang, G. Zhou, and Q. Wang, "Fast and unsupervised action boundary detection for action segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022.

[34] N. Crasto, P. Weinzaepfel, K. Alahari, and C. Schmid, "Mars: Motion-augmented rgb stream for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019.

[35] C. Plizzari, M. Planamente, G. Goletto, M. Cannici, E. Gusso, M. Matteucci, and B. Caputo, "E2 (go) motion: Motion augmented event stream for egocentric action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022.

[36] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks for action recognition in videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018.

[37] C. Feichtenhofer, "X3d: Expanding architectures for efficient video recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020.

[38] D. Kondratyuk, L. Yuan, Y. Li, L. Zhang, M. Tan, M. Brown, and B. Gong, "Movinets: Mobile video networks for efficient video recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021.

[39] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, 2009.

[40] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L2 hypersphere embedding for face verification," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017.

[41] M.-H. Song and R. I. Godøy, "How fast is your body motion? determining a sufficient frame rate for an optical motion tracking system using passive markers," *PloS one*, 2016.