Joint Encoding of Appearance and Motion Features with Self-supervision for First Person Action Recognition

Mirco Planamente
DAUIN,
Politecnico di Torino, Italy
Italian Institute of Technology
mirco.planamente@polito.it

Andrea Bottino
DAUIN,
Politecnico di Torino, Italy
andrea.bottino@polito.it

Barbara Caputo
DAUIN,
Politecnico di Torino, Italy
Italian Institute of Technology
barbara.caputo@polito.it

Abstract

Wearable cameras are becoming more and more popular in several applications, increasing the interest of the research community in developing approaches for recognizing actions from a first-person point of view. An open challenge is how to cope with the limited amount of motion information available about the action itself, as opposed to the more investigated third-person action recognition scenario. When focusing on manipulation tasks, videos tend to record only parts of the movement, making crucial the understanding of the objects being manipulated and of their context. Previous works addressed this issue with two-stream architectures, one dedicated to modeling the appearance of objects involved in the action, another dedicated to extracting motion features from optical flow. In this paper, we argue that features from these two information channels should be learned jointly to capture the spatio-temporal correlations between the two in a better way. To this end, we propose a single stream architecture able to do so, thanks to the addition of a self-supervised block that uses a pretext motion segmentation task to intertwine motion and appearance knowledge. Experiments on several publicly available databases show the power of our approach.

1. Introduction

Recognizing human actions from videos is one of the most critical challenges in computer vision since its infancy. The capability to automatically (and reliably) recognize the action performed by an individual or a group of people would have a tremendous impact on a plethora of applications, ranging from security and surveillance to autonomous driving, automatic indexing and retrieval of media content, human-robot, and human-computer interaction, and many others. Historically, most of the work has been done on third-person action, and activity recognition, an area where good progress has been made, and applications are already finding their way on the market. In the last years, the technological advances in the field of wearable devices led to a growing interest in first-person action recognition due to the possibility to capture activities following the user in mobility and without the need to place sensors in the environment.

When moving from third-person to first-person action recognition, one has to face several new challenges. A first issue is how to deal with strong egomotions, as data are usually acquired by wearable cameras mounted on the actor (very often on her/his head). A second and perhaps even more relevant challenge is the scarcity of available information about the pose of the main actor, as opposed to third-person videos. Most egocentric videos contain actions of the camera wearer interacting with objects [2], with only parts of the arm trajectory and the hand gestures visible in the captured data. Following this observation, it becomes crucial to extract from video frames as much information as possible on the objects being (or about to be) manipulated and their position.

Recent work has attempted to address the last issue with a combination of two pieces of information: the visual appearance of the object of interest, modeled by the spatial stream that processes RGB images, and the motion information, handled by the temporal stream that takes as input the optical flow extracted from adjacent frames (for a discussion on previous work we refer to section 2). Recent and successful state of the art approaches integrate the basic two-stream architecture with attention modules aimed at identifying the frames and the regions in the frames that are more informative for the task at end [29, 30].

Despite the good level of success obtained by these approaches, they also present two main disadvantages:

1. appearance and motion features are learned separately, and the final predictions of the two streams are merged only at the end of the network using (usually) simple weighted sums [27, 31, 29]. However, this choice is

sub-optimal since it does not model their correlated spatial-temporal relationships;

2. the most recent and successful methods push the envelope in the two-stream approach at the expense of a growth in the size of the overall architecture and hence the number of parameters. As a result, optimization is often performed in multiple stages.

In this paper, we address these issues. We move beyond the two-stream paradigm and propose an architecture that couples the modeling of motion and appearance information through a motion segmentation (MS) self-supervised task. This MS auxiliary task "forces" the backbone to learning an image embedding that focuses on object movements, a piece of information that is beneficial for the main task of action recognition. Thanks to the use of a self-supervised auxiliary task, this information is directly encoded in the inner layers of the backbone, hence leading to an intertwined learning of appearance and motion features. The effectiveness of this idea is demonstrated not only by our results (Section 4.5), but also by those obtained including this MS pretext task in other recent models (e.g., Ego-RNN [31] and LSTA-RGB [29], see section 4.5). The resulting architecture is relatively simple, as it consists of a standard ResNet-34 as the backbone, followed by a standard ConvLSTM, and the MS head includes a single convolutional block. Because of its simplicity, it can be trained end-to-end in a single stage, as opposed to several other two streams methods [31, 29]. Furthermore, it can use a smaller amount of frames than what done in previous works [31, 29], without any adverse effect on the performances. We call our architecture Self-supervised first Person Action Recognition network - SparNet.

To the best of our knowledge, SparNet is the first architecture for first-person action recognition employing a self-supervised task to learn about appearance and motion features jointly, as well as the first architecture able to achieve the state of the art on different publicly available databases without using a two streams approach. A thorough ablation study illustrating the inner workings of SparNet completes our experimental evaluation.

In the rest of the paper, we first revise previous work in action recognition, self-supervised learning, and we discuss into detail how we position ourselves with previous approaches that relate to some extent with SparNet (Section 2). Section 3 describes our proposed architecture, while experiments are reported in Section 4 and discussion on future work are reported in the conclusion.

2. Related works

First Person Action Recognition. The literature on first-person action recognition has long acknowledged that the motion of the hands, the appearance of the objects being used, and the interplay of these two components are the most critical characteristics to extract from raw data [5, 24, 6]. This approach has moved from handcrafted features-based works to the deep learning wave.

Indeed, deep networks have been successfully applied to third-person [8, 17, 28] as well as first-person action recognition [15, 14, 31], providing researchers with powerful and effective models for encoding appearance. However, when it comes to action recognition, these methods rely on simple aggregation of frame-wise decisions, and they completely neglect the temporal relationships and the dynamics between frames. Several approaches exploit Convolutional Long Short-Term Memory (ConvLSTM) networks to tackle this issue [29, 31]. ConvLSTMs attempt to model the temporal dependencies between frames by taking into account within-frame spatial correlations too. However, it is still not clear if the temporal features extracted by these networks are as effective as those obtained leveraging explicit optical flow data in capturing complex and unsteady motion dynamics.

To this end, many recent deep learning methods for action recognition follow the two-stream approach [27, 28, 17, 31]. This method addresses the two tasks of recognition from motion and recognition from appearance with different networks that are either combined with late fusion, i.e., before the classification step or fused at the decision level. Furthermore, while the appearance stream analyzes individual RGB frames, the motion stream usually processes a short sequence of stacked adjacent flow frames that, typically, have no one-to-one relation with the RGB input. This lack of correlation between the spatial and temporal information processed limits the network capabilities of extracting complex motion dynamics in long video segments.

Recent works attempted to further study the temporal aspects of videos using attention mechanisms [36, 31, 29] to find the most informative parts in images (spatial attention) or through videos (temporal attention). These approaches are generally cast within a two streams network framework. Although they have shown a reasonable degree of success, the resulting architectures tend to be heavy parameter-wise and often need to be trained in two stages or more.

Learning to Recognize with Self-Supervised Tasks. Self-Supervised Learning (SSL, [11]) has been recently introduced for learning visual features from unlabeled data. By choosing an auxiliary task that does not require human annotation of the data, it is possible to encode into the first layers of a network knowledge that proves beneficial as initialization when solving a classification problem on related data. The transfer of the model learned by solving the self-supervised task to a downstream classification net has proved to be useful in several contexts. Different authors have proposed several auxiliary tasks roughly organized into three main groups. The first relies only on original visual cues and involves either the whole image with geometric transformations (*e.g.* translation, scaling, rotation [7, 3]), clustering [1], inpainting [23] and colorization [37], or considers image patches focusing on their equivariance (learning to count [19]) and relative position (solving jigsaw puzzles [18, 20]). The second group uses either real or synthetic external sensory information. This approach is popular for multi-cue problems (visual-to-audio [21], RGB-to-depth [25]) and in robotics [10, 13]. Finally, the third group relies on video and the regularities introduced by the temporal dimension [35, 26]. The use of motion cues in self-supervised learning has been first proposed in [22], where authors trained a convolutional network to segment a static frame (in an unsupervised way) using motion data obtained from videos. Optical flow cues had been exploited in [9], to learn the visual appearance of obstacles in a Micro Air Vehicle landing environment. Also [12] used a self-supervised motion segmentation task applied to monoscopic visual odometry and ground areas labeling. Instead, [16] "transfer" optical flow information to pixel embeddings so that their difference matches the difference between optical flow vectors of the same pixels.

To the best of our knowledge, the specific self-supervised auxiliary task (motion segmentation) has been used here for the first time in the area of action recognition.

3. Architecture overview

The design of SparNet stems from the general observation that videos convey two complementary pieces of information related to spatial (appearance) and temporal (motion) clues, which often need to be considered jointly for action recognition. For instance, the correct interpretation of the actions of "opening" and "closing" a can depends merely on the hand motion direction. As opposed to previous work that followed the standard two streams approach, here we attempt to solve the problem of using a single stream to jointly exploit spatial and temporal clues in the egocentric action recognition process. The advantage we expect is a leaner algorithm that can be trained in a single stage, thus being faster than two-stream models, and that provides effective accuracy rates thanks to the possibility of exploiting both appearance and motion features.

In the basic version of the proposed architecture (Figure 1, action recognition block), we first extract a small number N of representative RGB frames for each input video segment. These images are then processed by a standard CNN backbone, and the resulting appearance embeddings are fed to a ConvLSTM network. Finally, the ConvLSTM output is first sent to an average pooling layer and then to a fully connected layer for classification.

While processing a small number of frames help reduce the computational burden of the model, the resulting appearance embeddings still lacks the motion information that is vital for the recognition process, a piece of information that the two-stream approaches exploit by leveraging explicit optical flow data.

To tackle this issue, we propose to extend the basic architecture into a multi-task network that is required to solve jointly two different problems. The first is the action recognition task. The second is a motion segmentation task (MS), which can be formalized as a self-supervised labeling problem aimed at minimizing the discrepancies between a binary map labeling pixels as either *moving* or *static* (which can be obtained from the input video segment in an unsupervised way) and the object movements predicted by the network when observing a single static RGB frame.

The pretext MS task serves two purposes. First, it acts as a data-dependent regularizer for action representation learning. Second, and most important, it aims at helping the appearance stream learn an embedding that encodes motion clues as well, thus making the learned features more rich and expressive for the main action recognition task. In other words, we argue that, by processing an input with these characteristics, the ConvLSTM can extract a more meaningful global video representation (in terms of appearance and both short and long-term motion dependencies among frames) than the one observable with the vanilla appearance embeddings. Thus, at test time, we can use only the action recognition block to predict the label sample and feed the model with (sparse) RGB frames only.

Going into details, let $\mathcal{S}$ be a training set consisting of samples $S_i = \{H_i, y_i\}_{i=1}^n$, where H_i is a set of N timestamped images $\{(h_i^k, t_i^k)\}_{k=1}^N$ uniformly sampled from the video segment. Let also $x = f_M(H|\theta_f, \theta_c)$ be the embedding of sample S computed by our model M, where parameters θ_f and θ_c define, respectively, the image embedding and the classification spaces. Finally, let $g(x)$ be a class probability estimator on the embedding x.

The action recognition and the MS task share a common trunk that is completed by two task-specific heads (the ConvLSTM network for action recognition and a shallow convolutional network for MS). The first objective of the learning step consists of estimating the model parameters that minimize a loss $\mathcal{L}_c$ for the action recognition head. This loss function is based on the cross-entropy between the predicted and the true labels:

$$\mathcal{L}_c(x, y) = -\sum_{i=1}^{n} y_i \cdot log(g(x_i))) \qquad (1)$$

Together with the aforementioned objective, we ask the network to solve the MS task by feeding the output of the backbone to a shallow head composed by a single convolutional block, aimed at both adapting the features to the MS task and reducing their channel number. This head ends with a fully connected layer of size s^2 followed by a softmax, and

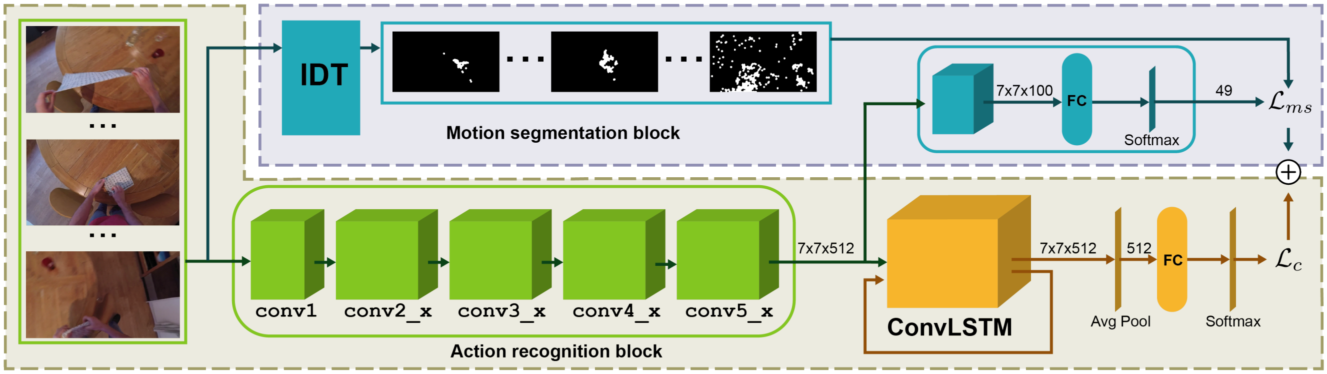

Figure 1. Overview of the SparNet architecture

it is trained with a loss $\mathcal{L}_{ms}$ based on the per-pixel cross entropy between the computed label image l and the ground truth (which is first downsampled to a size $s \times s$ and then vectorized). The estimated motion map l is obtained as a function of both image embedding x, which depends on (θ_f), and MS head parameters (θ_{ms}). Thus, the $\mathcal{L}_{ms}$ loss can be defined as:

$$\mathcal{L}_{ms}(x, m) = -\sum_{i=1}^{n}\sum_{k=1}^{N}\sum_{j=1}^{s^2} m_i^k(j) \cdot log(l_i^k(j)) \qquad (2)$$

where m is the ground truth.

Concluding, the optimal SparNet model is obtained by solving the following optimization problem:

$$\arg\min_{\theta_M} \mathcal{L}(x, y, m|\theta_M) = \\ \mathcal{L}_c(x, y|\theta_f, \theta_c) + \mathcal{L}_{ms}(x, m|\theta_f, \theta_{ms}) \qquad (3)$$

where $\theta_M = \{\theta_f, \theta_{ms}, \theta_c\}$. It should be noted that the self-supervised loss has the same relevance of the classification loss during the network training. As a consequence, their combination does not require the fine-tuning of an extra hyper parameter.

As for the implementation, while the design of SparNet network can leverage over many possible convolutional deep architectures, we choose for our experiments a ResNet-34 model pre-trained on ImageNet, which is both a lightweight and powerful backbone. The MS head receives in input the features extracted from the `conv5_x` block of the ResNet (whose size is $7 \times 7 \times 512$) and reduces their channels to 100. The size s^2 of the resulting motion maps is, therefore, 49. The ground-truth motion maps are computed with a method similar to the one described in [22], where the main difference is that we extract "stabilized" motion information exploiting the Improved Dense Trajectories (IDT) proposed in [32]. The main idea of IDTs is first to compensate for the effect of camera motion (by estimating the homography that relates adjacent frames) and then to label as *moving* the keypoints that can be tracked reliably for 10 frames and are not identified as camera motion.

4. Experiments

In this Section, we first introduce the datasets used in our experiments, along with some implementation details and the description of the training parameters used in the various experiments. Then, we discuss the results, which show the strength of SparNet in the analyzed benchmarks. Finally, we conduct an ablation analysis to show the effectiveness of the proposed self-supervised MS task and our single-stream approach, along with a comparison with recently proposed attention modules.

4.1. Datasets

We evaluated the proposed approach on three standard first-person action recognition datasets, namely GTEA-61, EGTEA+, and EPIK-KITCHEN.

GTEA-61 [4] is an egocentric dataset that includes videos depicting 7 daily activities performed by 4 different subjects. It includes high definition images (1280 × 720) captured with a head-mounted camera. Extended GTEA Gaze+ (EGTEA+, [14]) subsumes GTEA-61 and contains 29 hours of egocentric videos from 86 different sessions, which depicts 7 different meal preparations performed by 32 volunteers and divided in about 10,000 video segment. These segments are annotated with 106 fine-grained actions, which are characterized by a long-tail distribution that poses challenges in the recognition due to the large unbalance of available samples for the different classes (from few hundreds to about 30).

Finally, EPIC-KITCHENS is the largest of all these datasets [2]. It contains about 40,000 video segments in full HD (for an overall length of about 55 hours of recordings) depicting hundreds of daily actions performed by 32 volunteers in their kitchen. Each segment is labeled in terms of *verb* and *noun*, which are then combined to get the segment *action* label. Other challenges in the classification are due

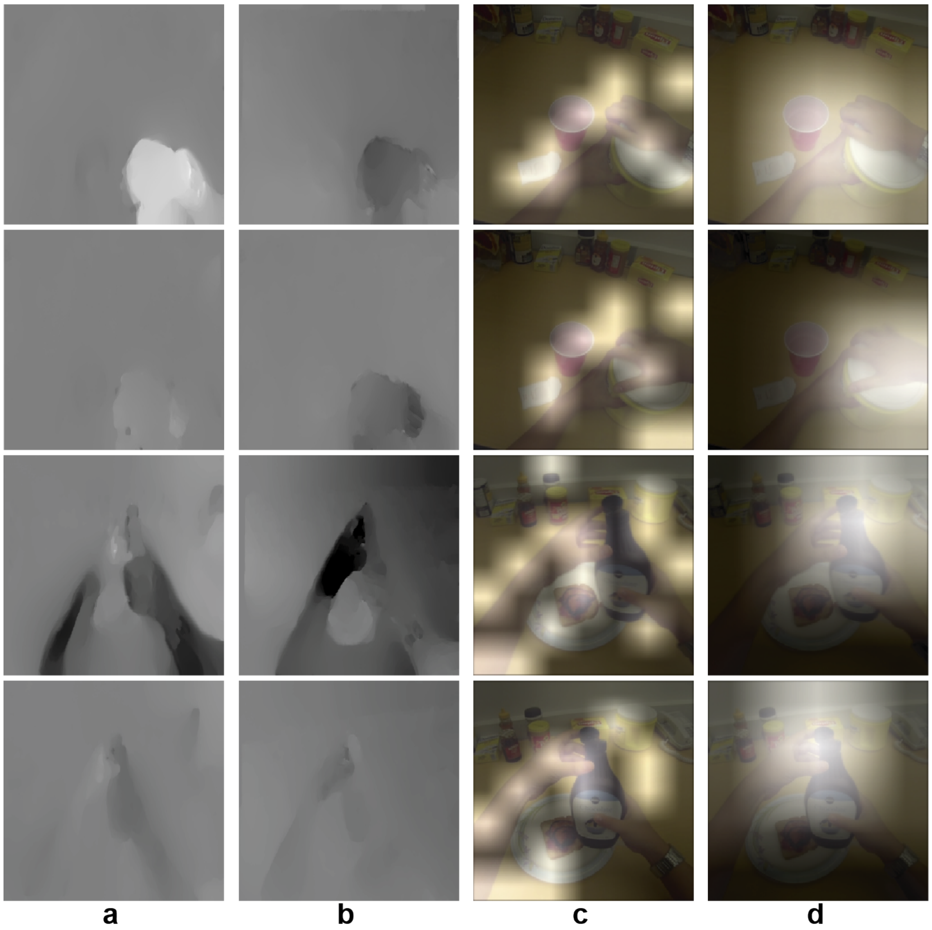

Figure 2. Visualization of where SparNet focus its attention. Each row shows two frames taken from a GTEA-61 video segment; **c**: **motion maps** predicted by the MS task, which present large similarities with the computed warp flow (i.e., the camera motion subtracted from the optical flow, represented as the X and Y displacements shown in **a**-**b**); **d**: "attention" (CAM) of the ResNet-34 backbone of SparNet, which shows where the motion-augmented appearance stream focuses its attention.

to the fact that not all the test classes have a significant number of training samples, and not all possible combinations of *verb* and *noun* correspond to an existing *action*.

4.2. Implementation details

SparNet is trained end-to-end on a single stage to minimize the loss defined by Eq. 3. The ResNet-34 is initialized with the weights trained on ImageNet. The ConvLSTM cell has 512 hidden units for temporal encoding and is initialized with the same approach described in [31]. During training, we use different learning rates for the various architectural blocks (backbone, MS head, ConvLSTM, and final classification layer). We set the number of training epochs to 500 for GTEA-61 and 100 for EGTEA, and we use ADAM as the optimization algorithm. Due to memory issues, we had to keep small the batch size for all datasets (namely, 4 for GTEA-61 and 8 for EGTEA+).

Each input video segment is decomposed into $N = 7$ frames, uniformly sampled in time. We follow the standard approach for the pre-processing procedure. It consists of resizing the image at the height of 256 pixels, maintaining the same height ratio to update the width, and then extracting from the image the actual training input as a random crop of size 224 × 224 pixels. To reduce overfitting issues and compensate for the possibly limited number of training samples, we use the data augmentation techniques proposed in [33], which exploit corner cropping, scale jittering, and random horizontal flipping approaches. At test time, we fed the network with the central crop of the frames. Concerning the ground truth for the MS task, we rescaled all videos before being processed by IDT at a fixed height of 540 pixels (where, again, the width is automatically resized with the same height ratio).

Concerning the results of SparNet, since the non-determinism caused by the inherent randomness in data preparation, data augmentation, and weight initialization may impact both the quality and the stability of the results, we tried to achieve as much as possible their reproducibility.

To this end, we made three runs for each experiment, defining an individual constant seed for each run across different datasets and parameters. Therefore, unless stated otherwise, we present in all tables SparNet results as the average accuracy over these three runs.

4.3. Experiments on GTEA-61 and EGTEA+

The experiments with GTEA-61 and EGTEA+ followed the protocols defined in [31, 29]. In detail, GTEA-61 defines four splits (where all video segments of one subject are included in the test set and those of the three remaining subjects in the training set) and two experiments. The first is based on a single fixed split (i.e., the one where the subject S2 goes into the training set), and the second requires to test all the splits and report the final average accuracy. The experimental protocol of EGTEA+ defines three pairs of non-overlapping training and test sets, and the results are defined in terms of average accuracy over the three splits.

Results are shown in Table 1, where we compare our approach against several state-of-the-art methods. Ego-RNN [31] and LSTA [29] are two models that have some similarities to SparNet. Ego-RNN leverage a ResNet-34 to compute a per-frame Class Activation Map (CAM) [38]. This CAM is then used as a spatial attention clue to modulate the features extracted from the last convolutional layer of the backbone (i.e., the output of block `conv5_x`), which are finally fed to a ConvLSTM. LSTA is an extension of Ego-RNN that directly integrates the attention layer into the LSTM cell. While both models are based on a two-stream architecture, authors presented as well their single-stream versions, which are referred in the Table as Ego-RNN RGB and LSTA-RGB. The Temporal Segment Network (TSN) [34] is a method that has been proposed for third-person recognition and aims at capturing long-range temporal structures by employing multi-layered processing of sparsely sampled video snippets. Finally, given the demonstrated relevance of attention modules in improving the recognition accuracy in first-person videos, we took into consideration as well EleAttG [36], which is a method that adds an attention gate to a Recurrent Neural Network. Thus, this model can also be seen (to some extent) as an attention-enhanced version of our action recognition block.

Method	GTEA-61(*)	GTEA-61	EGTEA+
TSN [34]	67.76	69.33	55.93
EleAttG [36]	59.48	66.77	57.01
Ego-RNN RGB [31]	63.79	—	—
LSTA RGB [29]	74.14	71.32	57.94
Ego-RNN [31]	77.59	79.00	60.76
LSTA [29]	79.31	80.01	61.86
SparNet	**80.18**	**80.30**	**63.51**

Table 1. Comparison with the state of the art on GTEA-61 (where (*) indicates the fixed split protocol) and EGETA+ datasets. The best results are highlighted in bold.

From Table 1, it can be seen that SparNet reaches the state of the art in all the benchmarks and experimental protocols. In particular, these results suggest the effectiveness of the MS task in empowering our single-stream approach. As a matter of fact, if we compare SparNet against the other single stream methods, we can appreciate the sensible improvements of the performances with respect to EleAttG, Ego-RNN RGB, and LSTA-RGB. This gap is reduced (but still present) for the two-stream approaches (TSN, Ego-RNN, and LSTA). We think this is a clear indication of the fact that the motion clues induced in the appearance stream by the self-supervised auxiliary task were indeed capable of improving the discriminative capabilities of the final embeddings, to an extent higher than that provided by using explicit optical flow information and without the need to include specific attention modules in the architecture.

4.4. Experiments on EPIC-KITCHENS

The experimental protocol for the action recognition benchmark aims at classifying each test segment into its action class. In order to assess the generalization properties of the method under analysis, the test set is divided into two splits, *seen* kitchens (S1, where each kitchen is present in both test and train sets) and *unseen* kitchens (S2, where the video segments shot in the same kitchen are all either in the train or in the test set).

Results are provided in terms of both aggregate metrics (as the top-1 and top-5 accuracy for the correct detection of *verb*, *noun* and *action* labels) and per-class metrics (in terms of precision and recall). The EPIC-KITCHENS recognition baseline includes different variants of Temporal Segment Network (TSN) [34], namely RGB-TSN, Flow-TSN, and two-stream TSN, none of which outperforms the others on all splits and metrics.

The action recognition task was solved by transforming SparNet in a multi-task network trained in parallel on both *verb* and *noun* recognition tasks using as loss the average cross-entropy of both heads.

$$\mathcal{L}_c = \mathcal{L}_{verb} + \mathcal{L}_{noun} \tag{4}$$

As for the *action* classification, the output label is obtained from the $p(verb)$ and $p(noun)$ softmax probabilities as $p(action\{verb, noun\}) = p(verb) \cdot p(noun)$.

At the time of submission, SparNet is participating in the ongoing Epic Kitchen Action Recognition Challenge (closing date November 22, 2019). The results of the various runs are reported in the supplementary material.

4.5. Ablation Study

In this section, we comprehensively evaluate SparNet on the fixed split of the GTEA-61 dataset. The baseline for our model is a plain ResNet-34 followed by a ConvLSTM, i.e., the action recognition block in Figure 1.

Impact of the self-supervised auxiliary task. The results of this analysis are summarized in Table 2, where we show, for different variants of the baseline and of SparNet, the average accuracy obtained on three runs of the learning process, the average test time and the average training time per epoch (both expressed in seconds). Ablation results clearly highlight the contribution of the MS task. The accuracy of the baseline is 73.85%, and the accuracy gain with SparNet is 7.03%. It is also possible to note that the introduction of an auxiliary task does not significantly burden the computational load of the model since the average training time per period increases by a minimum percentage (7.79%, with a difference in absolute terms of 3.24 seconds per epoch).

Figure 2 provides some hints about the capabilities of this pretext task to indeed instill motion information into the appearance stream. The figure shows that, despite the noise, the predicted motion maps computed from a single static frame (column c) have significant similarities with the warp flow (i.e., the flow obtained by subtracting from the optical flow the camera motion [34]) computed for the same frame (columns a and b). As a result, the backbone focuses most of its attention on moving objects (column d).

Since the effectiveness of the MS task (in terms of motion segmentation capabilities) depends primarily on the features it receives in input, we analyzed different options. Working with a ResNet architecture, a natural option is to extract features at the end of the principal residual blocks of the backbone (namely, the conv3_x, conv4_x and conv5_x blocks displayed in Figure 1). When changing the input, the only update required to the MS head relates to the size of its layers and to the scale factor applied to the ground truth to obtain a motion map with a dimension compatible with the final output of the head. It can be seen from Table 2 that the last block (conv5_x) is the one that provides the highest contribution. The features of this block show a relatively small gain with respect to the output of conv4_x (0.58%) but a larger gap with the lowest block conv3_x (3.46%). We think that is is a clear indication that the motion segmentation task benefits from leveraging high-level and more structured information for its analysis.

Method	Accuracy (%)	$T_{train}/epoch$
baseline (7 frames)	73.85	41.55
baseline (9 frames)	74.42	47.28
baseline (25 frames)	73.56	165.60
SparNet (7 frames)	**80.18**	44.79
SparNet (9 frames)	78.17	57.00
SparNet (25 frames)	78.16	169.54
SparNet @ conv3_x	76.72	54.77
SparNet @ conv4_x	79.60	48.71
SparNet @ conv5_x	**80.18**	47.8

Table 2. Ablation results on GTEA-61. The best result is highlighted in bold.

Another method parameter that can be analyzed is the number of input frames used for action recognition. Varying this number in the interval $[6, 25]$, we did not observe significant differences for smaller values (between 6 and 8) while the error started (slightly) increasing for higher values. For the sake of brevity, in Table 2, we report the optimal value (7) and the largest value tested (25). It can be seen that using 25 frames, the accuracy of SparNet drops by 2.02%. On the contrary, the baseline difference is negligible (0.25%). We conjecture that this phenomenon is due to the fact that a sparser frame sampling is more favorable in our case since, as also observed in [34], a dense temporal sampling results in highly redundant information that is unnecessary for capturing the temporal dynamic of the video. Another possible cause is that, due to the reduced number of training samples available in GTEA-61, a larger number of frames increase the overfitting issues that affect our method.

MS task vs. attention modules. One possible question is if the proposed MS auxiliary task can be beneficial to other models too. To this end, we performed a detailed comparison with the effect of MS on Ego-RNN RGB [31] and LSTA-RGB [29] (i.e., the single-stream versions of Ego-RNN and LSTA). We should also mention that both methods converge to the same baseline of SparNet when the CAM is deactivated (in Ego-RNN RGB) or a vanilla LSTM cell is used instead of the proposed LSTA cell (in LSTA-RGB). For these experiments, we modified both Ego-RNN RGB and LSTA-RGB architectures following the recipe used in SparNet (i.e., we fed the MS head with the features extracted from the conv5_x block adding a motion segmentation loss over all the input frames). Both methods were run using 25 frames in input, as in their original papers.

A necessary comment before introducing the results is that for the experiments with LSTA-RGB we used the code made publicly available by its authors. However, despite our efforts, we were not able to replicate the results presented in [29]. Nonetheless, even if the numbers we obtained were significantly lower, we still think they allow us to obtain valuable indications about the effect of MS on this model. With this clarification, results in Table 3 show that the MS task induces an improvement of 3.16% for Ego-RNN RGB and 2.86% for LSTA-RGB, confirming that the effectiveness of MS task is not limited to SparNet.

We can also note that for these two methods, the improvement over the baseline offered by the MS task is lower than the one provided in SparNet. This fact can be explained in terms of the implementation of the learning process in these methods. Both Ego-RNN RGB and LSTA-RGB retrain merely the last residual block of the backbone and not the whole ResNet. Thus, the effect of the MS task is limited to influencing the high-level features only, and it is not back-propagated to the ones at lower levels, which, in turns, prevent them from supporting the higher levels in learning new features that are more focused on the actual task of first-person action recognition.

Method	Accuracy (%)
Ego-RNN RGB [31]	63.79
Ego-RNN RGB + MS	66.95
LSTA-RGB [29]	74.14
LSTA-RGB (*)	57.19
LSTA-RGB (*) + MS	60.05
baseline (7 fr.) + CAM	65.22
SparNet (7 fr.) + CAM	72.98
baseline (25 fr.) + CAM	66.95
SparNet (25 fr.) + CAM	73.56

Table 3. Analyzing the contribution of MS task on Ego-RNN RGB and LSTA-RGB and of the CAM module on SparNet. The LSTA-RGB results indicated with (*) are these obtained in our experiments, the remaining one is taken from [29].

A related question is if SparNet can benefit from the introduction of an attention module, like the one used in both Ego-RNN RGB and LSTA-RGB. To this end, we modified the SparNet structure to exploit the CAM, in a way similar to that described in [31]. We first added the CAM module to the baseline and then to SparNet, in order to analyze the differences between the two approaches and the possible synergies between the CAM and our auxiliary task. It can be seen that the introduction of the CAM causes substantial accuracy drops in both the baseline and SparNet, even when different numbers of input frames are considered (8.63% and 6.61 for the baseline at, respectively, 7 and 25 frames, 7.2% and 4.6% for SparNet).

A possible explanation of this behavior is that the CAM in [31, 29] has the function of modulating the features extracted from the appearance stream. Since, in their case, the backbone is not fully retrained during learning, these features are not fully fine-tuned to the task at hand. On the contrary, our features are more "task-specific", and the simple use of the CAM could lead to emphasize specific features in spite of others, which indeed could still be relevant for the main action classification task.

5. Conclusions

In this paper, we presented SparNet, a single stream architecture for first-person action recognition. Its main feature is that it is able to jointly learn appearance and motion features thanks to the use of a self-supervised pretext task aimed at estimating, from a single static image, a motion-based segmentation of the input frame. This leads to a lighter architecture (with respect to the two-stream models, which is the mainstream approach for action recognition), which is trainable in a single stage, can work on a sparse sampling of the input video segment and achieves the current state of the art on several publicly available datasets.

Despite the promising results obtained, there is still room for further improvements. As future works, we are planning to investigate the contribution of other self-supervised pretext tasks and the possible integration of multiple auxiliary tasks, capable of further strengthening the discriminative capabilities of the final embeddings. Examples of such tasks are the jigsaw problem (which could help the model infer useful information about the spatial relationships between objects in the image), the identification of the correct temporal order of a sequence of images (to learn better temporal correlations between frames) or colorization, where an image is split into its intensity and color components, the former predicting the latter (which could help segmenting object of interest for the action recognition tasks, like the hands). Then, since SparNet is backbone agnostic, we will also investigate the contribution of different state-of-the-art models for the appearance stream. Finally, another option we are interested in is verifying the possibility of using modality hallucination approaches as an alternative to the MS task for instilling a "flavour" of motion into the (single stream) appearance features.

References

[1] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[2] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The dataset. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 753–771, Cham, 2018. Springer International Publishing. 1, 4

[3] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 3

[4] Alireza Fathi, Xiaofeng Ren, and James M. Rehg. Learning to recognize objects in egocentric activities. In *CVPR 2011*. IEEE, June 2011. 4

[5] R De Geest, E Gavves, A Ghodrati, Z Li, C Snoek, and T Tuytelaars. Online action detection. In *Proc ECCV*, 2016. 2

[6] Ross Ghirshick. Fast r-cnn. In *proc ICCV*, 2015. 2

[7] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 3

[8] K He, X Zhang, S Ren, and J Sun. Deep residual learning for for image recognition. In *Proc CVPR*, 2016. 2

[9] H. W. Ho, C. De Wagter, B. D. W. Remes, and G. C. H. E. de Croon. Optical-flow based self-supervised learning of obstacle appearance applied to MAV landing. *CoRR*, abs/1509.01423, 2015. 3

[10] Eric Jang, Coline Devin, Vincent Vanhoucke, and Sergey Levine. Grasp2vec: Learning object representations from

self-supervised grasping. In *Conference on Robot Learning*, 2018. 3

[11] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *arXiv preprint:1902.06162*, 2019. 2

[12] B. Lee, K. Daniilidis, and D. D. Lee. Online self-supervised monocular visual odometry for ground vehicles. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5232–5238, May 2015. 3

[13] Michelle A. Lee, Yuke Zhu*, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *International Conference on Robotic Automation (ICRA)*, 2019. 3

[14] Yin Li, Miao Liu, and James M. Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2, 4

[15] S Ma, L Sigal, and S Sclaroff. Learning activity progression in lstm for activity detection and early detection. In *Proc CVPR*, 2016. 2

[16] Aravindh Mahendran, James Thewlis, and Andrea Vedaldi. Cross pixel optical-flow similarity for self-supervised learning. In C.V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler, editors, *Computer Vision – ACCV 2018*, pages 99–116, Cham, 2019. Springer International Publishing. 3

[17] TT Mahmud, M Hasan, and A K Roy-Chowdhury. Joint prediction of activity labels and starting times in untrimmed videos. In *Proc ICCV*, 2017. 2

[18] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision (ECCV)*, 2016. 3

[19] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *International Conference on Computer Vision (ICCV)*, 2017. 3

[20] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[21] Andrew Owens and Alexei A. Efros. Audio-visual scene analysis with self-supervised multisensory features. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[22] Deepak Pathak, Ross B. Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6024–6033, 2017. 3, 4

[23] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[24] H Pirsiavash and D Ramanan. Detecting activities of daily living in first-person camera views. In *Proc CVPR*, 2012. 2

[25] Zhongzheng Ren and Yong Jae Lee. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[26] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. In *International Conference on Robotic Automation (ICRA)*, 2018. 3

[27] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pages 568–576, Cambridge, MA, USA, 2014. MIT Press. 1, 2

[28] E H Spriggs, D De La torre, and M Hebert. Temporal segmentation and activity classification from first-person sensing. In *Proc CVPR Workshop*, 2017. 2

[29] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. LSTA: Long Short-Term Attention for Egocentric Action Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 6, 7, 8

[30] Swathikiran Sudhakaran and Oswald Lanz. Attention is all we need: Nailing down object-centric attention for egocentric activity recognition. In *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, page 229, 2018. 1

[31] Swathikiran Sudhakaran and Oswald Lanz. Attention is all we need: Nailing down object-centric attention for egocentric activity recognition. In *British Machine Vision Conference*, 2018. 1, 2, 5, 6, 7, 8

[32] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, Sydney, Australia, 2013. 4

[33] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(11):2740–2755, 2019. 5

[34] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 20–36, Cham, 2016. Springer International Publishing. 6, 7

[35] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *International Conference on Computer Vision (ICCV)*, 2015. 3

[36] Pengfei Zhang, Jianru Xue, Cuiling Lan, Wenjun Zeng, Zhanning Gao, and Nanning Zheng. Adding attentiveness to the neurons in recurrent neural networks. *CoRR*, abs/1807.04445, 2018. 2, 6

[37] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*, 2016. 3

[38] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, June 2016. 6