arXiv:1907.09382v1 [cs.CV] 22 Jul 2019

Domain-Specific Priors and Meta Learning for Low-shot First-Person Action Recognition

Huseyin Coskun[1,2] M. Zeeshan Zia[2] Bugra Tekin [2] Federica Bogo [2]
Nassir Navab [1] Federico Tombari [1] Harpreet Sawhney[2]
[1] Technische Universität München [2] Microsoft

Abstract

The lack of large-scale real datasets with annotations makes transfer learning a necessity for video activity understanding. Within this scope, we aim at developing an effective method for low-shot transfer learning for first-person action classification. We leverage independently trained local visual cues to learn representations that can be transferred from a source domain providing primitive action labels to a target domain with only a handful of examples. Such visual cues include object-object interactions, hand grasps and motion within regions that are a function of hand locations. We suggest a framework based on meta-learning to appropriately extract the distinctive and domain invariant components of the deployed visual cues, so to be able to transfer action classification models across public datasets captured with different scene configurations. We thoroughly evaluate our methodology and report promising results over state-of-the-art action classification approaches for both inter-class and inter-dataset transfer.

1. Introduction

Automatically recognizing human hand actions at close range is an important problem for applications such as assembly line inspection, augmented reality training and operations, and home scenarios. Creating annotated training data for reliable learning of fine scale actions and activities in videos with deep convolutional neural networks is a daunting challenge that limits the scalability, diversity and deployability of research in real world systems. Current approaches either address clip level classification and detection or demand detailed annotations of objects, hands and actions for fine scale recognition.

Within this scope, we note that the science of deep video understanding trails behind deep image understanding: for instance, transfer learning that has been established at least since 2013 [21] for images was only validated comprehensively in 2017 for videos [6] due to the lack of large-scale action recognition datasets. Similarly, insights into internal CNN representations that were already available for image-trained CNNs in 2014 [69] were only comparably studied in 2018 [16] for video CNNs. Likewise, relatively little work exists in learning transferable representations for first-person action recognition, and tends to employ relatively complex approaches such as language [42, 66], sequence alignment [67], or probabilistic reasoning [49].

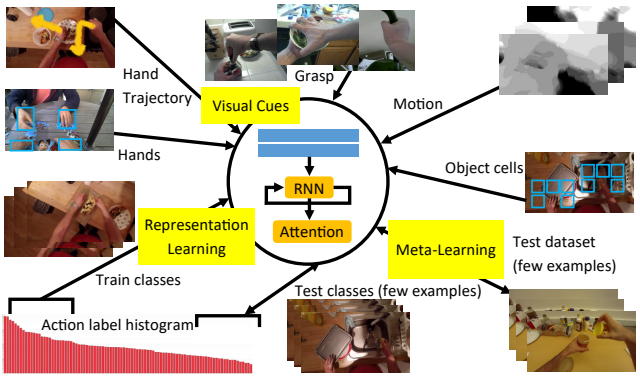

Figure 1. Leveraging domain-specific visual cues in first-person-views that decouple foreground action from background appearance and meta-learning, to enable low-shot transfer of action representations.

We investigate recognizing primitive actions in first-person videos when only a few training examples are available. We focus on first-person action parsing since it is particularly challenging in its detail and variability of objects and context involved in fine scale actions. First-person action parsing is inherently compositional in that it involves interaction between hand poses, manipulated objects and motion. Learning truly generalizable deep models for the task requires combinatorially large first-person video datasets. Potentially every object that maybe held in the hands needs to be captured in-situ while covering the entire gamut of hand shapes and appearances, as well as poses and motions that are natural to that specific object and action.

To address this fundamental scalability limitation, we study three related issues in this paper:

1. **Visual cue learning from independent datasets**: We first propose cues relevant to the task of first-person action understanding as strong priors to regularize learning and inference. It is unrealistic to assume availability of labels for every visual cue relevant to a given video understanding task in a single training set. We consider a set of a-priori visual features for action recognition, and explore *using disparate datasets* to learn each individual cue. We then test on first-person action datasets in which limited level of annotations are available. For instance, numerous datasets are available for object detection: we investigate the efficacy of transferring this training to action datasets while using object level context and detection.

2. **Inter-class transfer**: We observe that most existing first-person action datasets are severely imbalanced [9, 33] with few actions with large number of examples while most actions fall in the long tail of the action distribution. We investigate training action classification models on one set of classes with relatively large number of training examples and transferring to (testing on) a disjoint set of classes given only a few training examples for the latter. Success in this endeavor would enable incrementally adding actions "in-the-field" to the set of recognizable actions without demanding large number of annotations.

3. **Inter-dataset transfer**: We also explore transfer learning as per its most common interpretation, *i.e.* by transferring models trained on one dataset to another dataset. Achieving this capability would enable models that can generalize across significant scene and object appearance, *e.g.* permit fine-tuning a model originally trained for "kitchen activity" on factory floor tasks.

To tackle these few-shot learning problems, we explore the use of meta learning for action recognition. Recent work [18, 19, 47] successfully applied meta learning to image classification tasks; however, its use for video classification has received less attention [18]. We build on the Model-Agnostic Meta-Learning (MAML) algorithm [18], combining it with an attention mechanism to improve its performance on temporal sequences. We call this approach Attentive MAML (A-MAML).

We validate a complete pipeline (Section 3) for transferring first-person action representations across datasets and classes. Our ablation studies yield further insights into the nuances of performing such transfer (Section 4). Figure 1 depicts our use of multiple contextual cues trained on their respective datasets and used in our investigation to demonstrate multiple transfer learning tasks.

Our contributions are summarized as follows:

1. We introduce strong priors for object and action context that decouple foreground action from background appearance in first-person videos including hand regions as hard focus-of-attention cues, grasp-classification, class-agnostic object-object interactions, and hand trajectory. Crucially, we demonstrate successful transfer of these cue extractors from diverse image-only datasets.

2. We highlight the effectiveness of these powerful domain cues in the context of transfer learning, for both inter-class and inter-dataset transfer. We perform a thorough evaluation across two large-scale first-person action recognition datasets, outperforming both powerful ablative baselines as well as state-of-the-art approaches for action classification.

3. We explore the use of meta learning in action recognition scenarios and propose Attentive MAML, an algorithm combining MAML [18] with an attention mechanism for simplified and more effective training.

2. Related Work

We review below the rich literature on action recognition, transfer learning and few-shot learning, with a particular focus on first-person actions.

Hand-crafted features for action recognition. Early approaches in third-person scenarios (*e.g.* surveillance videos) typically rely on hand-crafted spatio-temporal features such as HOG-3D [28], STIP [29], SIFT-3D [53] or dense trajectories [40] and combine them using bag-of-words. Egocentric videos present specific challenges, like camera motion, large occlusions, background clutter [33]. In this scenario, traditional visual features have been shown to perform poorly [12, 14, 45]. A number of approaches proposed to focus instead on object-centric representations [12, 15, 45]. Additionally, the use of egocentric cues like camera and head motion [27, 32, 51], hand motion and pose [32] and gaze information [13, 32] have been explored. Li et al. [33] provide a systematic analysis of motion, object and egocentric cues for first-person action recognition, showing that they can be combined with motion-compensated traditional features.

Learned features. With the growing availability of data [5, 6, 55, 59] and the advent of deep learning, the emphasis has shifted towards learning video features with deep neural networks. DNN architectures for action recognition can be roughly grouped into three classes: 3D ConvNets [24, 26, 61, 62, 63], Recurrent Neural Networks (RNNs) [11, 34, 43, 54] and two-stream networks [17, 57, 65]. Recently, Carreira and Zisserman [6] proposed Two-Stream Inflated 3D ConvNets (I3D), which model two streams (spatial and flow) with two 3D ConvNets.

Similar ideas have been applied to egocentric scenarios. However, progress has been limited by the lack of huge amounts of annotated data, which only recently started to become available [9, 22, 32]. Ma et al. [38] train a two-stream network to explicitly learn egocentric features like

hand segmentation and object labels. Similarly, other approaches focus on hand- or object-based features [3, 37, 58]. The recently proposed Object Relation Network (ORN) [4] and Temporal Relational Reasoning Network (TRN) [70] classify activities by learning contextual relationships between detected semantic object instances and frames. In all these works, it remains unclear how well these learned features generalize across datasets.

Transfer Learning. There exists a rich literature [8] on transfer learning of "pre-CNN" features for action recognition, focusing especially on transfer across input modalities. In videos, Karpathy et al. [26] transfer CNN-based features learned on a huge dataset of sport videos to recognize actions on UCF-101 [59]. Sigurdsson et al. [56] learn a joint representation for actions from first- and third-person videos. In the egocentric domain, Wray et al. [66] propose to map a single action to multiple verb labels to generalize video retrieval across datasets; they do not provide quantitative results. To the best of our knowledge, no work has provided an analysis of transfer learning methodologies for egocentric action recognition so far.

Few-shot Learning. While it has been actively studied for image understanding purposes [1, 18, 31, 47, 64], few shot learning has received far less attention for video-based techniques. Two common strands of research for few-shot learning are metric-learning approaches [64] that measure similarity to few-shot example inputs and meta-learning approaches [18] that learn how to update the parameters for a small number of examples. Within a metric learning setup, Mettes et al. [41] proposed to learn spatial-aware object embeddings for zero-shot action localization. [67] adopts the metric learning technique of Matching Networks and utilizes correlations to localize actions of previously unseen classes. In contrast to the existing approaches, we propose a meta-learning strategy for video understanding that learns how to update model parameters for few-shot examples.

3. Proposed Approach

We conjecture that a key problem in transferring learned representations for activity understanding in first-person videos is the strong coupling between hand-object interaction ("the action") happening in the foreground and the appearance of the background. Since datasets for first-person action are still fairly small, training an action recognition model on a given dataset inevitably causes the model to over-fit to coincidental regularities in scene appearance.

Hence, we aim to obtain better generalization by having the learned model focus on the foreground activity, while neglecting background appearance. Fortunately, the knowledge of first-person configuration provides us with ample opportunity to inject inductive biases such as the existence of left and right hands, hand-object configurations and motion, which we can exploit to decouple foreground action

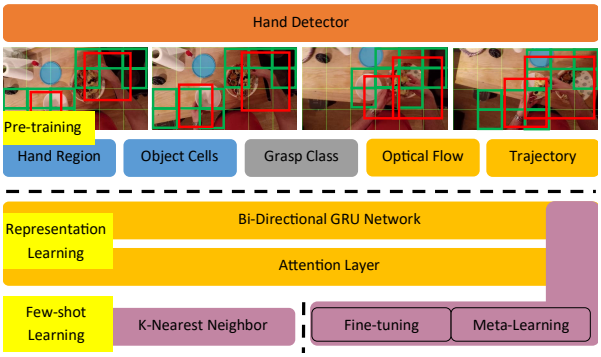

Figure 2. System overview: The hand detector provides extended *hand-context* (red boxes) and activates object cells (dark green) laid out in a fixed grid (light green). Training proceeds in three stages: (i) visual cue extractors, trained from disparate image datasets, (ii) Bi-Directional GRU and Attention layer, trained from source domain videos with action labels, and (iii) transfer learning: we experiment with fine-tuning, KNN and meta learning.

context from the background. Specifically, we extract multiple features ("visual cues") inside the hand region, as obtained from a hand bounding box detector, and use them to train temporal models from a source dataset.

Since most action recognition datasets do not contain all the relevant visual cue labels, we utilize cue-relevant image datasets to train feature representations for the visual cues. Figure 2 depicts our use of hand regions and object context regions within a multi-layer architecture. Off-the-shelf hand detection is used to define regions occupied by hands, as well as object regions nearby the hands ("object cells"). We also employ pre-trained grasp network features and motion cues such as optical flow and pixel trajectory information. Each of these features is embedded in a fixed dimensional representation that is used to train a bi-directional temporal network whose output is processed by an attention layer. The features for hand and object regions are computed using a pre-trained object detection network and grasp gestures use a pre-trained grasp network.

3.1. Visual Cues

We define a-priori visual cues and process them through their respective DNNs to compute feature descriptors to then train a temporal model. More details about the feature extraction process are available as supplementary material.

Hand detection as "hard attention mechanism". We train a Faster R-CNN [48] detector to localize the left and right ego-hands in individual video frames. We further employ a simple tracker to predict hand location from previous detections, if the detector temporarily fails to provide a bounding box. In turn, we only compute CNN-derived features for our visual cues from local regions defined around the left and right hand bounding boxes.

Local Hand Context. Intuitively, first-person actions

are implied by the region of the scene close to the subject's hands, including any objects being held in, or close to the hands. With this motivation, we extend the bounding boxes by a fixed factor s_{hand}, such that it includes local context, which we refer to as "hand-context box" (red bounding boxes in Figure 2). We compute intermediate convolutional features from a pre-trained object detector by Region-of-Interest (RoI) pooling [20] within the extended bounding box.

Object-interaction modeling. As we are in a transfer learning setting, we cannot rely on knowledge of explicit object classes for hand-object and object-object reasoning [4]. A universal action model should be able to represent entirely unknown object classes e.g. have the ability to be trained on a "kitchen" dataset and still be able to low-shot learn and test on videos depicting actions on the factory floor manipulating unknown tools, with few examples.

Thus, we draw inspiration from [52] which proposes a simple grid cell representation of the scene, to model object-object interactions on the "CLEVR", visual question-answering dataset [25]. We define a fixed grid over the frame, and "activate" the five cells surrounding the hand detection bounding box (green boxes in Figure 2). The object features are then computed as the *max-pool* operation (element-wise max) between (intermediate convolutional layer) features obtained per activated cell. We interpret this operation as computing a feature descriptor representing object-object and hand-object interactions.

Grasp features. Visual hand pose estimation is a challenging problem, which becomes even harder when hands manipulate objects. However, we do not need precise joint location to correctly interpret an interaction. With this intuition, we draw inspiration from the robotic manipulation literature, and utilize fine-grained grasp classification [50] as an approximation of hand poses, to aid action classification. Specifically, we train a CNN on cropped *hand-context* regions to predict grasp classes [50], and use the pre-logit feature vector as our grasp features.

Optical Flow. We follow common knowledge from action recognition literature [10, 57] to inject explicit motion cues as training data. Specifically, we modify an InceptionResNet v2 network [60] to feed in the horizontal and vertical optical flow channels for *N* frames centered around current frame. The network is trained to classify the optical flow frames into action classes *w.r.t.* training labels. Again, we extract intermediate convolutional features [20] within the *hand-context box* as the feature representation which gets fed into the temporal model.

Hand Trajectory. In addition to using optical flow to explicitly represent motion, we revisit pre-deep learning ideas of encoding image trajectory of interest points [40] to aid action understanding. However, we limit ourselves to providing the 2D bounding box coordinates of the hands over the past few frames as our trajectory feature.

3.2. Network Architecture for Action Training

We utilize a standard recurrent neural network (RNN) architecture as the backbone for temporal reasoning.

Temporal Modeling. Our temporal model consists of an embedding layer followed by a bi-directional GRU network [7]. The embedding layer comprises a fully connected layer of size N $\times$ 256 and layer normalization, where *N* depicts the total dimensionality of input features. This temporal network has 128 hidden units. We use the same architecture for all experiments, adapting the input dimensionality by the visual cue size, for the sake of comparing various cue-combinations.

Attention Layers. We use the recently proposed self-attention mechanism [36] to process the output from the GRU network. Self-attention mechanism allows us to encode the arbitrary size GRU outputs to a fixed sized output. In particular, given a sequence of GRU outputs $S = \{s_1, s_2, \ldots, s_t\}$ where $s_n = [\overrightarrow{GRU}(x_t, s_{t-1}), \overleftarrow{GRU}(x_t, s_{t-1})]$ (state from both temporal directions of information flow) and x_t is the frame at time t, we compute the attention matrix A with $A = softmax(W_{s2} tanh(W_{s1} S^T))$. Here, $W_{s2} \in R^{256 \times 100}$ and $W_{s1} \in R^{100 \times 3}$ denotes weight matrices while attention matrix, $A \in R^{3 \times T}$. Using A we can effectively embed the given states S to fixed length by: $E = AS$. Self attention mechanism is followed by two fully connected layers.

3.3. Training for Transfer

We argue that the visual cues introduced in Sec. 3.1 are particularly effective for transfer learning. We employ K-Nearest Neighbors (KNN) and fine-tuning of the action network as baselines for low-shot learning on a few target domain examples, transferring the RNN representation learned on the source datasets. We further introduce an attentive meta-learning mechanism, inspired by [18]. We first briefly describe our KNN and fine-tuning settings, then detail our meta-learning approach.

KNN. We use features extracted from the attention layer of our network in a KNN framework to perform few-shot matching. Specifically, we randomly select *L* samples per test class and classify an activity by a majority vote of its neighbors. Model prediction is assigned to the most frequent label among its *K* nearest neighbors. For evaluation, we repeat this process 50 times and average the results.

Fine-tuning. We select *L* samples per class and fine-tune our model using these sequences. We perform two different fine-tuning: only logits and full parameters. In our experiments all models are trained for a fixed number of updates ($10K$). We repeat fine tuning experiments 15 times for each model. The supplementary material reports the parameters

for only logits and full parameters fine tuning, as well as the standard deviation values of the model accuracy.

Attentive Model Agnostic Meta Learning (A-MAML). We build on the MAML training algorithm proposed in [18], which aims at finding an optimal set of network parameters that can be transferred to new classes using only a few training samples and updates. The algorithm consists of two parts – *meta-learner* and *learner*. While the *meta-learner* trains over a range of tasks, the individual learners optimize for specific tasks. Here, the tasks correspond to K-shot N-class action classification problems. In the course of the algorithm, first, a support set consisting of K labelled examples for N classes is selected from the full training data. In the context of few-shot learning, K here corresponds to a small number of training examples. The learner optimizes over this training set for a small number of updates. While specializing on a specific task, the network does not yet generalize across a variety of different tasks. We would like to find a set of optimal network parameters such that the overall model generalizes across tasks and task-specific learning requires less updates. To this end, a meta-objective is defined across different tasks minimizing the same training loss. We sample a new task at the end of each training and update the objective function with gradient descent on the new support set.

Experimentally, we observed convergence issues when training our network with MAML. As also reported by [2], MAML is notoriously difficult to train; the use of RNNs exacerbates this difficulty [39, 44], since its optimization relies on vanilla gradient descent during *learner* updates. We propose to overcome this problem by optimizing the task specific loss *only* over the attention layer parameters. Attention has been shown to be more effective than RNNs for transfer learning [46], but not within a meta-learning context. We demonstrate that attention-based meta-learning results in improved accuracy and higher generalization power for few-shot action recognition. In this framework, the bidirectional GRU acts as a class-agnostic embedding network, while the attention layer effectively acts as the classifier.

4. Experimental Results

Datasets. We evaluate our model on two relatively large first-person video datasets: EPIC Kitchens (EPIC) [9] and Extended GTEA Gaze+ (EGTEA) [32]. EPIC contains 55 hours of recordings, featuring 28 subjects performing daily activities in different kitchen scenarios. Each subject records and labels their own activities. EPIC provides labels as primitive actions, *i.e.* verbs that define the action of the subject (*e.g.* close, open, wash); in total, there are 125 such primitive action labels. EGTEA also records activities in kitchen scenarios. The dataset contains 106 different activity annotations, that are defined as (*verb*, *noun*) pairs.

R18* [23]	32.05
I3D-18* [4]	34.20
ORN* [4]	40.89
Ours (Hand)	38.54
Ours (Hand+Obj)	**42.21**

Table 1. Activity classification accuracy on the EPIC dataset. Entries marked with * are taken from [4].

To keep label similarity with EPIC, we consider only verb labels. This gives us 22 distinct primitive activity labels.

In order to evaluate the ability of our models on *inter-class* transfer learning, we define a specific training/test split on EPIC. We choose action classes with "enough" samples for training, and less represented classes for testing. Our training set includes frames from 12 classes (*close*, *cut*, *mix*, *move*, *open*, *pour*, *put*, *remove*, *take*, *throw*, *turn-on*, *wash*); for each of these classes, we have more than 300 sequences in EPIC. Note that these correspond to fairly generic actions, hence they are good candidates for training a base network. While there are fewer sequences for our test classes in EPIC, given the long tail distribution of activities in the dataset they provide a representative test set (we have on average 20 sequences per test class). We use 5 subjects (*S01*, *S10*, *S16*, *S23*, *S30*) for testing and the rest of the subjects for training. We train on this set our flow network and temporal models, as well as four state-of-the-art baselines – I3D, Two Stream I3D, TRN, and Two Stream TRN. We use EGTEA in our *inter-dataset* transfer experiments, and provide details about training/test split and data processing in Sec. 4.2.

In the following, we specify the datasets used to learn each of the visual cue extractors introduced in Sec. 3.1 and briefly describe the training procedures adopted for our RNN model, and for the baselines. Note that all the cue extractor networks are based on an Inception Resnet-v2 [60] backbone. For more details about training and architectures, we refer the reader to the supplemental.

Hand Detector. We use the EgoHands dataset [3] to train our hand detector. The dataset contains more than $4,000$ images, with ground-truth hand masks. We compute bounding boxes corresponding to such masks, and use them for training.

Hand-context and Object cell features. We simply extract feature descriptors for hand-context and object-cell regions from an Inception-ResNet-v2 (Faster R-CNN backbone, *Mixed_6a* features), trained for MS-COCO detection.

Grasp features. Our grasp classification network is trained on the GUN71 dataset [50]. GUN71 contains $12,000$ RGB-D images, labeled with the corresponding grasp types. There are 71 grasp types in total. We use our previously trained hand detector to estimate *hand-context* regions, and crop and resize these regions to a size of 350×350.

Optical flow. We train our flow estimation network on

EPIC, using the 12-class training set described above. We pre-compute optical flow by using the TV-L1 algorithm proposed in [68], with the same parameters used in [9]. At training time we feed 3 consecutive frames to the network, and minimize the cross-entropy loss for action classification.

Temporal Model. We train our temporal model on the same set of videos used for our flow network. We subdivide each video into segments, so that no segment is longer than 12 seconds, whereas for efficiency each mini-batch is constructed out of same-size sequences.

Baseline Models. We consider I3D [6], TRN [70] and their two-stream versions as state-of-the-art baselines, using their publicly available implementation [6, 70]. I3D and TRN take as input RGB image sequences; Two-Stream TRN and I3D take as input both RGB images and flow fields, thus requiring twice more parameters as I3D and TRN. As we did for our flow network training, we use the flow fields pre-computed with [68]. Both I3D and TRN, and their two-stream versions, require fixed input video length at training time; therefore, we subdivide each sequence into 15 segments and randomly pick one segment per sequence. More details about the training process are available as supplementary material. At test time, we run the model 10 times and select an output action class via majority voting.

4.1. Visual Cue Effectiveness

The first experiment we propose to assess the effectiveness of our visual cues aims at comparing our approach against three state-of-the art methods for action recognition: ORN [4], ResNet (R18) and I3D-18 as implemented in [4]. We evaluate all the approaches on EPIC, using the train/test split described in [4]. Results are shown in Table 1. Note here that while the baseline methods train their networks on the EPIC dataset, we do not rely on training on the target dataset. However, our approach still outperforms the baselines demonstrating strong generalization ability. We observe that our class-agnostic object-object interaction features add to the accuracy obtained for *hand-context* features alone, and outperform all three state-of-the-art competitors.

In a second experiment, we evaluate our visual cues when training and test sets have the same activity labels, *i.e*. the 12 classes from EPIC. We refer to this experiment as *Intra-dataset* (see first two columns of Table 2), and compare against I3D [6] , Two Stream I3D [6], TRN [70] and Two Stream TRN [70] trained as described above. In addition, we design two further baseline methods: *Coco-Global* and *Coco-Global+Flow-Global*. Here, *Coco-Global* extracts global features from the input RGB frame (obtained from a Inception-ResNet-v2 network trained for MS-COCO [35] detection, *Mixed_6a* features), whereas *Coco-Global+Flow-Global* extracts additional global features from optical flow, plugging these features into our temporal model. These baselines mimic most state-of-the-art approaches to action recognition which feed similar global features into their respective temporal models [58, 30]. Also in this case, Table 2 shows that our complete pipeline is able to outperform compared approaches by significant margins, *i.e.* 7.5 percentage points (pp) and 1.6 pp versus best baseline model for KNN and Softmax classifiers respectively.

4.2. Inter-dataset Transfer

We use the EGTEA dataset [32] for our inter-dataset transfer experiments. We sub-sampled the original sequences from 24 to 12 fps, and use the train and test splits provided in the dataset. We select those 10 classes that are also a subset of the EPIC training set (*i.e*., *close*, *cut*, *mix*, *move*, *open*, *pour*, *put*, *take*, *turn-on*, *wash*). Optical flow is computed via TV-L1 [68]. Table 2 and Table 3 report the results respectively in case of the KNN and fine tuning evaluations. In particular, Table 2 reports the percentage of correct classification of the KNN scheme using k equal to, respectively, 1, 10 and 20 nearest samples out of the selected number of samples from each class (*i.e*., 1 with k=1, 10 with k=10, 30 with k=20). Similarly, Table 3 shows the percentage of correct classification in the *fine tuning* experiment, where in this case each classification is obtained from the network's softmax, again tested in the three cases of 1, 10, 30 selected samples from each class.

We obtain the best results by combining *Hand*, *Obj*, *Flow*, *Grasp* and *Traj* features for 1 and 10 samples. For the case of 30 samples, Two-Stream TRN obtains the best results. We can argue that the superior performance is thanks to the effective segmentation of the activity from the background carried out by the employed visual cues.

Intuitively, we can argue that object-related cues (*i.e*., *Obj*) provide the biggest contribution in terms of performance since similar activities deploy similar objects. Also, we can observe that using trajectory-related cues consistently improves the accuracy in all cases. Moreover, the poor performance of flow-related cues in the KNN experiment can be explained by the fact that the two datasets significantly differ in terms of recording settings and camera setup: EPIC is recorded with a head-mounted GoPro, while EGTEA is recorded using eye-tracking glasses. Note that, in the *fine tuning* experiment, the *Flow* cue is the second best cue in terms of accuracy after the *Grasp* one.

4.3. Inter-class Transfer

We conduct an *inter-class* transfer experiment on EPIC by means of 14 classes (namely *adjust*, *check*, *dry*, *empty*, *fill*, *flip*, *insert*, *peel*, *press*, *scoop*, *shake*, *squeeze*, *turn*, *turn-off*). The results are shown in Table 2 and Table 3. We follow the same experiment protocol described in Sec. 4.2.

As for the KNN evaluation, the proposed method clearly

	EPIC: Intra-dataset		EPIC: Inter-class			EGTEA: Inter-dataset		
	KNN-20	SftMx	KNN-1	KNN-10	KNN-20	KNN-1	KNN-10	KNN-20
I3D [6]	26.05	42.3	12.8	18.2	19.4	12.3	19.9	25.3
Two-Stream I3D [6]	32.11	54.2	13.7	21.7	25.1	24.8	25.6	34.7
TRN [70]	34.0	54.5	13.2	22.7	27.9	18.2	33.7	42.5
Two-Stream TRN [70]	40.3	60.6	14.3	26.4	29.4	20.3	35.7	44.3
Coco-Global	40.0	41.7	12.1	17.5	20.4	24.6	35.5	40.1
Coco-Global + Flow-Global	48.9	52.2	14.5	21.3	25.2	25.6	37.3	42.5
Hand	48.1	52.2	16.5	26.2	30.7	26.5	37.2	41.9
+ Obj	50.0	55.9	16.9	27.8	32.2	29.8	**42.8**	48.0
+ Flow	53.2	57.5	17.0	27.6	32.4	25.9	36.4	41.6
+ Traj	49.7	53.7	15.2	22.7	27.2	27.2	37.6	42.5
+ Grasp	49.3	54.8	16.9	28.5	**36.7**	24.2	33.7	38.4
+ Obj + Flow	54.7	59.9	17.0	27.6	32.4	26.9	39.1	**49.8**
+ Obj + Flow + Grasp	56.4	61.6	**19.9**	29.5	35.1	29.1	41.4	46.6
+ Obj + Flow + Grasp + Traj	**56.9**	**62.2**	18.7	**30.3**	34.6	**31.5**	41.0	47.0

Table 2. Transfer learning experiments with KNN. The best and second best results are highlighted with **bold** and blue fonts, respectively.

	EPIC: Inter-class			EGTEA: Inter-dataset		
	1	10	30	1	10	30
I3D [6]	13.2	20.1	23.6	15.7	34.4	36.3
Two-Stream I3D [6]	14.8	23.4	31.8	19.8	38.0	43.5
TRN [70]	16.1	27.5	35.7	23.5	51.1	58.2
Two-Stream TRN [70]	18.7	29.7	39.3	26.2	55.8	**62.5**
Coco-Global	10.9	16.9	26.5	15.9	43.3	51.0
Coco-Global + Flow-Global	13.8	22.1	30.0	21.0	46.2	52.3
Hand	16.4	27.6	34.8	21.7	48.4	56.9
+ Obj	18.1	28.3	36.4	29.7	50.6	57.7
+ Flow	17.9	26.9	35.6	22.7	53.8	58.2
+ Traj	15.2	27.1	35.0	21.8	51.4	57.3
+ Grasp	19.7	29.8	38.1	23.8	52.7	58.5
+ Obj + Flow	17.8	27.0	37.9	28.3	54.5	59.0
+ Obj + Flow + Grasp	**20.6**	29.7	**39.8**	30.9	56.9	60.4
+ Obj + Flow + Grasp + Traj	20.3	**31.7**	38.5	**32.0**	**57.1**	61.2

Table 3. Transfer learning experiments with fine tuning. The best and second best results are highlighted with **bold** and blue fonts, respectively.

outperforms the state of the art by a large margin. In particular, the *Hand* + *Grasp* combination yields a 36.7% accuracy using 20 nearest samples: this is more than a 7.0 percentage improvement on the performance of the Two-Stream TRN, that scores 29.4%. A similar trend is exhibited in the *fine tuning* experiment, in this case the best combination is *Hand* + *Obj* + *Flow* + *Grasp* (*i.e.*, 39.8%), showing the importance of combining together multiple heterogeneous features.

In general, we observe that grasp contributes most to *inter-class* transfer accuracy, as the grasp type used for a specific object is quite discriminative for the specific activity being carried out, so that the temporal model can learn to distinguish actions only looking at a small number of training samples. Also, global features seem to under-perform in this case, this also highlights the importance of using the hand as a hard attention mechanism.

4.4. Learning to Transfer

We further assess the effectiveness of different transfer learning strategies and demonstrate that our attentive meta transfer-learning model, A-MAML, achieves better accuracy than competing transfer learning strategies for few-shot action recognition. Table 4 demonstrates our results for the 5-shot 5-class action recognition experiment on the EPIC dataset using the same training and test splits as defined

Figure 3. Qualitative results for our inter-class experiments on EPIC. Frames are selected uniformly at random; different rows correspond to different activities. The first 2 rows show correct predictions; the last 2 rows highlight failure cases. Even when wrong, our predictions still provide a reasonable explanation of the scene.

Feature Set	RNN-A-KNN	RNN-FT	RNN-A-FT	RNN-MAML	RNN-A-MAML	A-MAML
Coco-Global + Flow-Global	29.9	30.2	31.3	33.5	35.2	**35.7**
Ours (Hand Only)	30.8	31.1	32.8	34.6	35.3	**36.9**
Ours (All)	34.5	35.0	36.1	38.6	40.2	**41.4**

Table 4. Comparison of different transfer learning strategies on 5-shot 5-class action recognition experiments on EPIC. Our attention-based meta learning approach, A-MAML, significantly outperforms other transfer learning strategies. For *KNN* and *FT*, we follow the standard training procedure described in Sec. 3.3. Methods in [1,3,5,6] columns use identical models. For the case of [2,4] columns, methods use models without attention layer. The best and second best results are highlighted with **bold** and blue fonts, respectively.

in Sec. 4.3.

During training meta-learning models, 5 classes are randomly selected from which 5 samples are drawn for the learner updates. Hence, we evaluate our few-shot learning accuracy for the 5-way, 5-shot classification setting. The meta batch-size is set to 10 tasks. While the learner is trained with 5 steps, the meta-learner is trained for 2K steps. We use vanilla gradient descent with a learning rate of 0.001 within a specific task and Adam with a learning rate of 0.001 for the meta-learner updates.

Table 4 shows that meta-learning based transfer learning outperforms the baselines by a large margin and the proposed A-MAML training further improves the accuracy on all feature combinations. While training attention jointly with RNN yields compelling action recognition accuracy, the simpler model that relies on only attention-based training yields the best accuracy (A-MAML). Ultimately, our attention-based model provides us with a more structured memory for handling long-term dependencies as compared to RNNs and yields robust transfer performance across diverse tasks, as also observed by [46].

In Fig. 3, we visualize examples of our model predictions for 4 different activities for both correct predictions and failure cases. We note that some annotations in the dataset are confusing or wrongly annotated, for instance the third row depicts an example of the *turn* activity (the subject is turning on the kettle), while such activity is defined for most examples in the dataset as that of physically turning or rotating an object. This confusion is also label-wise other than sample-wise: *e.g.*, in addition to the *turn* label, the dataset also contains the *turn-on* label.

5. Conclusion

We proposed a methodology that does not demand detailed annotation, and utilizes contextual visual cues for effective activity learning. Specifically, we introduced a simple approach to decouple the foreground action from background appearance via hand detection. We leverage a set of cues, including class-agnostic object-object interaction, hand grasp, optical flow, and hand trajectory to train action RNNs, that we demonstrate to have superior transferability than state-of-the-art action models. We further propose Attentive MAML, an algorithm combining MAML with an attention mechanism for effective transfer learning. We believe that our inter-class and inter-dataset transfer learning results represent a step-forward in generalization across significant environment and object appearances.

References

[1] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. de Freitas. Learning to learn by gradient descent by gradient descent. In *NIPS*, 2016. 3

[2] A. S. Antreas Antoniou, Harrison Edwards. How to train your maml. In *ICLR*, 2019. 5

[3] S. Bambach, S. Lee, D. J. Crandall, and C. Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 3, 5

[4] F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori. Object level visual reasoning in videos. In *ECCV*, 2018. 3, 4, 5, 6

[5] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 2

[6] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the Kinetics dataset. In *CVPR*, 2017. 1, 2, 6, 7

[7] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 4

[8] D. Cook, K. Feuz, and N. Krishnan. Transfer learning for activity recognition: A survey. *Knowledge and Information Systems*, 36(3):537–556, 2013. 3

[9] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. In *ECCV*, 2018. 2, 5, 6

[10] J. W. Davis and A. Bobick. The Representation and Recognition of Action Using Temporal Templates. In *CVPR*, 1997. 4

[11] J. Donahue, L. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional netowrks for visual recognition and description. In *CVPR*, 2015. 2

[12] A. Fathi, A. Farhadi, and J. Rehg. Understanding egocentric activities. In *ICCV*, 2011. 2

[13] A. Fathi, Y. Li, and J. Rehg. Learning to recognize daily actions using gaze. In *ECCV*, 2012. 2

[14] A. Fathi and J. Rehg. Modeling actions through state changes. In *CVPR*, 2013. 2

[15] A. Fathi, X. Ren, and J. Rehg. Learning to recognize objects in egocentric activities. In *CVPR*, 2011. 2

[16] C. Feichtenhofer, A. Pinz, R. P. Wildes, and A. Zisserman. What have we learned from deep representations for action recognition? In *CVPR*, 2018. 1

[17] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 2

[18] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017. 2, 3, 4, 5

[19] C. Finn, K. Xu, and S. Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 9537–9548, 2018. 2

[20] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 4

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, 2014. 1

[22] R. Goyal et al. The "Something Something" Video Database for Learning and Evaluating Visual Common Sense. In *ICCV*, 2017. 2

[23] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[24] S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. *IEEE TPAMI*, 35(1):221–231, 2013. 2

[25] J. Johnson, B. Hariharan, L. v. d. Maaten, F.-F. Li, L. Zitnick, and R. Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *CVPR*, 2017. 4

[26] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional networks. In *CVPR*, 2014. 2, 3

[27] K. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In *CVPR*, 2011. 2

[28] A. Kläser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *BMVC*, 2008. 2

[29] I. Laptev. On Space-Time Interest Points. *IJCV*, 64(2–3):107–123, 2005. 2

[30] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. Temporal Convolutional Networks for Action Segmentation and Detection. In *CVPR*, 2017. 6

[31] K. Li and J. Malik. Learning to optimize. In *ICLR*, 2017. 3

[32] Y. Li, A. Fathi, and J. Rehg. Learning to predict gaze in egocentric video. In *ICCV*, 2013. 2, 5, 6

[33] Y. Li, Z. Ye, and J. M. Rehg. Delving into egocentric actions. In *CVPR*, 2015. 2

[34] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. Snoek. VideoLSTM convolves, attends and flows for action recognition. *CVIU*, 166(C):41–50, 2018. 2

[35] T. Lin, M. Maire, S. Belongie, J. Hays, P. P., D. Ramanan, P. Dollar, and C. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 6

[36] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2017. 4

[37] Y. S. M. Cai, K. M. Kitani. Understanding Hand-Object Manipulation with Grasp Types and Object Attributes. In *Robotics: Science and Systems*, 2016. 3

[38] M. Ma, H. Fan, and K. Kitani. Going deeper into first-person activity recognition. In *CVPR*, 2016. 2

[39] J. Martens and I. Sutskever. Learning recurrent neural networks with hessian-free optimization. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1033–1040. Citeseer, 2011. 5

[40] P. Matikainen, M. Hebert, and R. Suthankar. Trajectons: Action Recognition Through the Motion Analysis of Tracked Features. In *ICCVW*, 2009. 2, 4

[41] P. Mettes and C. G. M. Snoek. Spatial-aware object embeddings for zero-shot localization and classification of actions. In *ICCV*, 2017. 3

[42] A. Miech, J.-B. Alayrac, P. Bojanowski, I. Laptev, and J. Sivic. Learning from Video and Text via Large-Scale Discriminative Clustering. In *ICCV*, 2017. 1

[43] J. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015. 2

[44] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013. 5

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

[46] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. 5, 8

[47] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 2, 3

[48] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*, 2015. 3

[49] M. Rodriguez, C. Orrite, C. Medrano, and D. Makris. Fast Simplex-HMM for One-Shot Learning Activity Recognition. In *CVPRW*, 2017. 1

[50] G. Rogez, J. S. Supancic, and D. Ramanan. Understanding everyday hands in action from rgb-d images. In *ICCV*, 2015. 4, 5

[51] M. Ryoo and L. Matthies. First-person activity recognition: What are they doing to me? In *CVPR*, 2013. 2

[52] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrapm. A simple neural network module for relational reasoning. In *NIPS*, 2017. 4

[53] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 357–360. ACM, 2007. 2

[54] S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. In *ICLR Workshops*, 2016. 2

[55] A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in Homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 2

[56] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari. Actor and Observer: Joint modeling of first and third-person videos. In *CVPR*, 2018. 3

[57] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 2, 4

[58] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A Multi-Stream Bi-Directional Recurrent Neural Network for Fine-Grained Action Detection. In *CVPR*, 2016. 3, 6

[59] K. Soomro, A. Zamir, and S. M. UCF101: A dataset of 101 human action classes from videos int he wild. In *CRCV-TR-12-1*, 2012. 2, 3

[60] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017. 4, 5

[61] G. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *ECCV*, 2010. 2

[62] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015. 2

[63] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE TPAMI*, 40(6):1510–1517, 2018. 2

[64] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one-shot learning. In *NIPS*, 2016. 3

[65] L. Wang, Y. Xiong, Z. Whang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal Segment Networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 2

[66] M. Wray, D. Moltisanti, and D. Damen. Towards an Unequivocal Representation of Actions. In *CVPRW*, 2018. 1, 3

[67] H. Yang, X. He, and F. Porikli. One-shot Action Localization by Learning Sequence Matching Network. In *CVPR*, 2018. 1, 3

[68] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint Pattern Recognition Symposium*, pages 214–223. Springer, 2007. 6

[69] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 1

[70] B. Zhou, A. Andonian, A. Oliva, and A. Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018. 3, 6, 7