My View is the Best View: Procedure Learning from Egocentric Videos

Siddhant Bansal[1], Chetan Arora[2], and C.V. Jawahar[1]

[1] Center for Visual Information Technology, IIIT, Hyderabad
[2] Indian Institute of Technology, Delhi
siddhant.bansal@research.iiit.ac.in

Abstract. Procedure learning involves identifying the key-steps and determining their logical order to perform a task. Existing approaches commonly use third-person videos for learning the procedure, making the manipulated object small in appearance and often occluded by the actor, leading to significant errors. In contrast, we observe that videos obtained from first-person (egocentric) wearable cameras provide an unobstructed and clear view of the action. However, procedure learning from egocentric videos is challenging because (a) the camera view undergoes extreme changes due to the wearer's head motion, and (b) the presence of unrelated frames due to the unconstrained nature of the videos. Due to this, current state-of-the-art methods' assumptions that the actions occur at approximately the same time and are of the same duration, do not hold. Instead, we propose to use the signal provided by the temporal correspondences between key-steps across videos. To this end, we present a novel self-supervised Correspond and Cut (CnC) framework for procedure learning. CnC identifies and utilizes the temporal correspondences between the key-steps across multiple videos to learn the procedure. Our experiments show that CnC outperforms the state-of-the-art on the benchmark ProceL and CrossTask datasets by 5.2% and 6.3%, respectively. Furthermore, for procedure learning using egocentric videos, we propose the EgoProceL dataset consisting of 62 hours of videos captured by 130 subjects performing 16 tasks. The source code and the dataset are available on the project page https://sid2697.github.io/egoprocel/.

1 Introduction

Imagine showing an autonomous agent how to prepare a sandwich, and it learns the steps required for it! Motivated by this vision, our work focuses on developing a framework that allows an agent to identify the steps required to perform a task and their order after observing multiple visual demonstrations by experts. Given a set of instructional videos for the same task, procedure learning [18, 19, 63] broadly consists of two steps, (a) assigning all the frames to the K key-steps (including the background), and (b) discovering the logical ordering of the key-steps required to perform the task. Procedure learning differs from action segmentation as it aims to *jointly* segment common key-steps (actions required to accomplish a task, as shown in Figure 1) across a given set of videos. In contrast,

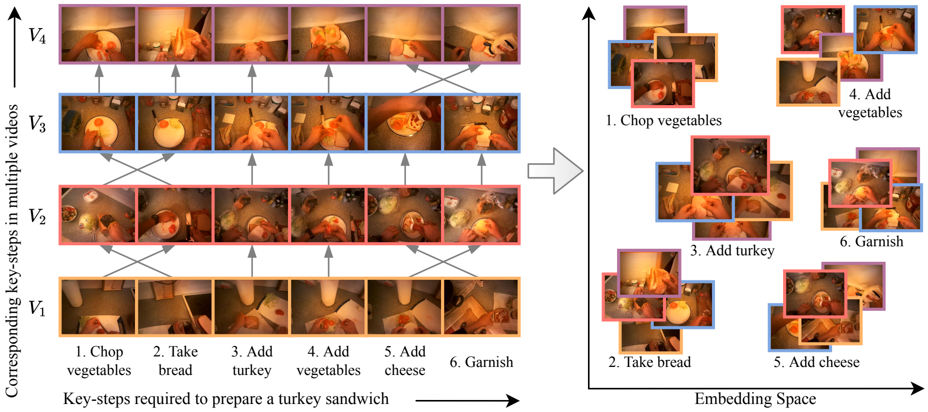

Fig. 1. The left-hand side figure shows six key-steps required to prepare a turkey sandwich [47] across four egocentric videos. The arrows among the videos highlight the change in the ordering of corresponding key-steps. This work utilizes these correspondences and aims to learn an embedding space where the corresponding key-steps have similar embeddings (right-hand side figure). To this end, we propose CnC which learns the embedding space and utilizes it to localize the key-steps and identify their ordering.

action segmentation aims to identify actions (unrelated to their relevance to accomplishing a task) from a *single* video. Furthermore, procedure learning deals with additional or missing key-steps and background actions unrelated to the task and identifies an ordering of the key-steps.

Existing instructional videos datasets [2,19,33,38,51,67,77,79] majorly consist of third-person videos. Here, the camera is kept far from the expert, to avoid interference in the actual task. Due to this, the manipulated objects are typically small or sometimes invisible. Additionally, third-person videos can be captured from various positions, leading to wide variations in the camera viewpoints for the same task [11]. Further, most datasets comprise videos scraped from the internet (YouTube) [19,33,51,67,79], which are noisy and have large irrelevant segments. In contrast, egocentric cameras are typically harnessed to the subject's head and have a standardized location. They provide a clearer view of the executed task, including the manipulated objects. As a result, recent works have introduced datasets consisting of egocentric videos [9, 24, 32, 47, 57, 64], which have proven helpful for various tasks [23,31,46,54,65].

Motivated by the advantages of egocentric videos over third-person videos, we propose an egocentric videos dataset for procedure learning: EgoProceL. EgoProceL consists of 62 hours of egocentric videos of 16 tasks ranging from making a salmon sandwich to assembling a Personal Computer (PC), thereby ensuring diversity of tasks and facilitating generalizable methods. However, egocentric videos come with their own set of challenges. For example, the camera view undergoes extreme movements due to the wearer's head motion, introducing frames unrelated to the activity and unavailability of the actor's pose [65].

To overcome the challenges and learn the procedure from egocentric videos, we propose utilizing the signal provided by temporal correspondences across videos. As shown in Figure 1, critical moments like putting a slice of turkey on the bread while preparing a turkey sandwich are present across all the videos. To exploit the signal provided by such temporal correspondences, we propose a self-supervised, three-stage, Correspond and Cut (CnC) framework for procedure learning. The first stage of the CnC uses the proposed self-supervised TC3I loss to learn an embedding space such that the same key-steps across the videos have similar embeddings (Figure 1). The second stage consists of the proposed ProCut Module (PCM). PCM performs clustering on the learned embeddings and assigns each frame to a key-step. The final stage of CnC creates a key-step sequence for each video and infers relevant ordering to perform the task.

Current works mostly use frame-wise metrics to evaluate the models developed for procedure learning [18, 19, 40, 63, 70]. While these metrics evaluate the procedure reasonably well compared to simply calculating the accuracy, they do not suit datasets with significant class imbalance. Furthermore, procedure learning datasets consist of significant background frames [79]. Hence, a model assigning all the frames to the background might achieve high scores. We propose to solve this problem by calculating the scores via the contribution of each key-step, leading to lower scores when models assign most of the frames to the background. Further, when comparing with the previous works, (a) we use CnC on standard third-person benchmark datasets [19, 79] and (b) employ existing metrics to evaluate. We show that CnC outperforms the state-of-the-art techniques for procedure learning (Table 2).

Contributions: The major contributions of our work are:

- To facilitate procedure learning from egocentric videos, we create the EgoProceL dataset. The dataset consists of 62 hours of videos captured by 130 subjects performing 16 tasks.
- We propose CnC, which utilizes the proposed TC3I loss and PCM to identify the key-steps and their ordering required to perform a task.
- We investigate the usefulness of egocentric videos over third-person videos for procedure learning. We observe an average improvement of 2.7% in the F1-Score when using egocentric videos instead of third-person videos.
- The EgoProceL dataset and the code written for this work are released on http://cvit.iiit.ac.in/research/projects/cvit-projects/egoprocel (mirror link).

2 Related Works

We aim to perform procedure learning in a self-supervised fashion, unlike previous works [53, 61, 77], which assume the availability of mapping between video frames and key-steps. Also, different from weakly supervised approaches [3, 7, 13, 30, 44, 45, 59, 60, 79], we neither use the number of key-steps required to perform the task nor an ordered or unordered list, as it requires viewing the videos or defining heuristics, leading to scalability issues [18, 19]. Additionally, learning various procedures requires numerous videos and annotating all the videos would consume considerable resources. Motivated by this, we create CnC as a self-supervised framework for procedure learning to create a scalable and efficient solution.

Multimodal Procedure Learning: Another class of methods work with multimodal data, like narrated text and videos [2, 10, 14, 22, 50, 62, 63, 76, 78]. These works use Automatic Speech Recognition (ASR) to obtain the text, which is not perfect. Due to this, the output needs to be manually cleaned, which is not scalable. Additionally, such methods assume an alignment between the text and videos [2, 50, 76], which might not be accurate for most cases [18, 19]. Instead, we use only the visual modality as an input to the framework. Due to this, we eliminate the need to obtain narrations that might be inaccurate and make our framework scalable.

Learning Key-step Ordering: Current works do not capture different key-step ordering to perform the same task. They either assume a strict ordering [19, 40, 70] or do not predict the order [18, 63]. However, we observe that subjects perform the same task in multiple ways (Figure 1), motivating us to capture different ways to accomplish the task. Therefore, the final stage of CnC aims to create a key-step order for each video and infer the relevant ordering to perform the task.

Representation Learning for Procedure Learning: Existing works on procedure learning employ various ways to create frame-wise features. To learn the representation space, Kukleva *et al*. [40] use relative timestamps of frames, and Vidal *et al*. [70] predict the representation and timestamps of the future frames. On the other hand, Elhamifar *et al*. either use the latent states obtained from an HMM [19] or discover and utilize attention features from individual frames [18]. However, these methods do not exploit the signal provided by temporal correspondences, which is crucial for procedure learning, as we show in this work.

Self-Supervised Representation Learning: Learning a representation space without annotations saves substantial time and energy when creating deep learning solutions. Motivated by this, recent works explore various pretext tasks to generate supervision signals for training deep learning architectures [6, 28, 68, 69, 73]. A few pretext tasks for learning image representations include image colourization [41, 42], object counting [48, 55], solving jigsaw puzzles [5, 36], predicting image rotations [20, 37], and reconstructing input images [29] from noise [71]. Pretext tasks for learning video representations include predicting future frames [1, 12, 26, 35, 66, 72], using temporal order and coherence as labels [21, 34, 43, 52, 75], and predicting the arrow of time [74].

Video representation learning methods mentioned above employ a single video. However, we want to identify similar key-steps in multiple videos for procedure learning. To this end, we build upon existing video alignment techniques [16, 27] and devise a loss function that works well for procedure learning. Note that procedure learning aims to find key-steps across a given set of videos; hence, it differs from video alignment.

Table 1. Comparison of datasets for Procedure Learning. The average number of key-steps and video length for EgoProceL are the highest, highlighting the complexity of the procedures included in EgoProceL

Dataset	Egocentric View	Manually Created	Avg. key-steps	Avg. Video Length (sec)	#tasks
Breakfast [38]	✗	✓	5.1	137.5	10
Inria [2]	✗	✗	7.1	178.8	5
ProceL [18]	✗	✗	8.3	251.5	12
CrossTask [79]	✗	✗	7.4	297	**18**
EgoProceL (ours)	✓	✓	**8.7**	**769.2**	16

3 EgoProceL: Egocentric Video Dataset for Procedure Learning

The EgoProceL dataset focuses on the key-steps required to perform a task instead of every action in the video. To construct EgoProceL, we take two approaches: (a) identifying publicly available datasets that we annotate for key-steps; (b) recording new tasks to expand the range of tasks. We follow the following criteria to shortlist from the public datasets: (1) The task should require multiple key-steps to perform. For example, preparing a sandwich involves a minimum of four key-steps [11]. (2) Videos of the same task must contain a similar set of key-steps. However, the order of the key-steps can differ. (3) To compare the performance of CnC in egocentric and third-person views, we require a dataset with recordings of the same task in both views. (4) We prefer longer videos with sparse key-steps to generate practical solutions.

We select CMU-MMAC [11], EGTEA Gaze+ [47], MECCANO [58], and EPIC-Tents [32] based on the above criteria. CMU-MMAC contains recordings of subjects performing the same task from one egocentric and four third-person views. Therefore, by using it, we compare the performance of CnC between egocentric and third-person views. Though these four datasets include a diverse range of tasks, they do not contain tasks where the subject works in a constrained environment and deals with small objects (*e.g.*, screws). To alleviate this, we include manually recorded videos of assembling and disassembling a Personal Computer (PC). This addition makes the dataset diverse and challenging in terms of variability in the size of objects involved and the complexity of key-steps (*e.g.*, fixing the motherboard requires fastening nine screws).

EgoProceL contains videos and key-step annotations for multiple tasks from CMU-MMAC [11] and EGTEA Gaze+ [47] and individual tasks like toy-bike assembly [58], tent assembly [32], PC assembly, and PC disassembly. EgoProceL consists of 62 hours of annotated egocentric videos, including 16 tasks with an average duration of 13 minutes. To annotate the videos for key-steps, we create a list of key-steps for each task, *e.g.*, assembling a PC requires 'Fix motherboard', 'Fix hard disk', ..., 'Place the cabinet cover'. We use ELAN [17] to annotate each video by marking the start and end location during which the key-step occurs.

Along with various procedure learning tasks, EgoProceL is appropriate for understanding hand-object interaction, action forecasting and recognition, and

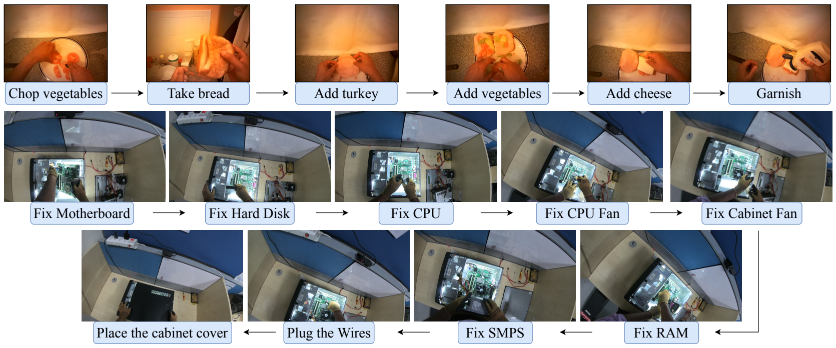

Fig. 2. Key-step annotations for making turkey sandwich [47] and assembling a PC.

a shared study of videos and text. Figure 2 shows some example annotations and Table 1 compares EgoProceL with existing datasets. We also considered a few other datasets that did not satisfy the requirements mentioned above [9,64]. The reasons for their non-inclusion are given in the supplementary.

4 Correspond and Cut Framework for Procedure Learning

Humans often follow the same steps to perform any particular task, though the order of steps might be different. This work proposes a methodology which, given a set of videos of humans performing a task, learns similar embeddings across videos for the key-steps required to complete a task. Once we have the embeddings, learning a procedure reduces to clustering the embeddings for localizing the key-steps among all the videos. To learn the embeddings, we exploit temporal correspondences between the videos of the same task. For that purpose, we train a representation learning network using the proposed TC3I loss. TC3I builds on top of existing temporal video alignment methods [16,27]. After learning the embeddings, we use PCM, shown in Figure 3, to cluster and localize the underlying key-steps. PCM models the clustering problem as a multi-label graph cut problem and solves it to localize the key-steps. Once we localize the key-steps using PCM, we use the frame's relative location in a video to generate the key-step ordering for each video.

Notation: CnC takes in $V = \{V_i : i \in \mathbb{N}, 1 \leq i \leq n\}$ untrimmed videos of the same task, where n is the total number of videos. Each of the n videos can have a different number of frames. We denote the embedding function used to generate the frame-level embeddings as f_θ, which is a neural network with parameters θ. A video V_k with m frames is denoted as $V_k = \{f_k^1, f_k^2, \ldots, f_k^m\}$ and the video's frame-level embeddings are denoted as $f_\theta(V_k) = \{v_k^1, v_k^2, \ldots, v_k^m\}$. We assume K key-steps in a task, where K is a hyper-parameter.

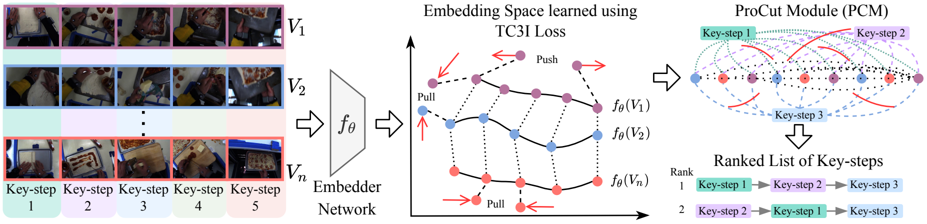

Fig. 3. Correspond and Cut (CnC) framework for Procedure Learning. CnC takes in multiple videos from the same task and passes them through the embedder network trained using the proposed TC3I loss. The goal of the embedder network is to learn similar embeddings for corresponding key-steps from multiple videos and for temporally close frames. The ProCut Module (PCM) localizes the key-steps required for performing the task. PCM converts the clustering problem to a multi-label graph cut problem solved using the Alpha Expansion algorithm [4]. The output provides the assignment of frames to the respective key-steps and their ordering.

4.1 Learning the Embeddings using the TC3I loss

We aim to learn similar embeddings for the frames with comparable semantic information across different temporal locations from multiple videos. For that purpose, we use Temporal Cycle Consistency (TCC) [16] to find correspondences across time in videos.

Consider two videos V_1 and V_2, with lengths p and q, respectively. To check if a point v_1^i in V_1 is cycle consistent, its nearest neighbour $v_2^j = \arg\min_{v_2 \in V_2} \|v_1^i - v_2\|$ is calculated in V_2. Then the process is repeated for v_2^j in V_1 to get $v_1^k = \arg\min_{v_1 \in V_1} \|v_2^j - v_1\|$. If $i = k$, then the point is considered cycle consistent. An acceptable embedding space consists of a maximum number of cycle-consistent points for a pair of sequences. Specifically, for a point v_1^i in V_1, we determine its soft nearest neighbor $\widetilde{v}_2$ in V_2 by using the softmax function as follows:

$$\widetilde{v}_2 = \sum_j \alpha_j v_2^j, \quad \text{where} \quad \alpha_j = \frac{e^{-\|v_1^i - v_2^j\|^2}}{\sum_k e^{-\|v_1^i - v_2^k\|^2}}. \tag{1}$$

Here α_j signifies the *similarity* between v_1^i and individual $v_2^j \in V_2$. Once we have the soft nearest neighbor, a similarity vector β_1^i is calculated. β defines the proximity between $\widetilde{v}_2$ and each frame $v_1^k \in V_1$ as:

$$\beta_1^i[k] = \frac{e^{-\|\widetilde{v}_2 - v_1^k\|^2}}{\sum_j e^{-\|\widetilde{v}_2 - v_1^j\|^2}}. \tag{2}$$

As β is a discrete distribution of similarities over time, it peaks around the i^{th} time index. To avoid this, a Gaussian prior is applied to β by minimizing the normalized squared distance $\frac{|i-\mu|^2}{\sigma^2}$ as the objective. By applying additional variance regularization, β is enforced to be peaky around i. Hence, the final cycle consistency loss between videos V_1 and V_2, corresponding to i^{th} frame of V_1 is:

$$L(V_1, V_2, v_1^i) = \frac{|i-\mu|^2}{\sigma^2} + \lambda \log(\sigma). \tag{3}$$

Here, $\mu_i = \sum_k \beta_1^i[k] \times k$ and $\sigma_i^2 = \sum_k \beta_1^i[k] \times (k - \mu_i)^2$, and λ is the regularization weight. Formulating TCC in this way ensures the model is not heavily penalized when it cycles back to close-by frames.

We observe that there are many repetitive frames in egocentric videos because of which cycle consistency loss often loops back to similar but temporally far-away frames. To alleviate the issue, we utilize the Contrastive-Inverse Difference Moment (C-IDM) loss [27] (a modified form of Inverse Difference Moment [8]) for applying temporal regularization on each video. The C-IDM loss between the two frames i and j of a video V_1 is computed as:

$$I(V_1, i, j) = (1 - \mathcal{N}(i,j))\, \gamma(i,j) \max\left(0, \zeta - d(i,j)\right) + \mathcal{N}(i,j) \frac{d(i,j)}{\gamma(i,j)}. \tag{4}$$

Here, $\gamma(i,j) = (i-j)^2 + 1$, $d(i,j)$ is the Euclidean distance between $f_\theta(v_1^i)$, and $f_\theta(v_1^j)$, ζ is the margin parameter, and $\mathcal{N}$ is the neighborhood function such that, $\mathcal{N}(i,j) = 1$ if $|i-j| \leq \sigma$, and 0 otherwise. Here, σ is the window size for separating temporally far away frames. The C-IDM loss encourages the embeddings of the temporally close frames to be similar and the embeddings of temporally far frames to be dissimilar. The final loss combines both TCC and C-IDM (referred to as TC3I loss from now on):

$$\begin{aligned} \text{TC3I}(V_1, V_2) = & \sum_{i \in V_1} L(V_1, V_2, v_1^i) + \sum_{j \in V_2} L(V_1, V_2, v_2^j) \\ & + \xi \sum_{i \in V_1} \sum_{j \in V_1} I(V_1, i, j) + \xi \sum_{i \in V_2} \sum_{j \in V_2} I(V_2, i, j). \end{aligned} \tag{5}$$

Here, ξ is a regularization parameter.

4.2 Localizing the Key-Steps using the ProCut Module

Once we learn the embeddings, we aim to localize the key-steps required for performing the task. Kukleva *et al.* [40] localize the key-steps by generating K clusters of embeddings using the K-Means algorithm [49]. However, they need to assume a fixed order of key-steps to assign frames to the key-steps. Instead, we propose a novel ProCut Module (PCM) for the purpose. PCM converts the clustering problem to a multi-label graph cut problem [25], as described below.

Let $G = \langle V, E \rangle$ be a graph consisting of a set of nodes V and a set of directed edges E connecting them. The node set V consists of K *terminal nodes* representing the key-steps, and *non-terminal nodes* (equal to the number of frames) representing the embeddings of the frames generated using the Embedder network. There are two kinds of edges in the graph: *t-links* connecting non-terminal nodes to the terminal nodes, and *n-links* connecting two non-terminal nodes.

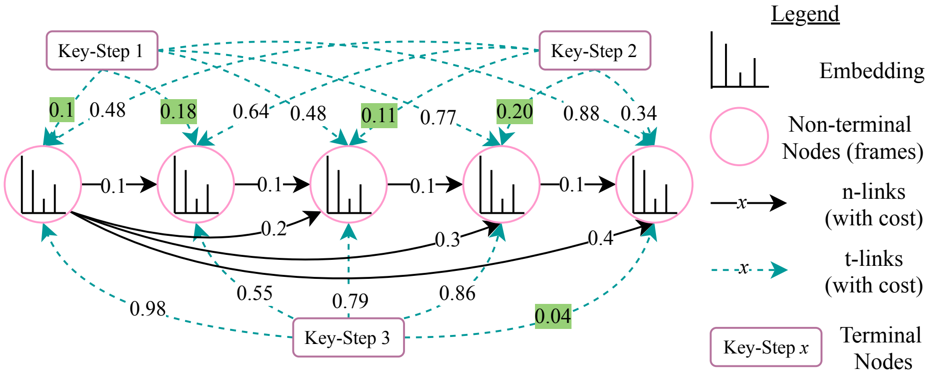

Fig. 4. ProCut Module (PCM). Non-terminal nodes in the graph represent the embeddings of the frames. Terminal nodes represent the key-steps required to perform the task. The terminal and non-terminal nodes are connected using the t-links. Non-terminal nodes are connected using the n-links. The numbers inscribed in arrows represent the cost of using the respective link. Costs highlighted in green represent the lowest cost to assign a frame to the key-step. For brevity, n-links are shown only for the first non-terminal node. Diagram best viewed in colour.

We use the Fuzzy C-Means algorithm [15] to assign a cost to the *t-links*. The algorithm performs soft clustering and calculates the probability of a frame belonging to each cluster. We subtract the probability value from 1 to obtain the cost of assigning a frame to each cluster. The cost value for the *n-links* is assigned based on the temporal distance between the nodes. For example, if the nodes are temporally closer (*e.g.*, nodes at positions 1 and 2 in Figure 4), the cost of assigning the same label to them is lower, otherwise (*e.g.*, for nodes at positions 1 and 5 in Figure 4), the cost is high. After creating the graph G, we use α-Expansion [4] to find the minimum cost cut. We use the discovered cut to assign frames to K labels. As shown in Figure 4, the lowest costs (highlighted in green) result in assigning the first and second frames to key-step 1, the third and fourth frames to key-step 2, and the last to key-step 3.

4.3 Determining Order of the Key-Steps

When it comes to determining the ordering of the key-steps, it makes sense to allow each video to have a distinct key-step ordering as there can be multiple ways to perform a task. However, current works either use a fixed order of key-steps to decode all the videos [19, 40, 70] or do not predict the ordering [18, 63]. One of the advantages of using CnC to determine the key-step is that it allows each video to have its independent order of the key-steps.

To infer the sequential order of key-steps, we calculate the normalized time for each frame v_i^n in video V_i consisting of p frames as $T(v_i^n) = \frac{n}{p}$ [40, 70]. Then we calculate the time instant for each cluster as the average normalized for frames assigned to it. The clusters are then arranged in increasing order of the average time, providing us with the sequence of key-steps used to perform the task in a video. Once we have key-step order for all the videos of the same task, we generate their ranked list based on the number of times the subjects followed a particular order. The order followed the most ends up being at the top of the ranked list. Doing this enables us to determine different sequential orders of key-steps to accomplish a task.

4.4 Implementation Details

We use ResNet-50 [28] as our backbone network to extract the features. Motivated by [16], for training the Embedder network, we use a pair of training videos at a time, select frames at random within the videos and optimize the proposed TC3I loss until convergence. The features are extracted from the *Conv4c* layer and a stack of c context frames features is created along the temporal dimension. We reshape our input video frames to 224×224. To aggregate the temporal information, we pass the combined features through two 3D convolutional layers followed by a 3D global max pooling layer, two fully-connected layers, and a linear projection layer to output the embeddings of dimension 128. We set the value of K to 7 and compare the performance of CnC with the other values of K in Table 6. Furthermore, for all our experiments, we follow the task-specific settings laid out in [18]. We use PyTorch [56] for all our experiments.

5 Experiments

5.1 Evaluation

Current works compute framewise F1-Score and IoU scores for key-step localization [18, 19, 40, 63, 70]. The F1-Score is a harmonic mean of precision and recall scores. For calculating recall, the ratio between the number of frames having correct key-steps prediction and the number of ground truth key-step frames across all the key-steps of a video is calculated. For precision, the denominator is the number of frames assigned to the key-steps. For calculations, the one-to-one mapping between the ground truth and prediction is generated using the Hungarian algorithm [39] following [2, 18, 19, 40, 63]. However, these metrics tend to assign high scores to models that assign most frames to a single cluster, as the key-step with most frames matches with the background frame's label in the ground truth. Furthermore, for untrimmed procedure learning videos, most of the frames are background, resulting in high scores.

Shen *et al.* [63] attempt to solve this problem by analyzing the MoF score, but as pointed out in [40], MoF is not always suitable for an imbalanced dataset. Instead, we propose calculating the framewise scores for each key-step separately and then taking the mean of the scores over all the key-steps. This penalises the cases when there is a large performance difference for different key-step, *e.g.*, when all the frames get assigned to a single key-step. Upon following this protocol, the scores for all the methods decrease. This paper presents the results generated using the proposed evaluation protocol unless otherwise mentioned.

Table 2. Procedure Learning from Third-person Videos. Comparison between state-of-the-art methods and CnC on benchmark third-person video datasets [19, 79]. Our method outperforms all the techniques using videos only (in F-Score). It even manages to give at par performance compared to the techniques using multi-modal input. **P**, **R**, and **F** represent precision, recall, and F-score respectively

	Input Modality	ProceL [19]			CrossTask [79]		
		P	**R**	**F**	**P**	**R**	**F**
Uniform	Video	12.4	9.4	10.3	8.7	9.8	9.0
Alayrc *et al.* [2]	Video + Narrations	12.3	3.7	5.5	6.8	3.4	4.5
Kukleva *et al.* [40]	Video	11.7	30.2	16.4	9.8	35.9	15.3
Elhamifar *et al.* [18]	Video	9.5	26.7	14.0	10.1	**41.6**	16.3
Fried *et al.* [22]	Video	–	–	–	–	28.8	–
Shen *et al.* [63]	Video + Narrations	16.5	**31.8**	21.1	15.2	35.5	21.0
CnC (*ours*)	Video	**20.7**	22.6	**21.6**	**22.8**	22.5	**22.6**

5.2 Procedure Learning from Third-person Videos

To test the generalizability of CnC on third-person videos and to ensure a fair comparison with existing methods [2, 18, 22, 40, 63], we perform experiments on third-person procedure learning benchmark datasets: ProceL [19] and CrossTask [79]. We obtain the results of previous works from [63]. Note that here we use the evaluation protocol employed by the previous works [18, 19, 40, 63]. As seen in Table 2, CnC outperforms other methods (in terms of the F-Score) utilizing only videos as the input modality. Further, with only video as the input modality, CnC even manages to perform at par with multi-modal methods. Previous works have used different forms of self-supervision. For example, [18] use the pseudo-labels provided by subset selection and [40] utilize the relative time-stamps of video frames. Instead, the comparison in Table 2 shows that the signal provided by corresponding frames is superior for the task of procedure learning.

5.3 Procedure Learning Results from Egocentric Videos

Baselines: We consider three baseline methods:

1. **Random.** Here we predict the labels by randomly sampling predictions from a uniform distribution with K values representing K key-steps.
2. **TC3I + HC.** Instead of PCM, we use the K-Means algorithm and generate K clusters from the representation space.
3. **TC3I + SS.** Here, instead of PCM, we use subset selection for the key-step assignment. The algorithm takes in the frame's embeddings and M (hyper-parameter) latent states obtained using K-Means [49]. It then selects a subset S (of size K) of the states as key-steps and finds the frames' assignments. We use the greedy algorithm used in [18] to perform subset selection. Refer to the supplementary material for the hyper-parameter values.

Table 3. Procedure Learning Results obtained on EgoProceL. Here, CnC performs the best, highlighting the effectiveness of the TC3I loss and PCM

	EgoProceL											
	CMU-MMAC		EGTEA G.		MECCANO		EPIC-Tents		PC Assembly		PC Disas.	
	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU
Random	15.7	5.9	15.3	4.6	13.4	5.3	14.1	6.5	15.1	7.2	15.3	7.1
TC3I + HC	19.2	9.0	20.8	7.9	16.6	**8.0**	15.4	7.8	21.7	11.0	24.9	14.1
TC3I + SS	19.7	8.9	20.4	7.9	16.3	7.8	15.9	7.8	24.8	11.9	23.6	14.0
CnC	**22.7**	**11.1**	**21.7**	**9.5**	**18.1**	7.8	**17.2**	**8.3**	**25.1**	**12.8**	**27.0**	**14.8**

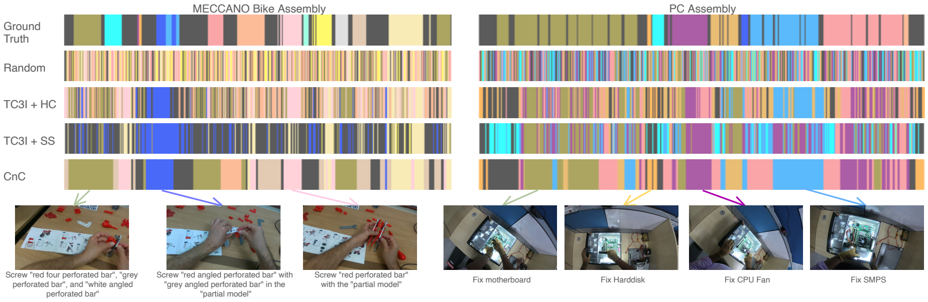

Fig. 5. Qualitative results for MECCANO and PC Assembly highlight the effectiveness of CnC. Additionally, PCM outperforms HC and SS when clustering the key-steps. Furthermore, due to the TC3I loss, CnC correctly identifies the key-steps that are short (fix a hard disk in PC Assembly). The gray segments denote the background.

Table 3 summarises the results obtained on EgoProceL using the baselines and proposed CnC. CnC performs higher than all the three baselines. This is due to (a) the ability of the TC3I loss to learn the representation space where similar key-steps lie close without enforcing any ordering or temporal constraints. Moreover, TC3I adds temporal coherency to the learned representations by adopting the C-IDM loss [27] (Figure 5). (b) PCM gains a comprehensive view of the problem by considering the cost of assigning each frame belonging to every key-step and its temporal relationship with the other frames. CnC performs better on long sequences as the TC3I loss compensates by searching for corresponding frames in the entire length of the videos, making it possible to learn a reasonable representation space despite the length of the videos. Further, the results in Table 3 show that PCM is superior for key-frame clustering and assignment along with TC3I as it results in the highest F-Score and IoU on EgoProceL. The gain in performance is because PCM considers the cost of assigning each frame to every key-step and its temporal relationship with the other frames (Figure 5). This allows PCM to gain a comprehensive view of the problem compared to HC, which does not consider the cost of each frame belonging to other key-steps and SS, which has lower generalisation power [18].

5.4 Egocentric vs. Third-person Videos

Here, we compare the results obtained after training CnC on multiple views from CMU-MMAC [11]. As seen in Table 4, the frame-wise F1-Score and IoU scores are the highest for the egocentric view. This is because egocentric videos offer lower occlusion by the expert's body and provide higher visibility of hand-object interactions. This highlights one of the central hypotheses of this paper: the effectiveness of using egocentric videos over third-person videos for procedure learning. Also, we observe that the results vary for third-person videos due to the camera placement. This increases one variable when creating data for procedure learning. Alternatively, egocentric videos use head-mounted cameras, eliminating uncertainty.

Table 4. Egocentric vs. Third-person results. We use different views from [11] for comparison. We obtain better results using CnC on egocentric videos highlighting their effectiveness. **P**, **R**, and **F** denote precision, recall, and F-score respectively

View	P	R	F	IoU
TP (Top)	17.4	18.4	17.9	8.1
TP (Back)	18.8	21.5	20.0	8.5
TP (LHS)	21.2	22.7	21.8	9.7
TP (RHS)	19.8	21.7	20.6	8.7
Egocentric	**21.6**	**24.4**	**22.7**	**11.1**

5.5 Ablation study

Here, we quantitatively evaluate our design choices. Due to space constraints, results for [11] and [47] are provided here, and the rest are in the supplementary.

Table 5. Effectiveness of the TC3I loss. TC3I loss outperforms other losses as it focuses on corresponding frames and employs C-IDM for temporal coherency

Experiment	CMU-MMAC [11]			EGTEA Gaze+ [47]		
	Precision	F-Score	IoU	Precision	F-Score	IoU
TCC + PCM	18.5	19.7	9.5	17.5	19.7	8.8
LAV + TCC + PCM	18.8	19.7	9.0	16.4	18.6	7.5
LAV + PCM	20.6	21.1	9.4	17.4	19.1	8.0
TC3I + PCM (CnC)	**21.6**	**22.7**	**11.1**	**19.6**	**21.7**	**9.5**

Effectiveness of the TC3I Loss: Here, we replace the TC3I loss in CnC with TCC [16], LAV [27], and a combination of LAV and TCC [27] to study the efficacy of the proposed TC3I loss. TC3I loss in Table 5 obtains the highest F-Score and IoU. As observed in our initial set of experiments, TCC loss lacks temporal coherency due to which temporally close frames do not lie close in the learned representation space, resulting in lower results when compared to TC3I and LAV, which account for temporal coherency using the C-IDM loss. For LAV + TCC, our observations are consistent with [27] because there is no performance gain when directly combining LAV and TCC losses since LAV works on L2-normalised embeddings, whereas TCC does not [27]. The LAV loss

Table 6. Selecting ***K*****.** Results with various values of K. Numbers in bold are highest in the respective row, and underlined numbers are highest in the respective column

Experiment	CMU-MMAC [11]				EGTEA Gaze+ [47]			
	K=7	K=10	K=12	K=15	K=7	K=10	K=12	K=15
Random	**15.7**	12.7	11.6	10.4	**15.4**	12.3	11.4	10.4
TC3I + HC	**19.2**	17.4	16.3	16.8	**20.8**	17.8	16.7	17.3
TC3I + SS	**19.7**	17.3	17.0	15.7	**20.4**	17.8	16.7	16.8
CnC	**22.7**	19.1	20.4	20.1	**21.7**	19.9	19.9	19.9

performs better than TCC and LAV + TCC; however, the results are not better than TC3I because the Soft-DTW used in LAV accounts for global alignment. However, LAV does not focus on the per-frame features [27], which is beneficial when looking for similar key-steps in different videos. The TC3I loss overcomes these issues by focusing on correspondences in multiple videos at frame level and adding temporal coherency by adopting the C-IDM loss.

Selecting the value of ***K*** Table 6 contains results of CnC and the baselines as the function of K. Additionally, it features the results after replacing PCM with HC and SS as the function of K. Here, key observations are: (a) CnC performs the best when $K = 7$, (b) the results do not change significantly for CnC as K increases. However, we observe a decline in the results for HC and SS as K increases, highlighting the effectiveness of PCM for key-step localisation.

6 Conclusion

Learning procedures from the visual demonstration of a task by an expert, is an important step in scaling the learning capabilities of autonomous agents. Unlike current state-of-the-art techniques, instead of third-person videos, we have proposed procedure learning from first-person viewpoint. Given the unavailability of the datasets for the purpose, we proposed the EgoProceL containing egocentric videos for procedure learning. We also proposed a new technique, CnC, for procedure learning from egocentric videos that utilize the proposed TC3I loss to learn an embedding space in a self-supervised fashion. Finally, we employ PCM to identify the key-steps. Our results demonstrate the superiority of using the egocentric view and the effectiveness of the proposed technique for procedure learning.

Acknowledgements. The work was supported in part by the Department of Science and Technology, Government of India, under DST/ICPS/Data-Science project ID T-138. We acknowledge Pravin Nagar and Sagar Verma for sharing the PC Assembly and Disassembly videos recorded at IIIT Delhi. We also acknowledge Jehlum Vitasta Pandit and Astha Bansal for their help with annotating a portion of EgoProceL.

References

1. Ahsan, U., Sun, C., Essa, I.: DiscrimNet: Semi-Supervised Action Recognition from Videos using Generative Adversarial Networks. In: Computer Vision and Pattern Recognition Workshops (CVPRW) 'Women in Computer Vision (WiCV)' (2018) 4
2. Alayrac, J.B., Bojanowski, P., Agrawal, N., Laptev, I., Sivic, J., Lacoste-Julien, S.: Unsupervised learning from Narrated Instruction Videos. In: Computer Vision and Pattern Recognition (CVPR) (2016) 2, 4, 5, 10, 11
3. Bojanowski, P., Lajugie, R., Bach, F., Laptev, I., Ponce, J., Schmid, C., Sivic, J.: Weakly Supervised Action Labeling in Videos under Ordering Constraints. In: European Conference on Computer Vision (ECCV) (2014) 3
4. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence (2001) 7, 9
5. Carlucci, F.M., D'Innocente, A., Bucci, S., Caputo, B., Tommasi, T.: Domain Generalization by Solving Jigsaw Puzzles. In: Computer Vision and Pattern Recognition (CVPR) (2019) 4
6. Carreira, J., Zisserman, A.: Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In: Computer Vision and Pattern Recognition (CVPR) (2017) 4
7. Chang, C.Y., Huang, D.A., Sui, Y., Fei-Fei, L., Niebles, J.C.: D3TW: Discriminative Differentiable Dynamic Time Warping for Weakly Supervised Action Alignment and Segmentation. In: Computer Vision and Pattern Recognition (CVPR) (2019) 3
8. Conners, R.W., Harlow, C.A.: A Theoretical Comparison of Texture Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence (1980) 8
9. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., Wray, M.: Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. In: European Conference on Computer Vision (ECCV) (2018) 2, 6
10. Damen, D., Leelasawassuk, T., Haines, O., Calway, A., Mayol-Cuevas, W.: You-Do, I-Learn: Discovering Task Relevant Objects and their Modes of Interaction from Multi-User Egocentric Video. In: British Machine Vision Conference (BMVC) (2014) 4
11. De La Torre, F., Hodgins, J., Bargteil, A., Martin, X., Macey, J., Collado, A., Beltran, P.: Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) database. In: Robotics Institute (2008) 2, 5, 13, 14
12. Diba, A., Sharma, V., Gool, L., Stiefelhagen, R.: DynamoNet: Dynamic Action and Motion Network. In: International Conference on Computer Vision (ICCV) (2019) 4
13. Ding, L., Xu, C.: Weakly-Supervised Action Segmentation with Iterative Soft Boundary Assignment. In: Computer Vision and Pattern Recognition (CVPR) (2018) 3
14. Doughty, H., Laptev, I., Mayol-Cuevas, W., Damen, D.: Action Modifiers: Learning From Adverbs in Instructional Videos. In: Computer Vision and Pattern Recognition (CVPR) (2020) 4
15. Dunn, J.C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. Journal of Cybernetics (1973) 9
16. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: Temporal Cycle-Consistency Learning. In: Computer Vision and Pattern Recognition (CVPR) (2019) 4, 6, 7, 10, 13

17. ELAN (Version 6.0) [Computer software]. (2020).: Nijmegen: Max Planck Institute for Psycholinguistics, The Language Archive. Retrieved from https://archive.mpi.nl/tla/elan, https://archive.mpi.nl/tla/elan 5
18. Elhamifar, E., Huynh, D.: Self-supervised Multi-task Procedure Learning from Instructional Videos. In: European Conference on Computer Vision (ECCV) (2020) 1, 3, 4, 5, 9, 10, 11, 12
19. Elhamifar, E., Naing, Z.: Unsupervised Procedure Learning via Joint Dynamic Summarization. In: International Conference on Computer Vision (ICCV) (2019) 1, 2, 3, 4, 9, 10, 11
20. Feng, Z., Xu, C., Tao, D.: Self-Supervised Representation Learning by Rotation Feature Decoupling. In: Computer Vision and Pattern Recognition (CVPR) (2019) 4
21. Fernando, B., Bilen, H., Gavves, E., Gould, S.: Self-Supervised Video Representation Learning with Odd-One-Out Networks. In: Computer Vision and Pattern Recognition (CVPR) (2017) 4
22. Fried, D., Alayrac, J.B., Blunsom, P., Dyer, C., Clark, S., Nematzadeh, A.: Learning to Segment Actions from Observation and Narration. In: Association for Computational Linguistics (ACL) (2020) 4, 11
23. Furnari, A., Farinella, G.: Rolling-Unrolling LSTMs for Action Anticipation from First-Person Video. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020) 2
24. Grauman, K., et al.: Ego4D: Around the World in 3,000 Hours of Egocentric Video. In: Computer Vision and Pattern Recognition (CVPR) (2022) 2
25. Greig, D., Porteous, B., Seheult, A.: Exact Maximum A Posteriori Estimation for Binary Images. Journal of the Royal Statistical Society Series B-Methodology (1989) 8
26. Han, T., Xie, W., Zisserman, A.: Video Representation Learning by Dense Predictive Coding. In: Workshop on Large Scale Holistic Video Understanding, ICCV (2019) 4
27. Haresh, S., Kumar, S., Coskun, H., Syed, S.N., Konin, A., Zia, Z., Tran, Q.H.: Learning by Aligning Videos in Time. In: Computer Vision and Pattern Recognition (CVPR) (2021) 4, 6, 8, 12, 13, 14
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Computer Vision and Pattern Recognition (CVPR) (2016) 4, 10
29. Hinton, G.E., Zemel, R.S.: Autoencoders, Minimum Description Length and Helmholtz Free Energy. In: Neural Information Processing Systems (1993) 4
30. Huang, D.A., Fei-Fei, L., Niebles, J.C.: Connectionist Temporal Modeling for Weakly Supervised Action Labeling. In: European Conference on Computer Vision (ECCV) (2016) 3
31. Huang, Y., Cai, M., Li, Z., Sato, Y.: Predicting gaze in egocentric video by learning task-dependent attention transition. In: European Conference on Computer Vision (ECCV) (2018) 2
32. Jang, Y., Sullivan, B., Ludwig, C., Gilchrist, I., Damen, D., Mayol-Cuevas, W.: EPIC-Tent: An Egocentric Video Dataset for Camping Tent Assembly. In: International Conference on Computer Vision (ICCV) Workshops (2019) 2, 5
33. Ji, L., Wu, C., Zhou, D., Yan, K., Cui, E., Chen, X., Duan, N.: Learning Temporal Video Procedure Segmentation From an Automatically Collected Large Dataset. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2022) 2

34. Jin-woo Choi and Gaurav Sharma and S. Schulter and Jia-Bin Huang: Shuffle and Attend: Video Domain Adaptation. In: European Conference on Computer Vision (ECCV) (2020) 4
35. Kim, D., Cho, D., Kweon, I.S.: Self-Supervised Video Representation Learning with Space-Time Cubic Puzzles. In: AAAI Conference on Artificial Intelligence (2019) 4
36. Kim, D., Cho, D., Yoo, D., Kweon, I.S.: Learning Image Representations by Completing Damaged Jigsaw Puzzles. In: Winter Conference on Applications of Computer Vision (WACV) (2018) 4
37. Komodakis, N., Gidaris, S.: Unsupervised representation learning by predicting image rotations. In: International Conference on Learning Representations (ICLR) (2018) 4
38. Kuehne, H., Arslan, A.B., Serre, T.: The language of actions: Recovering the syntax and semantics of goal-directed human activities. In: Computer Vision and Pattern Recognition (CVPR) (2016) 2, 5
39. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly (1955) 10
40. Kukleva, A., Kuehne, H., Sener, F., Gall, J.: Unsupervised learning of action classes with continuous temporal embedding. In: Computer Vision and Pattern Recognition (CVPR) (2019) 3, 4, 8, 9, 10, 11
41. Larsson, G., Maire, M., Shakhnarovich, G.: Colorization as a Proxy Task for Visual Understanding. In: Computer Vision and Pattern Recognition (CVPR) (2017) 4
42. Larsson, G., Maire, M., Shakhnarovich, G.: Learning Representations for Automatic Colorization. In: European Conference on Computer Vision (ECCV) (2016) 4
43. Lee, H.Y., Huang, J.B., Singh, M.K., Yang, M.H.: Unsupervised Representation Learning by Sorting Sequences. In: International Conference on Computer Vision (ICCV) (2017) 4
44. Li, J., Lei, P., Todorovic, S.: Weakly Supervised Energy-Based Learning for Action Segmentation. In: International Conference on Computer Vision (ICCV) (2019) 3
45. Li, J., Todorovic, S.: Set-Constrained Viterbi for Set-Supervised Action Segmentation. In: Computer Vision and Pattern Recognition (CVPR) (2020) 3
46. Li, Y., Fathi, A., Rehg, J.M.: Learning to Predict Gaze in Egocentric Video. In: International Conference on Computer Vision (ICCV) (2013) 2
47. Li, Y., Liu, M., Rehg, J.M.: In the Eye of Beholder: Joint Learning of Gaze and Actions in First Person Video. In: European Conference on Computer Vision (ECCV) (2018) 2, 5, 6, 13, 14
48. Liu, X., van de Weijer, J., Bagdanov, A.D.: Leveraging Unlabeled Data for Crowd Counting by Learning to Rank. In: Computer Vision and Pattern Recognition (CVPR) (2018) 4
49. Lloyd, S.: Least squares quantization in PCM. IEEE Transactions on Information Theory (1982) 8, 11
50. Malmaud, J., Huang, J., Rathod, V., Johnston, N., Rabinovich, A., Murphy, K.: What's Cookin'? Interpreting Cooking Videos using Text, Speech and Vision. In: HLT-NAACL (2015) 4
51. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In: International Conference on Computer Vision (ICCV) (2019) 2

52. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and Learn: Unsupervised Learning Using Temporal Order Verification. In: European Conference on Computer Vision (ECCV) (2016) 4
53. Naing, Z., Elhamifar, E.: Procedure Completion by Learning from Partial Summaries. In: British Machine Vision Conference (BMVC) (2020) 3
54. Ng, E., Xiang, D., Joo, H., Grauman, K.: You2Me: Inferring Body Pose in Egocentric Video via First and Second Person Interactions. In: Computer Vision and Pattern Recognition (CVPR) (2020) 2
55. Noroozi, M., Pirsiavash, H., Favaro, P.: Representation Learning by Learning to Count. In: International Conference on Computer Vision (ICCV) (2017) 4
56. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Neural Information Processing Systems (2019) 10
57. Pirsiavash, H., Ramanan, D.: Detecting activities of daily living in first-person camera views. In: Computer Vision and Pattern Recognition (CVPR) (2012) 2
58. Ragusa, F., Furnari, A., Livatino, S., Farinella, G.M.: The MECCANO Dataset: Understanding Human-Object Interactions From Egocentric Videos in an Industrial-Like Domain. In: Winter Conference on Applications of Computer Vision (WACV). pp. 1569–1578 (2021) 5
59. Richard, A., Kuehne, H., Gall, J.: Action Sets: Weakly Supervised Action Segmentation Without Ordering Constraints. In: Computer Vision and Pattern Recognition (CVPR) (2018) 3
60. Richard, A., Kuehne, H., Iqbal, A., Gall, J.: NeuralNetwork-Viterbi: A Framework for Weakly Supervised Video Learning. In: Computer Vision and Pattern Recognition (CVPR) (2018) 3
61. Sener, F., Yao, A.: Zero-Shot Anticipation for Instructional Activities. In: International Conference on Computer Vision (ICCV) (2019) 3
62. Sener, O., Zamir, A.R., Savarese, S., Saxena, A.: Unsupervised Semantic Parsing of Video Collections. In: International Conference on Computer Vision (ICCV) (2015) 4
63. Shen, Y., Wang, L., Elhamifar, E.: Learning To Segment Actions From Visual and Language Instructions via Differentiable Weak Sequence Alignment. In: Computer Vision and Pattern Recognition (CVPR) (2021) 1, 3, 4, 9, 10, 11
64. Sigurdsson, G.A., Gupta, A., Schmid, C., Farhadi, A., Alahari, K.: Actor and Observer: Joint Modeling of First and Third-Person Videos. In: Computer Vision and Pattern Recognition (CVPR) (2018) 2, 6
65. Singh, S., Arora, C., Jawahar, C.V.: First Person Action Recognition Using Deep Learned Descriptors. In: Computer Vision and Pattern Recognition (CVPR) (2016) 2
66. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised Learning of Video Representations Using LSTMs. In: International Conference on Machine Learning (ICML) (2015) 4
67. Tang, Y., Ding, D., Rao, Y., Zheng, Y., Zhang, D., Zhao, L., Lu, J., Zhou, J.: COIN: A Large-Scale Dataset for Comprehensive Instructional Video Analysis. In: Computer Vision and Pattern Recognition (CVPR) (2019) 2
68. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: Learning Spatiotemporal Features with 3D Convolutional Networks. In: International Conference on Computer Vision (ICCV) (2015) 4

69. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A Closer Look at Spatiotemporal Convolutions for Action Recognition. In: Computer Vision and Pattern Recognition (CVPR) (2018) 4
70. VidalMata, R.G., Scheirer, W.J., Kukleva, A., Cox, D., Kuehne, H.: Joint Visual-Temporal Embedding for Unsupervised Learning of Actions in Untrimmed Sequences. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2021) 3, 4, 9, 10
71. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and Composing Robust Features with Denoising Autoencoders. In: International Conference on Machine Learning (ICML) (2008) 4
72. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating Videos with Scene Dynamics. In: Neural Information Processing Systems (2016) 4
73. Wang, X., Girshick, R.B., Gupta, A., He, K.: Non-local Neural Networks. In: Computer Vision and Pattern Recognition (CVPR) (2018) 4
74. Wei, D., Lim, o., Zisserman, A., Freeman, W.T.: Learning and Using the Arrow of Time. In: Computer Vision and Pattern Recognition (CVPR) (2018) 4
75. Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., Zhuang, Y.: Self-Supervised Spatiotemporal Learning via Video Clip Order Prediction. In: Computer Vision and Pattern Recognition (CVPR) (2019) 4
76. Yu, S.I., Jiang, L., Hauptmann, A.: Instructional Videos for Unsupervised Harvesting and Learning of Action Examples. In: ACM International Conference on Multimedia (2014) 4
77. Zhou, L., Xu, C., Corso, J.J.: Towards Automatic Learning of Procedures From Web Instructional Videos. In: AAAI Conference on Artificial Intelligence (2018) 2, 3
78. Zhukov, D., Alayrac, J.B., Laptev, I., Sivic, J.: Learning Actionness via Long-range Temporal Order Verification. In: European Conference on Computer Vision (ECCV) (2020) 4
79. Zhukov, D., Alayrac, J.B., Cinbis, R.G., Fouhey, D., Laptev, I., Sivic, J.: Cross-task weakly supervised learning from instructional videos. In: Computer Vision and Pattern Recognition (CVPR) (2019) 2, 3, 5, 11