Unsupervised Learning of Deep Feature Representation for Clustering Egocentric Actions

by

Bharat Lal Bhatnagar, Suriya Singh, Chetan Arora, C V Jawahar

in

International Joint Conference on Artificial Intelligence (IJCAI 2017)
(*IJCAI-2017*)

Melbourne, Australia

Report No: IIIT/TR/2017/-1

Centre for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
August 2017

Unsupervised Learning of Deep Feature Representation for Clustering Egocentric Actions

Bharat Lal Bhatnagar* , Suriya Singh* , Chetan Arora+ , C.V. Jawahar*
CVIT, KCIS, International Institute of Information Technology, Hyderabad*
Indraprastha Institute of Information Technology, Delhi+

Abstract

Popularity of wearable cameras in life logging, law enforcement, assistive vision and other similar applications is leading to explosion in generation of egocentric video content. First person action recognition is an important aspect of automatic analysis of such videos. Annotating such videos is hard, not only because of obvious scalability constraints, but also because of privacy issues often associated with egocentric videos. This motivates the use of unsupervised methods for egocentric video analysis. In this work, we propose a robust and generic unsupervised approach for first person action clustering. Unlike the contemporary approaches, our technique is neither limited to any particular class of action nor requires priors such as pre-training, fine-tuning, etc. We learn time sequenced visual and flow features from an array of weak feature extractors based on convolutional and LSTM autoencoder networks. We demonstrate that clustering of such features leads to the discovery of semantically meaningful actions present in the video. We validate our approach on four disparate public egocentric actions datasets amounting to approximately 50 hours of videos. We show that our approach surpasses the supervised state of the art accuracies without using the action labels.

1 Introduction

Use of egocentric cameras such as GoPro, Google Glass, Microsoft SenseCam has been on the rise since the start of this decade. The benefits of first person view in video understanding have generated interest of computer vision researchers and application designers alike.

Given the wearable nature of egocentric cameras, the videos are typically captured in an always-on mode, thus generating long and boring day-logs of the wearer. The cameras are typically worn on the head, where sharp head movement of the wearer introduce severe destabilization in the captured videos, making them difficult to watch. Change in perspective as well as unconstrained motion of the camera also makes it difficult to apply traditional computer vision algorithms for egocentric video analysis.

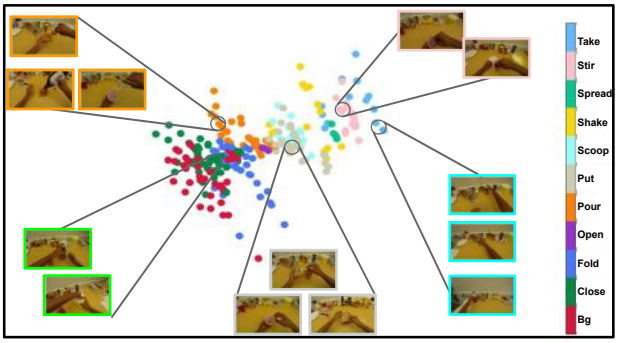

Figure 1: In this paper we present an approach for unsupervised learning of wearer's actions which, in contrast to the existing techniques, is generic and applicable to various first person scenarios. We learn action specific features and segment the video into semantically meaningful clusters. The figure shows 2D feature embedding using our proposed technique on GTEA dataset.

First person action recognition is an important first step for many egocentric video analysis applications. The task is different from classical third person action recognition because of unavailability of standard cues such as actor's pose. Most of the contemporary works on first person action recognition use hand-tuned or deep learnt features for specific action categories of interest, restricting their generalization and wider applicability. The features proposed for actions involving hand and object interactions (e.g., 'making coffee', 'using cell phone', etc.) [Fathi *et al.*, 2011a; Fathi *et al.*, 2012; Singh *et al.*, 2016b; Ma *et al.*, 2016] and for actions which do not involve such interactions (e.g., 'walking', 'sitting', etc.) [Ryoo and Matthies, 2013; Kitani *et al.*, 2011] are very different from each other and applicable only for the actions of their interest. Similarly, the features proposed for long term actions (e.g., 'applying make-up', 'walking', etc.) [Poleg *et al.*, 2014; Poleg *et al.*, 2015] are unsuitable for short term actions (e.g., 'take object', 'jump', etc.), and vice versa. We propose an unsupervised feature learning approach in this paper, which can generalize to different categories of first person actions.

Many of the earlier works in first person action recognition, [Spriggs *et al.*, 2009; Pirsiavash and Ramanan, 2012; Ogaki *et al.*, 2012; Matsuo *et al.*, 2014; Singh *et al.*, 2016b; Singh *et al.*, 2016a], have focused on supervised techniques. Scarcity of manually annotated examples is a natural restric-

tion for supervised approaches. Getting annotated examples is even harder for egocentric videos due to difficulty in watching such videos as well as the accompanying privacy issues. These reasons limit the potential of data driven supervised learning based approaches for egocentric video analysis. Unsupervised action recognition appears to be the natural solution to many of the aforementioned issues. While most of the earlier approaches [Singh *et al.*, 2016b; Singh *et al.*, 2016a; Kitani *et al.*, 2011; Pirsiavash and Ramanan, 2012; Lu and Grauman, 2013] can be seen as variants of bag of words model, we use convolutional and LSTM autoencoder network based unsupervised training which can seamlessly generalize to all style of actions.

Contributions: The specific contributions of the work are as follows:

1. We propose a generic unsupervised feature learning approach for first person action clustering using weak learners based on LSTM autoencoders. The weak learners are able to capture temporal patterns at various temporal resolutions. Unlike state of the art, our approach can generalize to various first person action categories viz., short term, long term, actions involving hand object interactions and actions without such interactions.
2. The action categories discovered using our approach are semantically meaningful and similar to the ones used in supervised techniques (see Figure 1). We validate our claims through extensive experiments on four disparate public egocentric action datasets viz., GTEA , ADL-short, ADL-long, and HUJIEGOSEG. We improve the action recognition accuracy obtained by state of the art supervised techniques on all the egocentric datasets.
3. Though not the focus of this paper, we show that the proposed approach can be effectively applied to clustering third person actions as well. We validate the claim by experiments on 50 SALAD dataset.
4. The proposed approach can capture action dynamics at various temporal resolutions. We show that using our clustering approach at various hierarchical levels yields meaningful labels at different semantic granularity.

2 Related Work

Most of the techniques for first person action recognition can be broken down into two categories: supervised and unsupervised.

Supervised Techniques In hand object interaction setting, Fathi *et al.*[Fathi *et al.*, 2011a] have focused on short term actions and have used cues such as optical flow, pose, size and location of hands in their feature vector. In a similar setting, Pirsiavash and Ramanan [Pirsiavash and Ramanan, 2012] propose to recognise activities of daily life by detecting salient objects. Later, Li *et al.*[Li *et al.*, 2015] and Singh *et al.*[Singh *et al.*, 2016a] have adapted popular trajectories features [Wang *et al.*, 2011; Wang and Schmid, 2013], from the third person action recognition literature. They recognize first person short term hand object interactions by incorporating the egocentric cues such as head motion, gaze and salient regions. Recently, Singh *et al.*[Singh *et al.*, 2016b] have proposed a three-stream CNN architecture for short term actions involving hand object interactions. Ma *et al.*[Ma *et al.*, 2016] propose a similar multi-stream deep architecture for joint action, activity and object recognition in egocentric videos.

In a different first person action setting which does not involve hand object interaction, Singh *et al.*[Singh *et al.*, 2016a] show that trajectory features can also be applied for such actions. Poleg *et al.*[Poleg *et al.*, 2014] have focussed on recognising long term actions of the wearer using motion cues of the camera wearer. Later, Poleg *et al.*[Poleg *et al.*, 2015] proposed to learn a compact 3D CNN network with flow volume as input for long term action recognition.

Unsupervised Techniques In a third person action setting, Niebles *et al.*[Niebles *et al.*, 2008] proposed to learn categories of human actions using spatio-temporal words in temporally trimmed videos. In a similar scenario, Jones *et al.*[Jones and Shao, 2014] use contextual information related to an action, such as the scene or the objects, for unsupervised human action clustering. Unsupervised techniques have also been deployed to learn important people and objects for video summarization tasks [Lee *et al.*, 2012; Lu and Grauman, 2013] and identify the significant events for extraction of video highlights [Yang *et al.*, 2015] from egocentric videos. Autoencoders have been popularly applied for unsupervised pre-training of deep networks [Srivastava *et al.*, 2015; Erhan *et al.*, 2010] as well as for learning feature representation in an unsupervised way [Vincent *et al.*, 2010]. Recently, LSTM autoencoders have been used to learn video representations for the task of unsupervised extraction of video highlights [Yang *et al.*, 2015].

The work closest to us is by Kitani *et al.*[Kitani *et al.*, 2011] who have proposed an unsupervised approach to discover short term first person actions. They use hand-crafted global motion features to cluster the video segments. However, the features are highly sensitive to camera motion and often result in over-segmentation or discovering classes that are not semantically meaningful.

3 Proposed Approach

Our focus is on learning representation of an egocentric video for the purpose of clustering first person actions. We follow a two stage approach in which we first learn the frame level representation from an array of convolutional autoencoder networks, followed by multiple LSTM autoencoder networks to capture the temporal information (see Figure 2).

3.1 Learning Frame Level Feature Representation

Instead of using a conventional approach of training a single large autoencoder, we use multiple smaller autoencoders, each learning a different representation. Our approach is inspired from boosting technique in machine learning, where ensemble of weak learners are able to outperform a single more complicated classifier. Additionally, training a large autoencoder may require large amount of training samples.

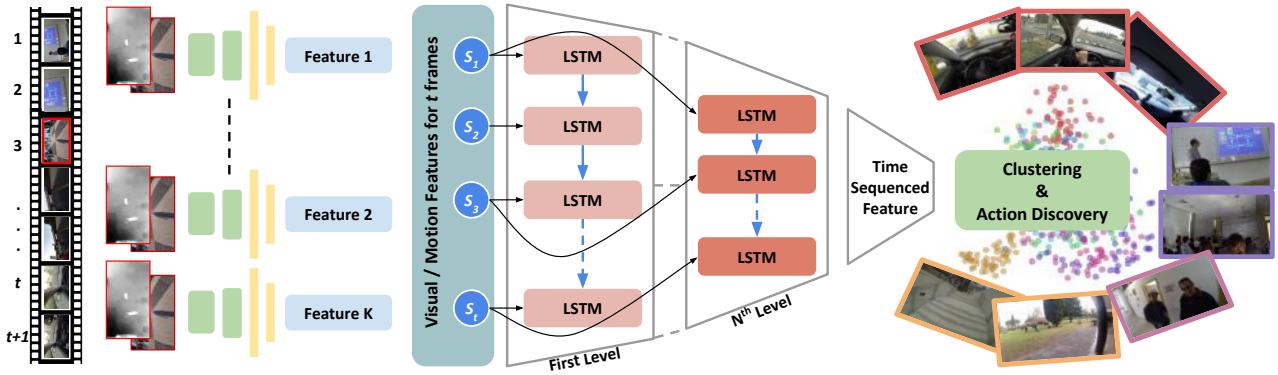

Figure 2: Our approach for unsupervised egocentric action clustering and discovery using time sequenced flow and visual features. We use K stacked autoencoder networks to learn frame level representations from input video splices. A splice is a video segment of consecutive t frames. We further learn the temporal representation by using pooled frame level representations as input to the LSTM networks. Each LSTM network captures information at varying temporal resolutions, forming a temporal pyramid. We assume a complete unsupervised setting for feature learning where only the video is given. Clustering these features yields semantically meaningful action categories.

Multiple small autoencoders can be trained relatively easily with small resource requirements.

We use dense optical flow (for motion) and raw frames (for appearance) as inputs to train the autoencoders. We train separate autoencoders for each type of input to avoid confusing the network and do a late fusion of the learned features to get the final embedding. We note that, apart from using optical flow and raw frame, other type of egocentric cues, such as hands and saliency maps used in [Singh *et al.*, 2016b; Li *et al.*, 2015; Ma *et al.*, 2016], could have also been used in our pipeline. However, using flow and frame as inputs keeps the model generalizable to different kinds of first person actions.

We divide an input sequence into multiple sub-sequences, referred to as a *splice*, of a fixed temporal window length t. The choice of window length is based on the actions to be processed. We have used a 2 second window for short term actions and 10 seconds for long term actions. Note that the splice (or temporal window) may or may not be overlapping, depending on the availability of data. Overlapping temporal windows allows to generate more, but potentially very similar, training samples from a video.

We divide each video into K sets of splices, each containing same number of splices. However, this implies that, depending upon the video length, the size of a set in different videos can be different. Let $N = \{N_1, N_2, \ldots, N_K\}$, be the set of all the sets of splices thus generated. During the train time, we randomly assign a set N_k of splices to each of the K autoencoders. The underlying idea in doing so is that each autoencoder would learn features specific to the subset it received. Let v_k^i be a video splice in the set $k \in K$. We train our model by minimizing the following objective in an unsupervised fashion.

$$\min_{\theta_k} \sum_{i=1}^{|k|} (\mathcal{Y}_k(v_k^i, \theta_k) - v_k^i)^2 \quad \forall k \in K$$
$$\mathcal{Y}_k(v, \theta_k) = \mathcal{Y}^d(\mathcal{Y}^e(v, \theta_k^e), \theta_k^d)$$

where $\mathcal{Y}_k(v, \theta_k)$ represents the output of stacked convolutional autoencoder with learned network parameters $\theta_k = \{\theta_k^e, \theta_k^d\}$, where e and d denote encoder and decoder respectively. Note that, though the training set for an autoencoder is a set of splices, the actual input to the network is a single frame (RGB or flow). Managing the training samples at the splice level (set of t consecutive frames) ensures that the temporal adjacent frames goes to the same learner, allowing them exploit and learn similarity between consecutive frames due to the similar action being performed (ignoring the noise due to splices at the action boundary). We observe this empirically also, where the loss for the weak learners is much higher if the samples are chosen independently at the frame level (with consecutive frames going to different weak learners).

At the test time, we pass each frame, j, of a splice i, through each of the learnt autoencoder, k, and create a feature vector for every frame in the following way.

$$\mathcal{Y}_k^{(i,j)} = \mathcal{Y}_k^e(v^{(i,j)}, \theta_k^e)$$
$$f_i^j = \langle \mathcal{Y}_1^{(i,j)}, \mathcal{Y}_2^{(i,j)}, \ldots, \mathcal{Y}_K^{(i,j)} \rangle.$$

Here $\langle \ldots \rangle$ denotes concatenation operation.

It is possible that some of the autoencoders end up learning similar features, but we do not try to control it specifically. Figure 3 shows visualisation of the filter responses from our trained autoencoders. The responses indicate that our network is able to capture both global features such as camera motion as well as localized features such as hands and objects.

3.2 Learning Temporal Features

Most of the contemporary works [Singh *et al.*, 2016a; Kitani *et al.*, 2011; Pirsiavash and Ramanan, 2012; Lu and Grauman, 2013; Wang *et al.*, 2011; Wang and Schmid, 2013] on first person action recognition adapts a variant of bag of words approach, which most importantly ignores the sequential nature of the action data. We believe, the sequence information is crucial and use LSTM autoencoder networks to learn temporal patterns in the proposed model.

Figure 3: Filter responses from frame level learners. The left image is a filter response corresponding to RGB stream, whereas right image corresponds to optical flow stream. Response looks interpretable (detecting hands and its movement), despite individual network being modelled as a weak learner.

We use set of frame level features, f_i^j, corresponding to splice j, extracted from the first stage to train the LSTM autoencoder networks. Similar to the first stage, here too, we minimize the mean-squared-error between the input and its reconstruction. Once trained, this LSTM network provides a time sequenced embedding based on both the frame level features and the order in which they appear. Formally, the LSTM network minimizes the following objective:

$$\min_{\phi} \sum_{i=1}^{n} \sum_{j=1}^{t} (\mathcal{R}(f_i^j, \phi) - f_i^j)^2$$

$$\mathcal{R}(f, \phi) = \mathcal{R}^d(\mathcal{R}^e(f, \phi^e), \phi^d)$$

where, t is the length of the splice, n is the number of splices in the train set, and $\mathcal{R}(f, \phi)$ represents LSTM autoencoder with learned network parameters $\phi = \{\phi^e, \phi^d\}$. At test time, we use only encoder network to get a representation $s_i = \langle R^e(f_i^1, \phi^e), R^e(f_i^1, \phi^e), \ldots, R^e(f_i^t, \phi^e) \rangle$, for the splice i.

We further learn time sequenced feature at different temporal resolutions (see Figure 2) similar to temporal pyramids used in [Singh *et al.*, 2016a; Pirsiavash and Ramanan, 2012; Ryoo *et al.*, 2015]. Each LSTM autoencoder learns time sequenced feature at fixed but different temporal resolution and offset. The feature representation from each LSTM encoder network is concatenated yielding a coarse to fine temporal pyramid representation. Such features adjust the temporal resolution for activities being performed at various speeds by different subjects.

In the end, we use the K-Means clustering to cluster the video splices on the basis of features obtained from the LSTM autoencoder.

3.3 Architecture Details

The input to the network is a gray-scale raw frame and dense optical flow down-sampled to a resolution of 64×36. Prior to the training, we normalize the input data to be in the range $[-1, 1]$. The encoder network consists of 2 convolutional layers and 2 fully connected layers, followed by the decoder network with 2 fully connected and 2 convolutional layers. We keep stride equal to 1 everywhere and use 2×2 max pooling. All the layers have a *tanh* activation function. We use the popular ADAM solver [Kingma and Ba, 2015] for the optimization. Unlike the large number of filters often used in deep networks for supervised classification, we use 8 and 18 convolutional filters, each of size 5×5 in the first two layers of the autoencoders. This reduces the number of parameters and keeps the training stage faster and simpler. This is also in consonance with our objective of learning multiple weak learners. The spatial representation from K spatial weak learner for each frame results in a $100 \times K$ dimensional vector. In our experiments we have kept $K = 20$.

The LSTM autoencoder is a simple two layer architecture. The first layer is a sequence-to-sequence LSTM that takes the feature of a splice and outputs a lower dimensional encoding to the decoder, which also is a sequence-to-sequence LSTM layer that reconstructs the input matrix but in a reversed order. We do the inversion similar to [Srivastava *et al.*, 2015], to avoid learning the trivial identity function. We use RMS-prop solver [Tieleman and Hinton, 2012] here. We learn a time sequenced coarse to fine feature representation with 3 levels of temporal pyramid. The ratios of temporal extent of each level are 0.5, 1 and 2 times of the splice size respectively.

Figure 4: Example of first person actions we propose to learn in this work. As can be seen, the actions vary from simple 'put', 'take' etc. in a controlled environment to slightly tricky actions like 'driving', 'walking' etc. in the wild. Our unsupervised approach discovers and detects these actions in untrimmed egocentric videos.

Dataset	Classes	Action Style	Supervised SoA	Ours
GTEA	11	S+O	0.68 [2016b]	**0.69**
ADL - short	21	S+O	0.37 [2016b]	**0.39**
ADL - long	12	L+O	N.A.	0.35
HUJIEGOSEG	7	L+X	0.89 [2015]	**0.90**

Table 1: Details of egocentric action datasets used for experimentation. **L/S :** Long or Short term action, **O/X :** actions involve object interaction or not. We compare with state of the art for supervised action recognition task. See the text on evaluation methodology for details.

4 Datasets and Evaluation

We have tested the proposed method on multiple datasets containing variety of egocentric action categories. We have used GTEA [Fathi *et al.*, 2011b] and ADL-short [Singh *et al.*, 2016a] for short term, hand-object coordinated videos, ADL-long [Pirsiavash and Ramanan, 2012] for long term hand-object coordinated videos and HUJIEGOSEG [Poleg *et al.*, 2014] for long term videos without the handled objects. Figure 4 illustrates some of the the action classes that we cluster using our approach. Though not the focus of this paper, we have also experimented in a third person setup,50 SALAD [Stein and McKenna, 2013], to establish

Sequence	Optical Flow	Frame	Flow+Frame
Cheese	0.57	**0.76**	**0.76**
CofHoney	0.59	0.70	**0.78**
Coffee	0.50	**0.72**	0.70
Hotdog	0.52	0.56	**0.63**
Pealate	0.44	**0.62**	**0.62**
Peanut	0.48	**0.73**	0.71
Tea	0.49	**0.66**	**0.66**
Average	0.50	0.67	**0.69**

Table 2: Accuracy of proposed approach using flow and appearance on GTEA dataset. The two input modalities provide complementary information and using both improves the results.

the generality of our approach. We evaluate the clustering performance using popular clustering assessment metrics, normalized mutual information (NMI) and homogeneity score as used in [Yang *et al.*, 2016].

We observe that the clusters from our approach corresponds to semantically meaningful actions and can be potentially compared with supervised action recognition approaches as well. However, we do not have a one to one mapping between the clusters and the labels. Similar to [Kitani *et al.*, 2011; Poleg *et al.*, 2015], we formulate the mapping problem as a bipartite matching or an assignment problem [Kuhn, 1955]. We have experimented with both greedy and Hungarian method [Kuhn, 1955] for the inference. The cost of assigning cluster i to class label j is computed as the F1 score weighted by population for class j when i is assigned to j. We fix the number of clusters as number of distinct actions from the ground truth for these experiments. We also investigate different model parameters such as input modes, cluster assignment algorithms, etc. With the proposed evaluation strategy, our model can perform non overlapping splice level prediction. For comparison with supervised frame level prediction techniques, we assign the label of the splice to each of the frame in it and then compute the accuracy. It may be noted that the splice boundary may not be aligned with action boundary and does not give any undue advantage to the proposed scheme.

Table 1 summarizes the details of various datasets and comparison with state of the art supervised action recognition techniques using the suggested evaluation methodology.

5 Experiments and Results

We verify the semantically meaningful nature of the clusters produced from the proposed method by performing 'cluster to class label' mapping as described earlier. This allows us to compute and compare the classification accuracy with the supervised methods. Table 1 shows the comparison with the state of the art techniques chosen on the basis of the best performance in terms of classification accuracy on the benchmark datasets. We improve the state of the art on all datasets without using any action labels. Figure 5 serves to indicate that, though we make no attempt to temporally regularize the cluster assignment, the adjacent splices do get similar feature representation and fall in the same cluster. Figure 6 shows example frames clustered correctly and incorrectly with our approach.

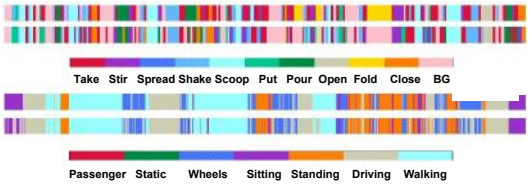

Figure 5: The top sub-figure contains error visualization for 11 activities of subject S2 from GTEA dataset. The bottom one contains a similar visualization for 7 long term actions from HUJIEGOSEG dataset. Each action label has been color coded. Top row in each sub-figure show splice level ground truth and bottom ones show temporal segmentation results using our approach.

Figure 6: Clustering examples across various activities using our approach on different datasets. First three columns shows correct cluster assignment while the last column shows places where our approach fails to assign correct clusters. First row: 'take' action, last example is labeled as 'BG' because wearer usually performs 'take' with left hand while it is from the right hand in this rare example. Second row: 'driving' action, last example is confused with 'sitting' when the car is being refueled. Third row: 'spread' action, last example is confused with 'put' probably due to occurrence of an action boundary.

Next we validate the choice of input. Table 2 compares clustering performance on GTEA dataset [Fathi *et al.*, 2011b] using appearance, motion and their joint feature representation. We note that for actions involving hand object interaction (e.g., stir, scoop etc.) where object appearance and hand shape are important cues, appearance based features perform better than motion based features. However, for actions without object interactions (e.g., walking, standing, driving etc.) where the object and scene appearance are irrelevant to actions, motion based feature performs better. Using joint representation of appearance and motion based features further improves the overall performance. We use joint representation for all further experiments.

In Table 3, we compare per class performance of our method against the supervised state of the art methods for both short term (GTEA) and long term (HUJIEGOSEG) actions. Results show that our method performs better on almost all first person actions. Figure. 7 shows that some of the most confusing pairs are 'Walking-Standing', 'Drive-Sit' in long term and 'Spread-Pour', 'Scoop-Stir' in short term actions. It is interesting to note that these happen to be confusing classes from human perspective as well. One serious limitation of our unsupervised approach over the supervised methods is in case of 'background' class. It may be noted that

Action	[P.1]	[P.2]	Us
Walking	0.83	**0.86**	**0.86**
Sitting	0.62	0.84	**0.95**
Standing	0.47	0.79	**0.86**
Static	0.97	**0.98**	0.94
Driving	0.74	**1**	0.88
Passenger	0.43	0.82	**0.86**
Wheels	0.86	N.A.	**0.92**

Action	[S.]	Us
Pour	0.95	**1.0**
Fold	0.0	0.0
Take	**0.80**	0.69
Stir	0.42	**0.82**
Spread	0.87	**0.90**
Shake	0.66	**0.83**
Scoop	0.59	**0.88**
Put	0.61	**0.67**
Open	0.65	**0.70**
Close	0.43	**0.62**
BG	**0.59**	0.40

Table 3: Comparing with state of the art on HUJIEGOSEG long term action (left) and GTEA short term action (right) datasets. Here [P.1], [P.2] and [S.] refer to [Poleg *et al.*, 2014], [Poleg *et al.*, 2015] and [Singh *et al.*, 2016b] respectively

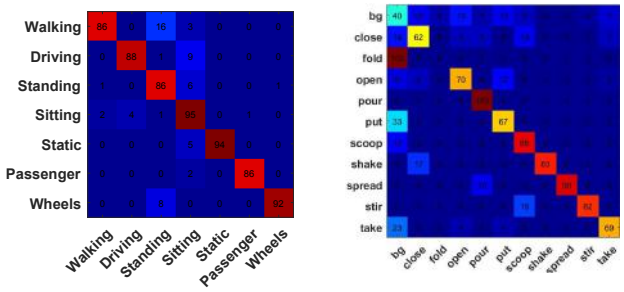

Figure 7: Right: confusion matrix showing clustering results on GTEA dataset. It is evident here that most errors are corresponding to 'BG' class. Please see the text for possible explanations. Notice that recall corresponding to other classes is very high. Left: confusion matrix for HUJIEGOSEG dataset. Here we see that sitting and driving are getting confused due to visual similarity of sitting with being inside a stationary car.

background class is comprised of all video segments where the wearer is not doing the other labeled action. This makes background class highly diverse making it difficult for the proposed network to learn a common representation for it.

We also investigate the influence of assignment criterion on the clustering performance. It can be seen in Table 4 that clustering results and quality are robust to assignment algorithm, indicating the presence of natural clusters. Note that our approach does not make any assumption regarding the nature of videos and thus can be applied to non egocentric videos as well. This experiment goes on to show the universality of our weak learner approach.

Number of clusters in k-Means clustering used by our approach is an important hyperparameter. We observe that setting different values for the hyperparameter allows us to cluster the video at different semantic granularity. Figure 8 shows the clusters obtained at finer levels when action is split into action+object and action+object+time by allowing different number of clusters in our approach. The F1 scores obtained for the three levels are 0.55, 0.50 and 0.46 respectively. Such an experiment was not possible for other datasets due to

Evaluation Criterion	Matching Criterion	
	Greedy	Hungarian
Accuracy	0.69	0.66
NMI	0.59	0.58
Homogenity	0.61	0.60
F1	0.66	0.78

Table 4: Comparing clustering performance using different matching algorithms. NMI score indicates cluster compactness and separation, Homogeneity indicates cluster purity and F1 score recall and precision.

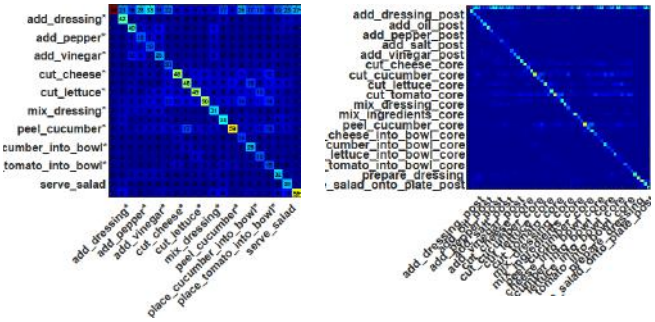

Figure 8: Confusion matrices obtained from our approach on 50 SALAD dataset below the base level granularity of actions. At base level the precision, recall and f1 of our approach are 0.54, 0.57, 0.55 as opposed to supervised approach [Stein and McKenna, 2016] 0.59, 0.58, 0.58 respectively. Here we show that more number of clusters using our features translate to semantically finer action labels. There are 10 basic action classes which we further split into 20 and 55 classes respectively. Left: with 20 clusters we roughly segment action+object. Sample labels here are cut-cucumber, cut-tomato, cut-lettuce etc. Right: with 55 clusters we get segmentation of the form action+object+time. Sample labels here are cut-cucumber-pre, cut-cucumber-core, cut-cucumber-post.

unavailability of hierarchical labeling in their corresponding ground truths. We also experiment with an alternate clustering technique, namely self organizing maps and the accuracy on GTEA dataset improved from 0.69 to 0.70.

6 Conclusion

Data intensive supervised approaches are difficult to use in privacy sensitive egocentric context. In this work we show that our simplistic and modular design for an unsupervised deep network is better than the existing state of the art supervised deep networks for first person action recognition task. We would also like to highlight that an ensemble of weak networks outperforms a single larger supervised network in our task. Through our experiments, we have shown that features learned from the proposed model are generic, and can be clustered at various semantic granularities. The proposed work significantly enhances the applicability of first person action recognition technique in practical scenarios.

References

[Erhan *et al.*, 2010] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, and Pascal Vincent.

Why does unsupervised pre-training help deep learning? In *JMLR*, 2010. 2

[Fathi *et al.*, 2011a] Alireza Fathi, Ali Farhadi, and James M Rehg. Understanding egocentric activities. In *ICCV*, 2011. 1, 2

[Fathi *et al.*, 2011b] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *CVPR*, 2011. 4, 5

[Fathi *et al.*, 2012] Alireza Fathi, Yin Li, and James M Rehg. Learning to recognize daily actions using gaze. In *ECCV*, 2012. 1

[Jones and Shao, 2014] Simon Jones and Ling Shao. Unsupervised spectral dual assignment clustering of human actions in context. In *CVPR*, 2014. 2

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 4

[Kitani *et al.*, 2011] Kris Makoto Kitani, Takahiro Okabe, Yoichi Sato, and Akihiro Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In *CVPR*, 2011. 1, 2, 3, 5

[Kuhn, 1955] Harold W. Kuhn. The hungarian method for the assignment problem. In *Naval Research Logistics Quarterly*, 1955. 5

[Lee *et al.*, 2012] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, 2012. 2

[Li *et al.*, 2015] Yin Li, Zhefan Ye, and James M. Rehg. Delving into egocentric actions. In *CVPR*, 2015. 2, 3

[Lu and Grauman, 2013] Zheng Lu and Kristen Grauman. Story-driven summarization for egocentric video. In *CVPR*, 2013. 2, 3

[Ma *et al.*, 2016] Minghuang Ma, Haoqi Fan, and Kris M. Kitani. Going deeper into first-person activity recognition. In *CVPR*, 2016. 1, 2, 3

[Matsuo *et al.*, 2014] K. Matsuo, K. Yamada, S. Ueno, and S. Naito. An attention-based activity recognition for egocentric video. In *CVPRW*, 2014. 1

[Niebles *et al.*, 2008] Juan Carlos Niebles, Hongcheng Wang, and Li Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *IJCV*, 2008. 2

[Ogaki *et al.*, 2012] Keisuke Ogaki, Kris Makoto Kitani, Yusuke Sugano, and Yoichi Sato. Coupling eye-motion and ego-motion features for first-person activity recognition. In *CVPRW*, 2012. 1

[Pirsiavash and Ramanan, 2012] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *CVPR*, 2012. 1, 2, 3, 4

[Poleg *et al.*, 2014] Yair Poleg, Chetan Arora, and Shmuel Peleg. Temporal segmentation of egocentric videos. In *CVPR*, 2014. 1, 2, 4, 6

[Poleg *et al.*, 2015] Yair Poleg, Ariel Ephrat, Shmuel Peleg, and Chetan Arora. Compact cnn for indexing egocentric videos. In *WACV*, 2015. 1, 2, 4, 5, 6

[Ryoo and Matthies, 2013] Michael S Ryoo and Larry Matthies. First-person activity recognition: What are they doing to me? In *CVPR*, 2013. 1

[Ryoo *et al.*, 2015] Michael S Ryoo, B Rothrock, and Larry Matthies. Pooled motion features for first-person videos. In *CVPR*, 2015. 4

[Singh *et al.*, 2016a] Suriya Singh, Chetan Arora, and C. V. Jawahar. Trajectory aligned features for first person action recognition. In *Pattern Recognition*, 2016. 1, 2, 3, 4

[Singh *et al.*, 2016b] Suriya Singh, Chetan Arora, and C V, Jawahar. First person action recognition using deep learned descriptors. In *CVPR*, 2016. 1, 2, 3, 4, 6

[Spriggs *et al.*, 2009] Ekaterina H Spriggs, Fernando De La Torre, and Martial Hebert. Temporal segmentation and activity classification from first-person sensing. In *CVPRW*, 2009. 1

[Srivastava *et al.*, 2015] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. In *JMLR*, 2015. 2, 4

[Stein and McKenna, 2013] Sebastian Stein and Stephen J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2013. 4

[Stein and McKenna, 2016] Sebastian Stein and Stephen J. McKenna. Recognising complex activities with histograms of relative tracklets. *Computer Vision and Image Understanding, Elsevier*, 154:82–93, 2016. 6

[Tieleman and Hinton, 2012] Tijmen Tieleman and Geoffrey Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude. In *COURSERA: Neural Networks for Machine Learning.Technical report*, 2012. 4

[Vincent *et al.*, 2010] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. In *JMLR*, 2010. 2

[Wang and Schmid, 2013] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 2, 3

[Wang *et al.*, 2011] Heng Wang, Alexander Klaser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *CVPR*, 2011. 2, 3

[Yang *et al.*, 2015] Huan Yang, Baoyuan Wang, Stephen Lin, David Wipf, Minyi Guo, and Baining Guo. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *ICCV*, 2015. 2

[Yang *et al.*, 2016] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *CVPR*, 2016. 5