How Shall We Evaluate Egocentric Action Recognition?

A. Furnari, S. Battiato, G. M. Farinella
University of Catania
{furnari,battiato,gfarinella}@dmi.unict.it

Abstract

Egocentric action analysis methods often assume that input videos are trimmed and hence they tend to focus on action classification rather than recognition. Consequently, adopted evaluation schemes are often unable to assess important properties of the desired action video segmentation output, which are deemed to be meaningful in real scenarios (e.g., oversegmentation and boundary localization precision). To overcome the limits of current evaluation methodologies, we propose a set of measures aimed to quantitatively and qualitatively assess the performance of egocentric action recognition methods. To improve exploitability of current action classification methods in the recognition scenario, we investigate how frame-wise predictions can be turned into action-based temporal video segmentations. Experiments on both synthetic and real data show that the proposed set of measures can help to improve evaluation and to drive the design of egocentric action recognition methods.

1. Introduction

State of the art methods for egocentric action analysis are generally designed to work on trimmed videos [25, 26, 37, 40, 52]. Under these settings, the original videos to be analyzed are assumed to be pre-segmented and methods take a short video clip as input and predict a label for it at inference time. While this scenario may be practical in the case of third person vision where video contents are often edited and hence "easier" to segment, it is particularly unlikely in the context of egocentric videos. Indeed, egocentric videos are often acquired in a continuous fashion and tend to be long and unstructured [13, 34]. To enable egocentric video understanding, methods should be able to segment an unedited sequence of frames to highlight the presence of specific actions. This includes detecting the temporal boundaries of the action (i.e., starting and ending frames), as well as its category. Moreover, as a matter of fact, video datasets for egocentric action analysis are generally collected in an "untrimmed" fashion, i.e., multiple subjects are asked to acquire a long video while they perform a set of egocentric actions or complex activities. Afterwards, videos are broken and manually annotated into labeled short segments (the trimmed videos) which can be used for supervised learning. Different egocentric video datasets have been acquired with this modality, e.g., BEOID [1], GTEA [8] and ADL [33]. This data collection scheme is generally different from the one employed for third person datasets, which tend to be organized as a collection of videos acquired from multiple sources [16, 42]. The former paradigm is akin to the one employed in the image classification task in which a single label has to be predicted for a given input image [3, 21]. The latter paradigm resembles object recognition, where objects have to be both localized and recognized in the image [11, 35].

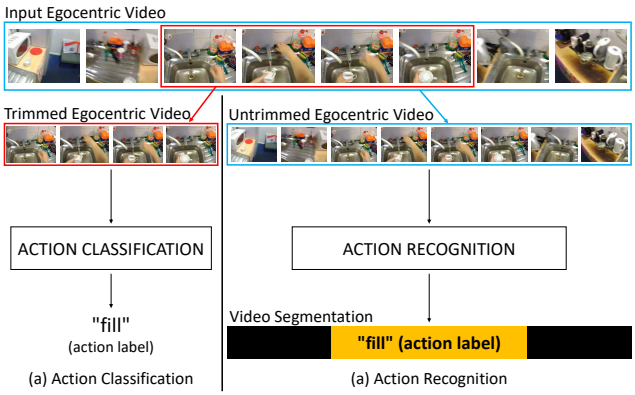


Figure 1. (a) Action classification versus (b) action recognition.

Reminiscent of the above distinction, in this paper we will refer to the trimmed scenario as "action classification" and to the untrimmed one as "action recognition". Other suitable terms for the action recognition scenario may be "temporal action localization" and "action-based video segmentation". Figure 1 shows two examples of the considered scenarios. We would like to emphasize that video segmentation is the desirable output in real egocentric vision applications. In a real application the input video cannot be reliably pre-segmented to proved trimmed clips to the action classification methods.

It should be observed that the distinction between trimmed and untrimmed scenarios is very well known in the literature related to third person action analysis. In such works, the two tasks are evaluated using different datasets and measures [16]. Nevertheless, as it will be better discussed in Section 2, works on egocentric action analysis do not generally account for the distinction. Even when there is explicit reference to the video segmentation task, methods are evaluated counting the number of correctly classified trimmed clips [20] or using frame-based measures [34, 43].

Such measures are not suitable to evaluate certain aspects of the produced results which are significant in real applications, e.g., boundary localization precision, misclassification, over-segmentation and under-segmentation [12].

We argue that lack of clarity in this regard can deceive the design of egocentric algorithms, and lead to the adoption of inconsistent evaluation approaches.

Our contribution to address the aforementioned issues is twice: 1) We investigate how action recognition methods should be evaluated both from a quantitative and qualitative point of view; 2) Since many action classification methods can provide frame-wise predictions, we investigate how their output can be exploited to obtain segment-based predictions and evaluate them properly using the considered measures.

We carry out the analysis by performing several experiments both in synthetic and real-world settings. Results highlight that the choice of a suitable set of evaluation measures is crucial to assess the actual performance of the considered methods and to guide the design of future approaches. The code of all considered measures is available at our web page *http://iplab.dmi.unict.it/EgoActionEvaluation/*.

2. Related Work

Different works have tackled the tasks of action classification and recognition both in the third person and first person scenarios.

Action Classification in Third Person Vision

Marszałek et al. [27] exploited scene context to improve action classification. Wang et al. designed dense trajectories [45] and improved trajectories [46] to encode local motion patterns and appearance for action classification. More recently, some notable action recognition methods based on deep learning have been proposed. Karpathy et al. [19] provided an extensive empirical evaluation of Convolutional Neural Networks (CNN) on large scale video classification exploring multiple approaches for adapting CNNs to videos. Simonyan et al. [39] designed a Two-Stream CNN (TS-CNN) capable of encoding both motion and appearance information to perform action classification. Feichtenhofer et al. [10] explored ways to spatio-temporally fuse the two streams to improve performance of the architecture. Wang et al. [47] proposed a framework for video-based action classification combining a sparse temporal sampling strategy and video-level supervision to enable learning from the whole action video.

All the aforementioned methods consider a pre-segmented (i.e., trimmed) video containing a single action as the basic unit of training and test. As result, most of these methods require videos to be pre-segmented also at inference time. Consequently, evaluation is generally carried out counting the number of correctly classified pre-segmented videos and analyzing performance by reporting the accuracy score or a confusion matrix.

Action Recognition in Third Person Vision

Duchenne et al. [5] addressed weakly-supervised learning and temporal localization of actions from third person videos. Action localization was evaluated using Average Precision (AP). Hoai et al. [17] investigated action recognition and automatic video segmentation as a join problem. Segmentation and classification performances are assessed jointly using fame-level accuracy. Qualitative assessment is also obtained with color-coded segmentation diagrams. Oneaţă et al. [30] exploited Fisher Vectors to aggregate a small set of low-level descriptors combined with linear classifiers. Action localization is obtained using Non-Maxima Suppression and results are measured in terms of mean Average Precision (mAP). Gaidon et al. [15] addressed the problem of localizing actions in hours of video introducing atomic action units termed "actoms". Performance is evaluated using Precision Recall curves and Average Precision considering different overlap thresholds to determine whether two segments constitute a correct match. Lea et al. [24] proposed Latent Convolutional Skip Chain Conditional Random Fields to learn a set of composable action primitives for action classification and localization. To evaluate localization results they propose two evaluation metrics which are designed to assess the influence of oversegmentation and offset.

Recently, the action recognition task in third person vision has been standardized thanks to the spread of new datasets and challenges. In particular the THUMOS challenge [16] defined the use of AP as a standard measure for the action localization task. As result, the measure has been largely adopted by recent methods [18, 38, 50].

Action Classification in First Person Vision

Most methods for action analysis in first person vision have focused on the classification scenario assuming pre-segmented videos as input [8, 7, 25, 26, 28, 37, 40, 48, 52].

Some authors used frame-based accuracy to evaluate their methods. Fathi et al. [7] presented a method to analyze daily activities performing inference about activities, actions, hands, and objects. Fathi et al. [8] also designed an approach for simultaneously classifying daily actions and predicting gaze.

Other authors evaluated performances in a trimmed scenario by counting the number of correctly classified videos and computing accuracy, average precision or confusion matrices. Ryoo et al. [37] proposed pooled motion features to encode several descriptors in a video-based representation for egocentric action classification. Li et al. [25] investigated a set of egocentric features and combined them with motion and object features for egocentric action classificaiton. Zhou et al. [52] built a framework for egocentric action classification which integrates feature maps based on motion, hands and active object region. Ma et al. [26] designed a CNN architecture to integrate several egocentric cues such as, hand segmentation, motion and active object detection to perform egocentric action and activity classification.

Action Recognition in First Person Vision

As anticipated, in first person vision, the distinction between action classification and recognition is not as clear as

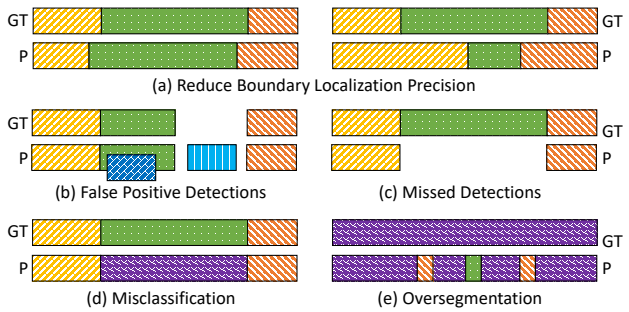


Figure 2. Different types of error affecting temporal action segmentation methods. GT stands for Ground Truth, while P stands for Predicted. Different colors represent different classes.

in the case of third person vision. In particular, even methods explicitly using terms such as "temporal segmentation" and "detection" are not evaluated using proper segment-based measures as happening in the third person vision scenario [9, 20, 33, 34, 36, 43, 49].

Spriggs et al. [43] explored the problem of inferring the activity performed in an egocentric video and segmenting each single action contained in the video. Pirsiavash et al. [33] proposed to exploit object-based representations to detect activities of daily living from egocentric videos. To localize activities, the authors considered a "background" class and employed a sliding window approach. Fathi et al. [9] presented a model for egocentric action analysis based on the changes in the state of objects and materials. A method for temporal activity segmentation is also investigated. Poleg et al. [34] introduced a method to segment long term activities such as running and standing from long egocentric videos. The previous discussed approaches have been evaluated using frame-based accuracy.

3. Egocentric Action Segmentation

Differently from algorithms working on a frame-by-frame basis, egocentric action recognition methods should output a consistent and temporally structured segmentation of the input video. Such output should be in principle very similar to the ground truth generally available for first-person action datasets [1, 8, 33], i.e., a set of temporal segments characterized by a starting frame, an ending frame and a class label. A confidence score is also generally assigned to each predicted segment.

Note that the output set of segments need not constitute a partition over the input video, i.e., some frames could refer to moments in which no action is being performed, in which case they would not be associated to any segment. Additionally, as noted by Spriggs et al. [43], especially when complex activities are performed, it is possible to have overlapping segments. Examples of these two properties are shown in Figure 2(b) where segmentations can contain "holes" (i.e., frames do not belonging to any segments) and segments may overlap.

Given its structured nature, egocentric video segmentation suffers from specific types of errors which can lower the quality of the obtained results. In particular, we consider five main issues: reduced boundary localization precision, false positive detections, missed detections, misclassification and oversegmentation.

Reduced boundary localization precision occurs when algorithms can recognize an egocentric action but predicted boundaries do not exactly match those reported in ground truth annotations. This phenomenon can be mild or severe as depicted in Figure 2(a). Assessing this type of error is not trivial since, as it has been noted by other authors [24, 48], annotators do not always agree on the temporal boundaries of specific action instances. However, this kind of evaluation is encouraged by recent work showing that accurate boundary localization [29] is important to perform correct action classification and that the annotation procedure can be standardized.

False positive detections consist in predicted action segments which do not match any ground truth segment. Examples of these errors are represented in Figure 2(b). Such errors can occur in areas in which no ground truth segment is present at all ("inactive" areas) or when additional overlapping segments are predicted.

Missed detections consist in ground truth segments which are not matched by any predicted segment at all. This is illustrated in Figure 2(c).

Misclassification occurs when a given action segment is correctly localized but the wrong class is assigned to it. An example of this error type is depicted in Figure 2(d). This error type is a special case of missed detections which may deserve particular attention as it allows to assess which component of the method is underperforming.

Oversegmentation occurs when multiple predicted action segments are contained within a single ground truth segment. An illustrated example of oversegmentation is reported in Figure 2(e). This error type is relevant in real scenarios in which the number and temporal extent of predicted actions might be significant. It should also be noted that, when the evaluation criterion allows a predicted segment to match only a single ground truth segment, this error type is very related to missed and false positive detections.

4. Evaluation Measures

4.1. Frame-Based Accuracy

Frame-Based accuracy is computed by predicting an action label for each frame of the video and considering the fraction of correctly classified frames. Since test videos usually contain also unlabeled parts, a "background class" is usually introduced, so that each frame can be assigned a unique label. Given its simplicity, this measure has largely been used to evaluate temporal segmentation results in past works [7, 8, 33, 34, 43]. While this measure can give a rough estimate on the percentage of times the method yields a correct label, it totally discards the temporal structure of the predictions and hence it is unfeasible to assess the actual segmentation capabilities of methods (e.g., for real applications).

4.2. Average Precision

Average Precision (AP) is a standard measure which can be used to evaluate recognition methods from an in-

formation retrieval perspective. This measure is commonly employed to evaluate object detection algorithms with respect to both object classification and localization [6]. In the context of temporal video segmentation, the measure is computed in a similar way, the main difference being that video segments are considered instead of bounding boxes and overlap is computed between temporal segments. The reader is referred to [6] for a review of the method in the context of object recognition.

Precision Recall curves and Average Precision are computed independently for each considered action class. A mean Average Precision (mAP) score is hence considered to summarize the performance of a given method. Given the data imbalance characterizing egocentric action datasets (i.e., some actions occur more often than others), in this paper, we compute mAP performing a weighted average of class-related AP scores. Weights are obtained considering the fraction of segments representing a given class contained in the ground truth.

4.3. MOTAP Curves

A limitation of AP consists in the fixed matching overlap threshold used for evaluation, which is usually set to 0.5. The choice of a fixed threshold allows to relax the constraint of an accurate segmentation. In this sense, a prediction perfectly matching the ground truth is evaluated in the same way as a prediction which retains an overlap of 0.5 with a ground truth segment. This is convenient when ground truth action boundaries are not reliable enough. However, as already investigated in [15], it can be very informative to inspect Matching Overlap Threshold - Average Precision curves (MOTAP). A MOTAP curve plots Average Precision against the overlap threshold used to match predicted and ground truth segments and shows the behavior of methods when different levels of boundary localization precision are required. This evaluation is the most suitable to assess the influence of different degrees of reduced boundary localization precision, as illustrated in the two examples reported in Figure 2(a). The Area Under an Overlap Threshold Average Precision curve (AUMOTAP) can be used to summarize the performance of methods over different levels of localization precision. Also in this case, mean MOTAP curves and mean AUMOTAP values are obtained performing a weighted average of class-related scores.

4.4. Precision and Recall

Average Precision summarizes over precision and recall. However, it might still be desirable to assess whether the main source of error concerns false positive or missed detections (Figure 2(b) and Figure 2(c)). Such two types of error are easily assessed computing precision and recall over the segments matched during the computation of Average Precision:

$$precision = \frac{\#\ matched\ predicted\ segments}{\#\ predicted\ segments} \quad (1)$$

$$recall = \frac{\#\ matched\ gt\ segments}{\#\ gt\ segments} \quad (2)$$

Specifically, precision is inversely proportional to the number of false positive detections, while recall is inversely proportional to the number of missed detections. In this paper, we do not consider confidence scores for the computation of precision and recall values and use the standard matching overlap threshold of 0.5.

4.5. Classification Precision

To assess the influence of misclassification (Figure 2(d)), we propose the classification precision score. We define this score as the ratio between the number of predicted segments matched to a ground truth segment of the same class and the number of predicted segments matched to a ground truth segment of any class:

$$c.prec. = \frac{\#pred.\ seg.\ match.\ with\ correct\ class}{\#pred.\ seg.\ match.\ with\ any\ class} \quad (3)$$

Also in this case, we do not consider confidence scores for the computation and use the standard matching overlap threshold of 0.5.

4.6. Inverse Oversegmentation Rate

Some authors have considered methods to assess and penalize the influence of oversegmentation in action recognition results. In particular, Lea et al. [24] proposed to use the maximum overlap between ground truth and predicted segments as a score. In case of oversegmentation, the maximum overlap between the ground truth segment and one of the predicted ones will be reduced (see Figure 2(e) for an example). To keep our measures within the framework of Average Precision, we propose the Inverse Oversegmentation Rate. We first define the Oversegmentation Rate (OR) as the ratio between the number of ground truth segments matching more than one predicted segment and the number of ground truth segments matching at least one predicted segment. Intuitively, in the case of oversegmentation, a given ground truth segment will be matched more than once and the OR score will increase. Hence, we define the Inverse Oversegmentation Rate (IOR) as follows:

$$IOR = 1 - \frac{\#gt\ seg.\ matched\ more\ than\ once}{\#gt\ seg.\ matched\ at\ least\ once} \quad (4)$$

In this case, we use 0 as matching overlap threshold and do not use scores for the computation of matches. The choice of the zero threshold is important to allow for the matching of predicted segments which might be very small due to oversegmentation.

5. Converting Frame-Wise Predictions into Temporal Segments

Many action analysis methods can be used to produce frame-wise predictions [10, 34, 39, 47]. When performances are evaluated using trimmed segment-based accuracy, predictions obtained for one or more frames within the segment are usually averaged to obtain a single posterior probability over the video segment. The segment is finally

classified by considering the label maximizing the posterior probability and assigning the corresponding probability value as confidence score.

To take advantage of such methods in the untrimmed scenario, it is necessary to convert frame-wise predictions into temporal segmentations. In the following sections, we discuss some available options to perform such conversion.

5.1. Segmentation by Connected Components (CC)

The most straightforward way to convert frame-wise predictions into temporal segments is to extract all connected components from the output list of labels. Given an extracted temporal segment, a confidence score can be assigned to it by averaging all posterior probabilities within the segment and selecting the maximum probability value.

5.2. Rejecting Negatives

In the untrimmed action recognition scenario, videos are likely to contain "negative" frames, i.e., frames in which no specific action is being performed. Egocentric action recognition methods should be able to correctly localize and classify actions and discard all negative frames or segments where no action is performed. This is a key property for real systems. This result can be achieved in different ways.

A simple strategy employed by some authors [33] is to train the method to recognize a negative or "background" class, by considering sequences where no action is performed. This basically adds one more class to the set of positive classes to be learned. After obtaining temporal segments from frame-wise predictions (e.g., by considering connected components), segments classified as belonging to the negative classes are simply discarded.

Another strategy investigated in this paper is to train an action classification method to discriminate between positive sequences (i.e., any sequence in which one of the considered classes is being performed) and negative ones (i.e., any other sequence). Once trained, the method can be used to obtain a "positive vs negative" segmentation (e.g., by considering connected components over frame-wise predictions). Negative segments are hence discarded and positive ones are classified by considering the average posterior probability over all frames contained in the segment.

Other approaches to reject negatives in temporal egocentric segmentation exist. For instance, in the context of location-based temporal segmentation of egocentric videos, some authors leveraged local entropy among predictions [14] or considered the use of heuristics [31, 32].

5.3. Enforcing Temporal Smoothing via a Hidden Markov Model

We also investigate the possibility of enforcing temporal coherence between predictions using a Hidden Markov Model (HMM). Since no general assumption between the possible order of performed actions is assumed, a HMM with a diagonal matrix is employed as done in [12, 44]. The considered HMM depends on a single parameter ε which controls the "amount of smoothing" induced in the state transition probabilities. We set to $\varepsilon = 10^{-4}$ in our experiments. The role of the considered HMM is to enforce temporal smoothing of the produced labels and avoid errors due to random label changes which can lower the quality of segmentations obtained considering connected components.

5.4. Non-Maxima Suppression (NMS)

A fairly standard way to obtain segmentation from frame-wise predictions is to employ Non-Maxima Suppression (NMS) [5, 30, 33]. In our experiments, we implement Non-Maxima Suppression as described by Oneaţă et al. [30]. Specifically: 1) To reduce the tendency of NMS to favor short windows, the proposed segments are re-scored multiplying confidence scores by their duration prior to applying NMS; 2) Allowed overlap for segments of the same class is set to 0 (versus common values of 0.2 or 0.5). This allows to obtain a non-overlapping set of predicted segments. However, it should be noted that, since non-maxima suppression is applied separately for each class, overlap between segments belonging to different classes is still allowed. Considering the minimum and maximum duration of ground truth segments, we produce proposals with a minimum scale of 10 frames and a maximum scale of 240 frames. Scales and temporal positions of proposals are varied with a stride of 1 frame.

6. Experimental Analysis

We performed two sets of experiments. The first one is aimed at assessing the properties of the evaluation scores discussed in Section 4 in controlled settings. Specifically, we perform a series of experiments with simulated data in which we artificially introduce errors of the types discussed in Section 3. The second set of experiments is performed on real data and is aimed at assessing the performance of the different methodologies to convert frame-wise predictions into temporal segments discussed in Section 5.

We considered the BEOID dataset [1] for our experiments. The dataset contains 58 videos acquired by 8 different subjects in 6 different environments. The dataset is provided with annotations for different actions performed by the subjects. After removing actions represented by less than 10 frames, we obtain a set of 29 different action labels. To perform experiments with real data, the dataset is randomly divided into three splits. We constrain the splits to contain data from different subjects in order to avoid overfitting.

6.1. Experiments with Synthetic Data

To assess the properties of the considered evaluation measures in controlled settings, we generate synthetic data starting from the ground truth annotations of BEOID dataset. Generated data takes the form of the output of a potential system to be evaluated with respect to ground truth.

Synthetic data is obtained by first copying the set of ground truth annotations for each video in the dataset, then perturbing them according to the schemes discussed below. Confidence scores are assigned to synthetic predictions by drawing random numbers comprised between 0 and 1. Note that this choice is considered to ensure that segments are

sorted randomly in the computation of AP scores. To mimic the presence of a specific error type, we build 5 different sets of simulated predictions according to the following schemes:

Reduced Boundary Precision. We first select a parameter $\sigma_{\%} \in [0, 1]$. For each segment, each of the two temporal boundaries b is modified by drawing a random number from a Gaussian distribution centered at b and with standard deviation equal to $\sigma = \sigma_{\%} \cdot d$, where d is the duration of the segment.

False Positive Detections. Chosen a parameter n, synthetic data is obtaining by introducing n new segments at random positions and with random duration in each video of the dataset. A random class is assigned to each newly introduced segment.

Missed Detections. A parameter $\alpha_{\%} \in [0, 1]$ is first chosen. For each video, $n = \alpha_{\%} \cdot m$ segments are deleted, where m is the total number of segments in the video.

Misclassification. Similarly to the case of missed detections, we choose a parameter $\alpha_{\%} \in [0, 1]$ and randomly change the class of $n = \alpha_{\%} \cdot m$ segments in each video, where m is the total number of segments in the video.

Oversegmentation. Synthetic predictions are obtained splitting a randomly selected segment in two parts. The operation is repeated n times on each video. At each iteration, the algorithm is allowed to choose a previously splitted segment.

Figure 3 reports the scores obtained with each considered measure on synthetic generated output segmentation as described in the previous section. Each plot reports how scores vary for different amounts of the considered perturbation.

Figure 3(a)-left reports Matching Overlap Threshold - Average Precision (MOTAP) curves for different choices of parameter $\sigma_{\%}$. As can be observed, increasing levels of boundary perturbation affect the decay of MOTAP curves and hence the related values of area under the curve which are reported in parenthesis in the legend. The plot on Figure 3(a)-right summarizes the trend of all scores with respect to different amounts of boundary perturbation. Both mAP and mAUMOTAP scores decay for increasing amounts of perturbation. However, mAP retains very high values up to a significant amount of perturbation of about $\sigma_{\%} = 0.2$. On the other hand, when $\sigma_{\%}$ exceeds 0.4, the mAUMOTAP score retains larger values than mAP, indicating that, while segments are not accurately localized, they still retain some small overlap with the ground truth. Reduced boundary localization precision also affects precision and recall, which follow the trend of the mAP score. Classification precision and IOR score retain large values and hence are not significantly affected by the perturbation. Note that this works as expected since the scores have not been designed to respond to the specific error type under analysis. Also frame-based accuracy is marginally influenced by the perturbation under analysis.

Figure 3(b) reports results related to experiments concerning false positive detections. As can be noted, the trend of the precision score smoothly decays when the number of insertions is increased. mAP and mAUMOTAP follow a similar trend. Note that mAP and mAUMOTAP scores are perfectly overlapped since boundaries are not perturbed in this experiment. Similarly to Figure 3(a), scores related to other error types (i.e., oversegmentation, classification precision and recall) are not affected by the perturbation as one would expect.

Figure 3(c) reports results related to experiments on missed detections. Recall values closely follow the trend of mAP and mAUMOTAP and decay for increasing amounts of perturbation. Other scores are not directly related to this error type and hence they are not affected by the perturbation.

Figure 3(d) reports results related to misclassification. Classification precision and precision score are perfectly overlapping and follow the trend of mAP and mAUMOTAP. Also recall is affected by the perturbation. However, it should be noted that this is the only plot in which the classification precision score is significantly decreasing. IOR and frame-based accuracy are not affected by the perturbation as expected.

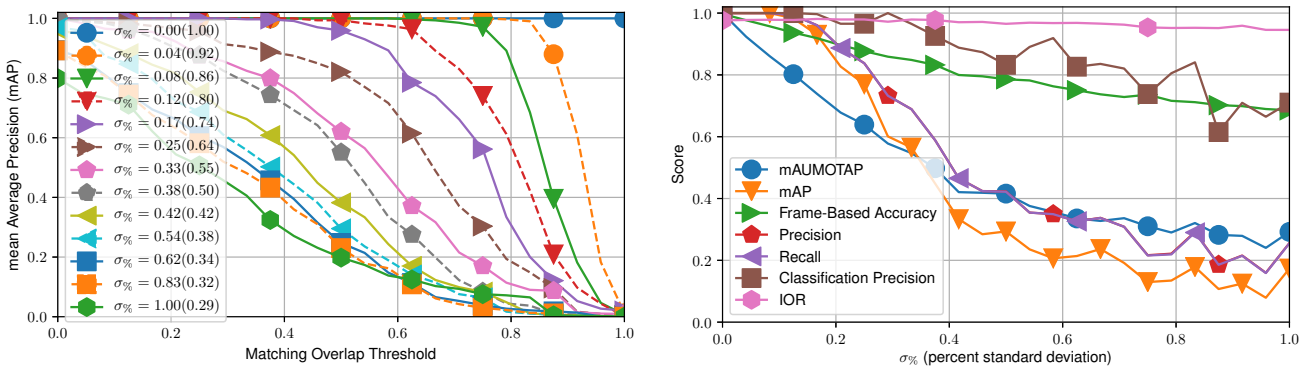
Figure 3(e) reports results related to experiments concerning oversegmentation. Differently than previous cases, the IOR score is significantly affected by the perturbation. As it can be expected, this error type also affects overall precision and, marginally, recall. mAUMOTAP and mAP follow similar trends and decay with increasing amounts of perturbation. Classification precision is not significantly affected since the main source of error is not misclassification. Frame-based accuracy is totally unable to capture this type of error and hence always retains the maximum value.

In general, mAP and mAUMOTAP scores are affected by all considered perturbations as one would expect. IOR and classification precision are significantly affected only by specific types of error, i.e., oversegmentation and misclassification respectively. This suggest that such scores can be effectively used as indicators for the specific error types they have been designed for. Precision and recall clearly respond to the introduction of false positive and missed detections. However they are also sensitive to other related errors which are special cases of false positive detections (e.g., oversegmentation) and missed detections (e.g., misclassification). Nevertheless, they are still suitable to assess what is the primary cause leading to low mAP values. Frame-based accuracy does not account for the temporal structure of the predictions and hence it is unable to capture any of the investigated error types.

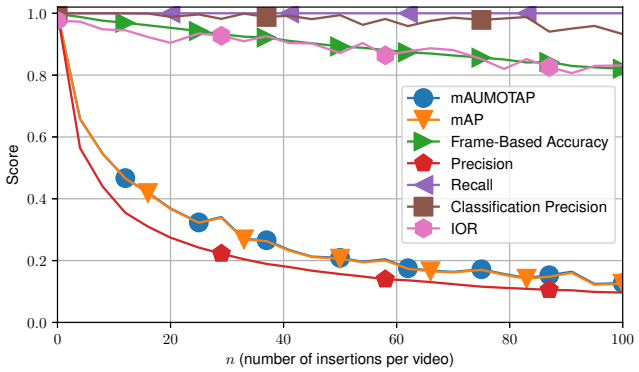
6.2. Experiments with Real Data

We finally perform experiments with real data. These experiments are aimed to assess the descriptiveness of the introduced measures in a real scenario, as well as to compare the different techniques to turn frame-wise predictions into temporal segmentations discussed in Section 5.

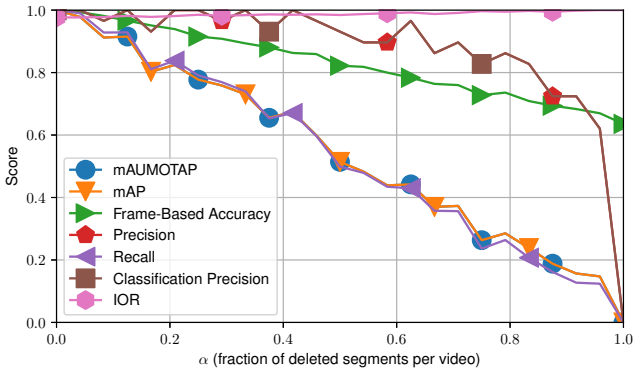
To perform experiments on real data, we train the state-of-the-art method proposed in [10] on the BEOID dataset. The considered model has been designed to classify actions in a trimmed scenario, but it can be used to obtain frame-wise predictions. Specifically, we train three different models. The first model is trained to discriminate between the



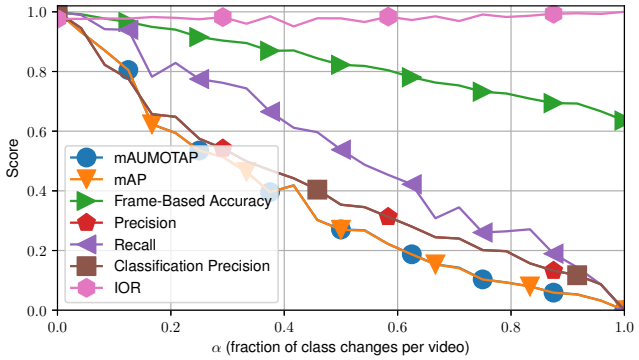
(a) Reduced Boundary Localization Precision



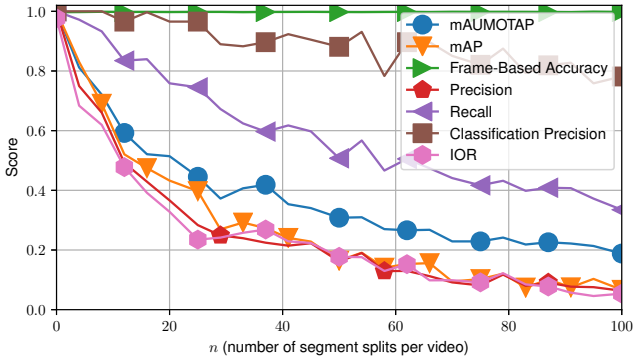
(b) False Positive Detections



(c) Missed Detections



(d) Miscalssification



(e) Oversegmentation

Figure 3. Results related to the artificial introduction of different types of errors in the ground truth segmentations. In plot (a), mean area under MOTAP curves is reported for each method in parenthesis in the legend.

Method	mAUMOTAP	mAP	F-B Accuracy	Precision	Recall	C. Precision	IOR
CCC	21,74	23,16	82,38	24,36	34,57	63,85	81,18
CCC*	29,15	32,39	82,20	44,65	34,25	60,43	99,41
NMS	31,10	23,44	86,96	28,67	29,22	70,65	98,37
CC	43,79	47,23	86,96	25,11	52,15	**84,13**	72,63
CC*	**45,32**	**53,12**	**86,97**	**65,20**	**57,70**	83,56	**99,97**

Table 1. Results related to experiments performed on real data. Methods marked with "*" employ an HMM to smooth predictions. "F-B accuracy" stands for Frame-Based Accuracy, while "C. Precision" stands for Classification Precision. Best scores are reported in bold.

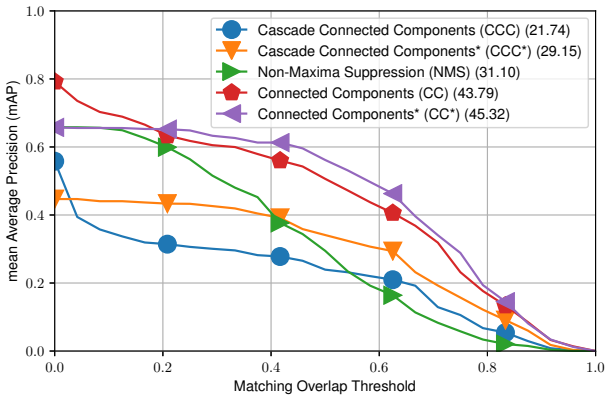


Figure 4. MOTAP curves related to experiments performed on real data. Mean area under the MOTAP curve is reported for each method in the legend. Methods marked with "*" employ an HMM to smooth predictions.

29 positive classes of the BEOID dataset. The second model is trained to discriminate between 30 classes including the original 29 positive classes, plus one additional negative class. The third one is trained to discriminate only between "positive" and "negative" sequences. We consider a "positive" sequence as any temporal segment labeled in the BEOID dataset and a "negative" as any other unlabeled video segment. Each model is trained three times according to the considered splits of the BEOID dataset in order to obtain unbiased predictions for each video.

The obtained predictions are combined with the techniques discussed in Section 5 to obtain the following methods:

Cascade Connected Components (CCC): positive vs negative frame-wise predictions are turned into positive vs negative segments applying the connected components method. Negative segments are discarded and positive ones are classified considering the predictions produced by the model trained on the 29 positive classes.

Cascade Connected Components* (CCC*): similar to the CCC method with the exception that the HMM is used to obtain initial positive vs negative frames.

Non-Maxima Suppression (NMS): predictions obtained using the model trained on the 30 classes (29 positive classes, plus a "negative" class) are turned into segments using non-maxima suppression. Negative segments are discarded.

Connected Components (CC): predictions obtained using the model trained on the 30 classes are turned into segments using the connected components method. Negative segments are discarded.

Connected Components* (CC*): similar to the CC method with the exception that the HMM is used to obtain labels before the extraction of connected components.

Figure 4 reports MOTAP curves related to experiments performed on real data. CCC is the least accurate method. Introducing an HMM to temporally smooth predictions allows to obtain an improvement of about 0.08 in terms of mAUMOTAP score. NMS performs better than the aforementioned methods especially for smaller matching overlap threshold values. This suggests that some segments are correctly detected by the method but not accurately localized. CC and CC* methods retain the best performances as compared to all competitor methods. Also in this case, the introduction of a Hidden Markov Model allows to improve overall localization accuracy.

Table 1 reports results according to the different considered evaluation measures. Interestingly, rankings obtained using mAUMOTAP and mAP scores do not always agree. This is, for instance, the case of the NMS method which outperforms CCC* only according to the mAUMOTAP. This is explainable by observing how MOTAP curves related to the two methods cross at a matching overlap threshold of about 0.4 in Figure 4. Observing the curve, it is evident how the NMS method actually dominates the competitor CCC* and hence how the mAUMOTAP score summarize the difference in performance better than mAP.

Also in this case, frame-based accuracy does not capture the difference in performance significantly. The introduction of the HMM allows to greatly improve precision (compare CCC* to CCC and CC* to CC) while keeping similar recall values and slightly decreasing classification precision. The introduction of the HMM allows to boost the IOR score (from 81.18 to 99, 41 for CCC and from 72, 63 to 99, 97 for CC). The main limits of non-maxima suppression seem to be related to reduced recall (i.e., many detections are missed) and reduced classification precision (i.e., many detection are assigned the wrong class).

7. Conclusion

We have investigated how egocentric action recognition methods should be evaluated. To overcome the limits of current evaluation schemes, we have proposed a set of different measures aimed to provide both qualitative and quantitative performance assessment of egocentric action recognition approaches. To better exploit current action classification methods, we have also investigated how frame-wise predictions can be turned into temporal segmentations. Experiments on both synthetic and real data have shown that the considered measures are highly descriptive and can be used to get qualitative insights on the performance of methods. Future works will be devoted to extend experiments to more egocentric datasets and state of the art methods.

Acknowledgment

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

[1] D. Damen, T. Leelasawassuk, and W. Mayol-Cuevas. You-Do, I-Learn: Egocentric unsupervised discovery of objects and their modes of interaction towards video-based guidance. *Computer Vision and Image Understanding*, 149:98–112, 2016.

[2] J. W. Davis and A. F. Bobick. The representation and recognition of human movement using temporal templates. In *International Conference on Computer Vision and Pattern Recognition*, pages 928–934, 1997.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[4] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005.

[5] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *International Conference on Computer Vision*, pages 1491–1498, 2009.

[6] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[7] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *International Conference on Computer Vision*, pages 407–414, 2011.

[8] A. Fathi, Y. Li, and J. Rehg. Learning to recognize daily actions using gaze. *European Conference on Computer Vision*, pages 314–327, 2012.

[9] A. Fathi and J. M. Rehg. Modeling actions through state changes. In *Computer Vision and Pattern Recognition*, pages 2579–2586.

[10] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Computer Vision and Pattern Recognition*, pages 1933–1941, 2016.

[11] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Visual object detection with deformable part models. *Communications of the ACM*, 56(9):97–105, 2013.

[12] A. Furnari, S. Battiato, and G. M. Farinella. On the exploitation of hidden markov models to improve location-based temporal segmentation of egocentric videos. In *Workshop on Wearable MultiMedia at ACM ICMR 2017*, 2017.

[13] A. Furnari, G. M. Farinella, and S. Battiato. Temporal segmentation of egocentric videos to highlight personal locations of interest. In *European Conference on Computer Vision Workshops*, volume 9913 of *Lecture Notes in Computer Science*, pages 474–489, 2016.

[14] A. Furnari, G. M. Farinella, and S. Battiato. Recognizing personal locations from egocentric videos. *IEEE Transactions on Human-Machine Systems*, 47(1):6–18, 2017.

[15] A. Gaidon, Z. Harchaoui, and C. Schmid. Temporal localization of actions with actoms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2782–2795, 2013.

[16] A. Gorban, H. Idrees, Y.-G. Jiang, A. Roshan Zamir, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. `http://www.thumos.info/`, 2015.

[17] M. Hoai, Z.-Z. Lan, and F. De la Torre. Joint segmentation and classification of human actions in video. In *Computer Vision and Pattern Recognition*, pages 3265–3272, 2011.

[18] D.-A. Huang, L. Fei-Fei, and J. C. Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *European Conference on Computer Vision*, pages 137–153, 2016.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[20] K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In *Computer Vision and Pattern Recognition*, pages 3241–3248, 2011.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[22] I. Laptev and P. Pérez. Retrieving actions in movies. In *International Conference on Computer Vision*, pages 1–8, 2007.

[23] C. Lea, A. Reiter, R. Vidal, and G. D. Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In *European Conference on Computer Vision*, pages 36–52, 2016.

[24] C. Lea, R. Vidal, and G. D. Hager. Learning convolutional action primitives for fine-grained action recognition. In *International Conference on Robotics and Automation*, pages 1642–1649. IEEE, 2016.

[25] Y. Li, Z. Ye, and J. M. Rehg. Delving into egocentric actions. In *Computer Vision and Pattern Recognition*, pages 287–295, 2015.

[26] M. Ma, H. Fan, and K. M. Kitani. Going deeper into first-person activity recognition. In *Computer Vision and Pattern Recognition*, pages 1894–1903, 2016.

[27] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *Computer Vision and Pattern Recognition*, pages 2929–2936, 2009.

[28] T. McCandless and K. Grauman. Object-centric spatio-temporal pyramids for egocentric activity recognition. In *British Machine Vision Conference*, volume 2, page 3, 2013.

[29] D. Moltisanti, M. Wray, W. Mayol-Cuevas, and D. Damen. Trespassing the boundaries: Labeling temporal bounds for object interactions in egocentric video. *International Conference on Computer Vision (ICCV)*, 2017.

[30] D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with fisher vectors on a compact feature set. In *International Conference on Computer Vision*, pages 1817–1824, 2013.

[31] A. Ortis, G. M. Farinella, V. D'Amico, L. Addesso, G. Torrisi, and S. Battiato. Organizing egocentric videos for daily living monitoring. In *First Workshop on Lifelogging Tools and Applications*, pages 45–54, 2016.

[32] A. Ortis, G. M. Farinella, V. DAmico, L. Addesso, G. Torrisi, and S. Battiato. Organizing egocentric videos of daily living activities. *Pattern Recognition*, 72:207 – 218, 2017.

[33] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *Computer Vision and Pattern Recognition*, pages 2847–2854. IEEE, 2012.

[34] Y. Poleg, C. Arora, and S. Peleg. Temporal segmentation of egocentric videos. In *Computer Vision and Pattern Recognition*, pages 2537–2544, 2014.

[35] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[36] M. S. Ryoo and L. Matthies. First-person activity recognition: What are they doing to me? In *Computer Vision and Pattern Recognition*, pages 2730–2737, 2013.

[37] M. S. Ryoo, B. Rothrock, and L. Matthies. Pooled motion features for first-person videos. In *Computer Vision and Pattern Recognition*, pages 896–904, 2015.

[38] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Computer Vision and Pattern Recognition*, pages 1049–1058, 2016.

[39] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[40] S. Singh, C. Arora, and C. Jawahar. First person action recognition using deep learned descriptors. In *Computer Vision and Pattern Recognition*, pages 2620–2628, 2016.

[41] P. Smith, N. da Vitoria Lobo, and M. Shah. Temporalboost for event recognition. In *International Conference on Computer Vision*, volume 1, pages 733–740, 2005.

[42] K. Soomro, A. Roshan Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

[43] E. H. Spriggs, F. De La Torre, and M. Hebert. Temporal segmentation and activity classification from first-person sensing. In *Computer Vision and Pattern Recognition Workshops*, pages 17–24, 2009.

[44] R. Templeman, M. Korayem, D. J. Crandall, and A. Kapadia. PlaceAvoider: Steering First-Person Cameras away from Sensitive Spaces. In *Annual Network and Distributed System Security Symposium*, pages 23–26, 2014.

[45] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79, 2013.

[46] H. Wang and C. Schmid. Action recognition with improved trajectories. In *International Conference on Computer Vision*, pages 3551–3558, 2013.

[47] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36, 2016.

[48] M. Wray, D. Moltisanti, W. Mayol-Cuevas, and D. Damen. Sembed: Semantic embedding of egocentric action videos. In *European Conference on Computer Vision Workshops*, pages 532–545, 2016.

[49] Y. Yan, E. Ricci, G. Liu, and N. Sebe. Egocentric daily activity recognition via multitask clustering. *IEEE Transactions on Image Processing*, 24(10):2984–2995, 2015.

[50] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *Computer Vision and Pattern Recognition*, pages 2678–2687, 2016.

[51] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *Computer Vision and Pattern Recognition*, volume 2, 2001.

[52] Y. Zhou, B. Ni, R. Hong, X. Yang, and Q. Tian. Cascaded interactional targeting network for egocentric video analysis. In *Computer Vision and Pattern Recognition*, pages 1904–1913, 2016.