Spatio-Temporal Action Instance Segmentation and Localisation

Suman Saha, Gurkirt Singh, Michael Sapienza, Philip H. S. Torr, and Fabio Cuzzolin

Published in:

Modelling human motion: From human perception to robot design [ISBN: 9783030467319] / edited by Nicoletta Noceti, Alessandra Sciutti, Francesco Rea (Springer, 2020).

Abstract Current state-of-the-art human action recognition is focused on the classification of temporally trimmed videos in which only one action occurs per frame. In this work we address the problem of action localisation and instance segmentation in which multiple concurrent actions of the same class may be segmented out of an image sequence. We cast the action tube extraction as an energy maximisation problem in which configurations of region proposals in each frame are assigned a cost and the best action tubes are selected via two passes of dynamic programming. One pass associates region proposals in space and time for each action category, and another pass is used to solve for the tube's temporal extent and to enforce a smooth label sequence through the video. In addition, by taking advantage of recent work on action foreground-background segmentation, we are able to associate each tube with class-specific segmentations. We demonstrate the performance of our algorithm on the challenging LIRIS-HARL dataset and achieve a new state-of-the-art result which is 14.3 times better than previous methods.

8.1 Introduction

The existing competing approaches [8, 18, 21, 25] address the problem of action detection in a setting where videos contain single action category and most of them are temporally trimmed. In contrast, this chapter addresses the problems of both spatio-temporal action instance segmentation and action detection. Here, we consider real-world scenarios where videos often contain co-occurring action instances belong to different action categories. Consider the example shown in Fig. 8.1, where our proposed model performs action instance segmentation and detection of two cooccurring actions "leaving bag unattended" and "handshaking" which have different spatial and temporal extents within the given video sequence. The video is taken from the LIRIS-HARL dataset [13]. In this chapter, we propose a deep learning based framework for both action instance segmentation and detection, and evaluate the proposed model on the LIRIS-HARL dataset which is more challenging than the standard benchmarks: UCF-101-24 [23] and J-HMDB-21 [13] due to its multilabel and highly temporally untrimmed videos. To demonstrate the generality of the segmentation results on other standard benchmarks, we present some additional qualitative action instance segmentation results on the standard UCF-101-24 dataset (Sect. 8.4.4).

Outline. This chapter is organized as follows. First we present an overview of the approach in Sect. 8.2. We then introduce the detailed methodology in Sect. 8.3. Finally, Sects. 8.4 and 8.5 present the experimental validation and discussion respectively.

Related publication. The work presented in this chapter has appeared in arXiv [20].

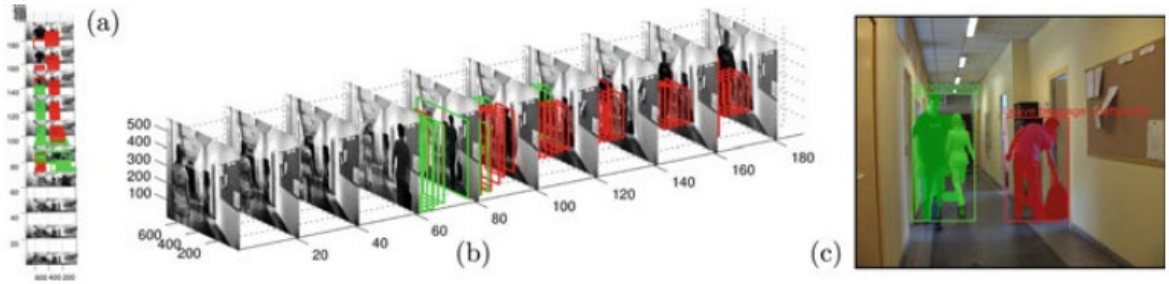


Fig. 8.1 A video sequence taken from the LIRIS-HARL dataset plotted in space-and time. a A top down view of the video plotted with the detected action tubes of class “handshaking” in green, and “person leaves baggage unattended” in red. Each action is located to be within a space-time tube. b A side view of the same space-time detections. Note that no action is detected at the beginning of the video when there is human motion present in the video. c Action instance segmentation results for two actions occurring simultaneously in a single frame

8.2 Overview of the Approach

An overview of the algorithm is depicted in Fig. 8.2. At test time, we start by performing binary human motion segmentation (a) for each input video frame by leveraging the human action segmentation [17], followed by a frame-level region proposal generation (b) (Sect. 8.3.1.1). Proposal bounding boxes are then used to crop patches from both RGB and optical flow frames (c). We refer readers to Section A.1 of [19] for details on optical flow frame computation. Crop image patches are resized to a fixed dimension and fed as inputs to an appearance- and a motion-based detection net work (d) (Sect. 8.3.2) to compute CNN fc7 features. Subsequently, these appearanceand motion-based fc7 features are fused, and later, these fused features are classified by a set of one-versus-all SVMs. Each fused feature vector is a high-level image representation of its corresponding warped region and encodes both static appearance (e.g. boundaries, corners, object shapes) and motion pattern of human actions (if there is any). Finally, the top k frame-level detections (regions with high classification scores) are temporally linked in time to build class-specific action tubes (e) and then, these tubes are trimmed (as in [21]) to solve for temporal action locali sation. Pixels belonging to each action tube are assigned class- and instance-aware action labels by taking advantage of both tube’s class score and the binary action segmentation maps computed in (a). At train time, first action region hypotheses are generated for RGB video frames using Selective Search [24] (Sect. 8.3.1.2), then, pretrained appearance and motion CNNs (d) are fine-tuned on the warped regions extracted from both RGB and flow frames. Subsequently, fine-tuned appearance and motion CNNs are used to compute fc7 features from both RGB and flow training frames, features are then fused and pass as inputs to a set of one-versus-all SVMs for training. A detailed descriptions of these above steps are presented in Sect. 8.3.

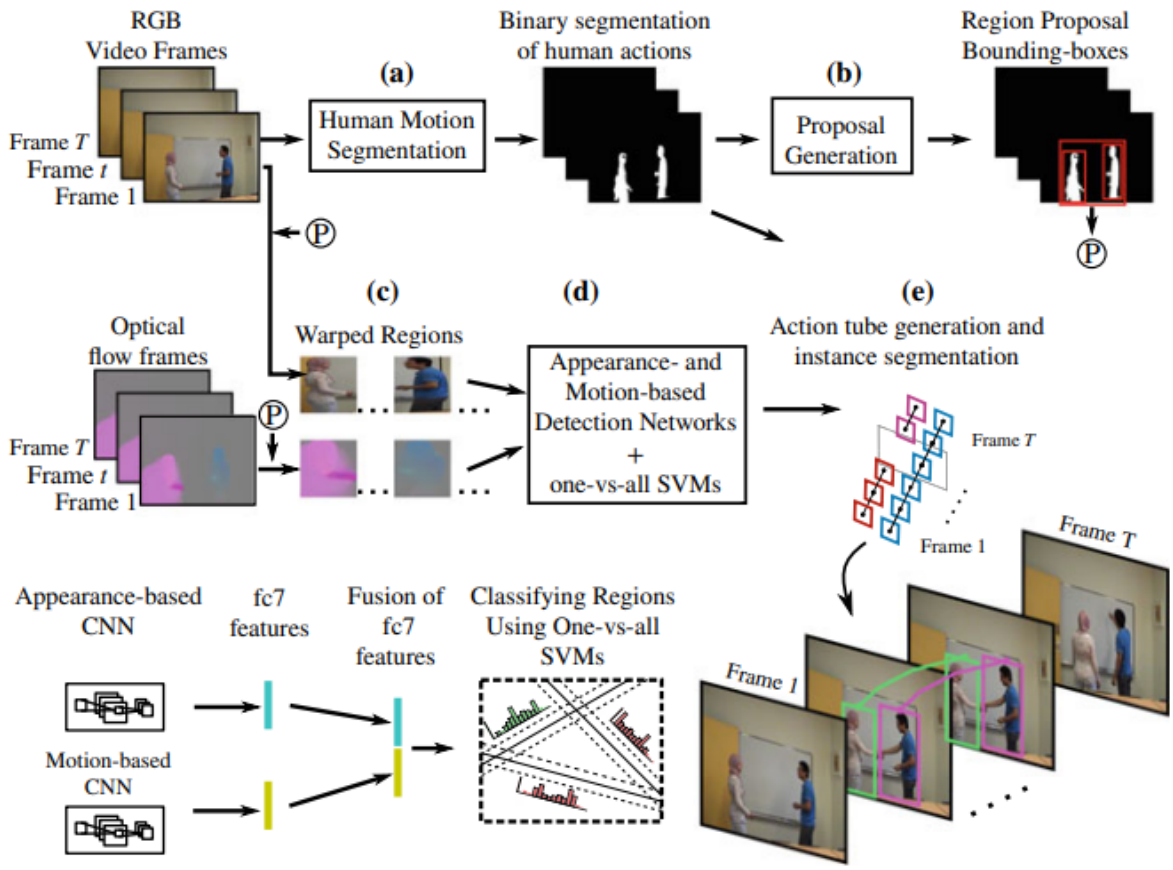


Fig. 8.2 Overview of the proposed spatio-temporal action instance segmentation and detection pipeline. At test time, a RGB video frames are fed as inputs to a human motion segmentation algorithm to generate binary segmentation of human actions; at this point these human silhouettes do not carry any class- and instance-aware labels, and they only have binary labels for foreground (and the pixels don't belonging to human silhouettes are labelled as background class). b Our region proposal generation algorithm accepts the binary segmented video frames as inputs and computes region proposal bounding boxes using all possible combinations of 2D connected components (2N − 1) present in the binary map. c Once the region proposals are computed, warped regions are extracted from both RGB and optical flow frames and fed as inputs to the respective appearance- and motion-based detection networks. d The detection networks compute fc7 appearance and motion features for each warped region, features are then fused and subsequently used by a set of one-vsall SVMs to generate action classification scores for each region. e Finally, frame-level detection windows are temporally linked as per their class-specific scores and spatial overlaps to build classspecific action tubes. Further, each pixel within the detection windows is assigned to an class- and instance-aware label by by utilising both the bounding-box detections associated with each classspecific action tubes and the binary segmentation maps (or human silhouettes) generated in (a)

8.3 Methodology

8.3.1 Region Proposal Generation

We denote each 2D region proposal ‘r’ as a subset of the image pixels, associated with a minimum bounding box ‘b’ around it. In the following sub sections we present our two different region proposal generation schemes: (1) the first one is based on human motion segmentation algorithm [17], and (2) the second one uses Selective Search algorithm [24] to generate 2D action proposals.

8.3.1.1 Proposals Based on Motion Segmentation

The human motion segmentation [17] algorithm generates binary segmentation of human actions (Fig. 8.2a). It extracts human motion from video using long term trajectories [3]. In order to detect static human body parts which don’t carry any motion but are still significant in the context of the whole action, it attaches scores to these regions using a human shape prior from a deformable part-based (DPM) model [6]. By striking balance between the human motion and static human-body appearance information, it generates binary silhouettes of human actions in space and time. At test time our region proposal algorithm accepts the binary segmented images produced by [17], and generates region proposal hypotheses using all possible combinations of 2D connected components ($2N - 1$) present in the binary map (Fig. 8.2b), where N is the number of 2D connected components present in each video frame (Sect. A.3 of [19]). In the following subsection, we briefly introduce the human motion segmentation pipeline.

Human Motion Segmentation. The human motion segmentation algorithm takes as input a sequence of RGB video frames (which contain human action) and outputs binary-labelled space-time video segments where pixels belong to an human action are labelled as foreground and remaining are as background. Firstly, in order to localise and rank “actionness” [4], a human motion saliency feature is computed by exploiting the foreground motion and human appearance information. Foreground motion is estimated by forming a camera model using long term trajectories [3] (Fig. 8.3) and human appearance based saliency map is generated using a DPM person detector [6] (Fig. 8.4a–c) trained on PASCAL VOC 2007 [5]. Secondly, to segment human actions, a hierarchical graph-based video segmentation algorithm [28] is used to extract supervoxels at different level of pixel granularity (i.e. different levels of segmentation hierarchy) (Fig. 8.5). The foreground motion and human appearance based saliency features are then encoded in the hierarchy of supervoxels using a hierarchical Markov Random Field (MRF) model. This encoding gives the unary potential components. To avoid a brittle graph due to a large number of supervoxels [12], the MRF graph is built with a smaller subset of supervoxels which are highly likely to contain human actions. Thus, a candidate edge is built between two neighbouring supervoxels based on their optical flow directions and overlaps with a person detection. In the MRF graph structure, supervoxels are nodes and an edge between two supervoxels are built if: (a) they are temporal neighbours i.e. neighbours in the direction of optical flow, or (b) spatial neighbours, i.e. both the supervoxels have high overlaps with a DPM person detection where the person detection has a confidence greater than a threshold. The

temporal supervoxel neighbours and the appearanceaware spatial neighbours (Fig. 8.4d, e) give the pairwise potential components. To avoid leaks and encourage better semantic information, supervoxels (constrained by appearance and motion cues) from higher levels in the hierarchy (Fig. 8.5) are supported by the higher-order potential. Finally, the energy of the MRF is minimised using the α-expansion algorithm [1, 15] and GMM estimation is used to automatically learn the model parameters. The final outputs of the human motion segmentation are the human foreground background binary maps as depicted in Fig. 8.6.

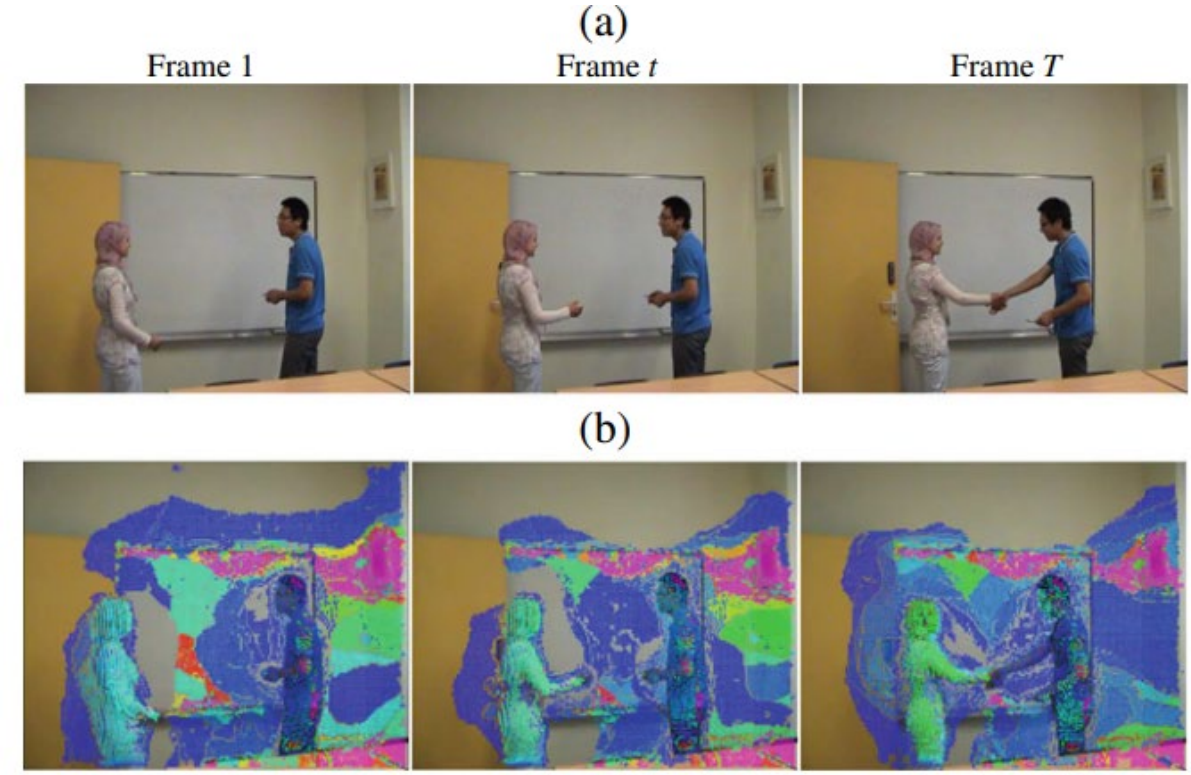


Fig. 8.3 a Three sample input video frames showing a "handshaking" action from a test video clip of LIRIS HARL dataset [26]. **b** The corresponding motion saliency response generated using long term trajectories [3] are shown for these three frames. Notice, the motion saliency is relatively higher for the person at the left, who first enters into the room and then approaches towards the person in the right for "handshaking". Also note that, motion saliency is computed on the entire video clip, for the sake of visualization, we pick three sample frames

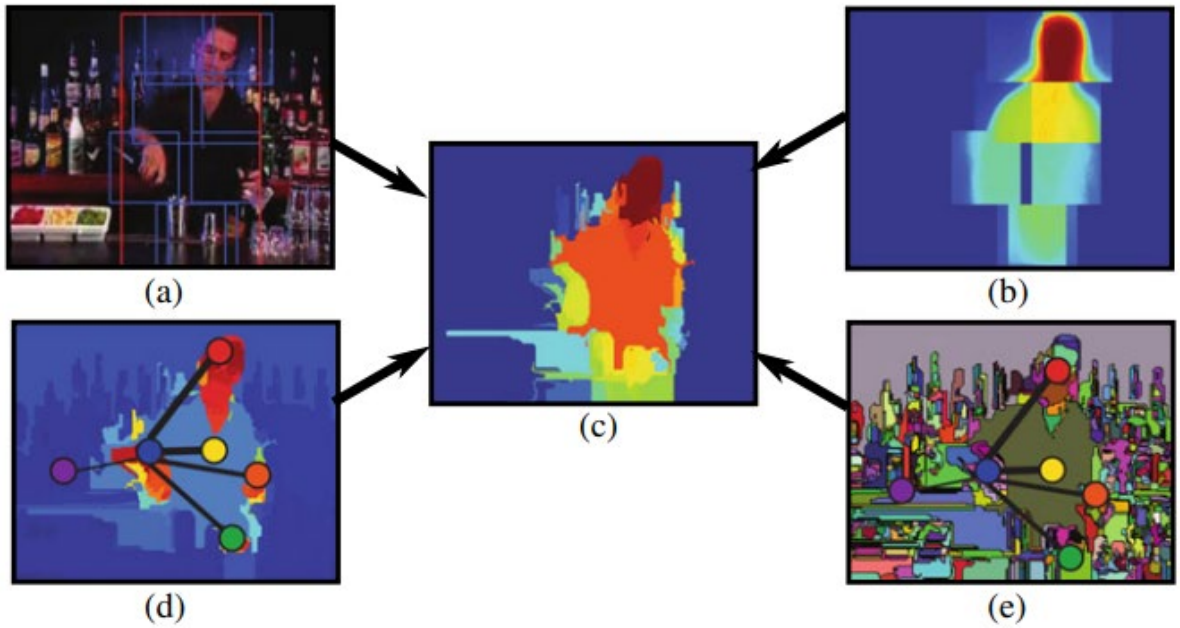


Fig. 8.4 a DPM based person detection. **b** Corresponding DPM part mask. **c** Supervoxel response for the DPM mask. **d** and **e** Pairwise connections of motion saliency map and segmentation respectively. This figure is taken from [17] with author's permission

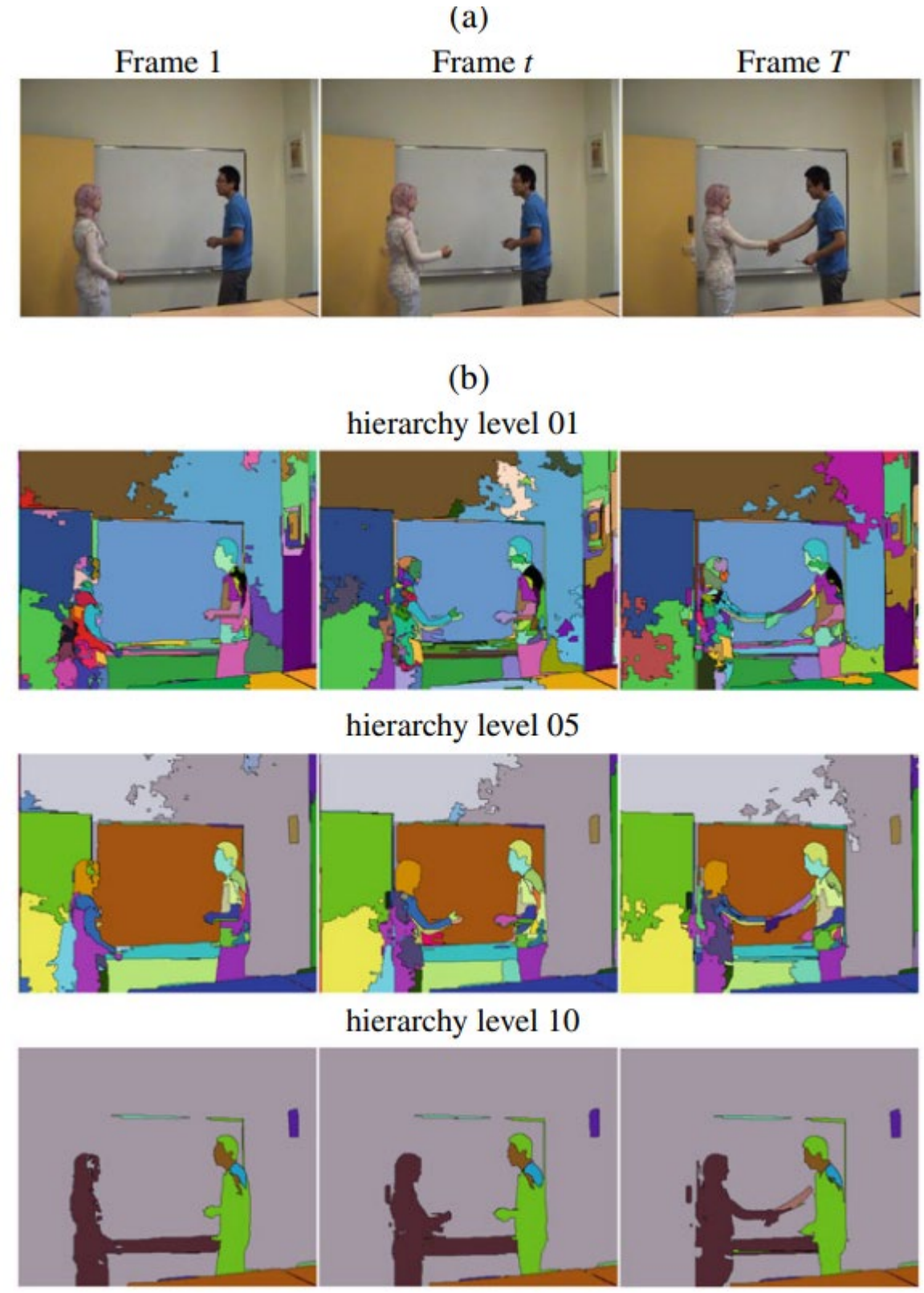


Fig. 8.5 a Three sample input video frames showing a "handshaking" action from a test video clip of LIRIS HARL dataset [26]. **b** The hierarchical graph based video segmentation results (at three different levels of hierarchy) are shown for these three frames. The three rows show segmentation results for hierarchy level 1, 5 and 10 respectively where 1 is the lowest level with supervoxels having smaller spatial extents and 10 is the highest level with supervoxels having relatively larger

spatial extents. Notice, the supervoxels belong to higher levels of segmentation hierarchy tend to preserve the semantic information and are less prone to leaks. Also note that, video segmentation is computed on the entire video clip, for the sake of visualization, we pick three sample frames

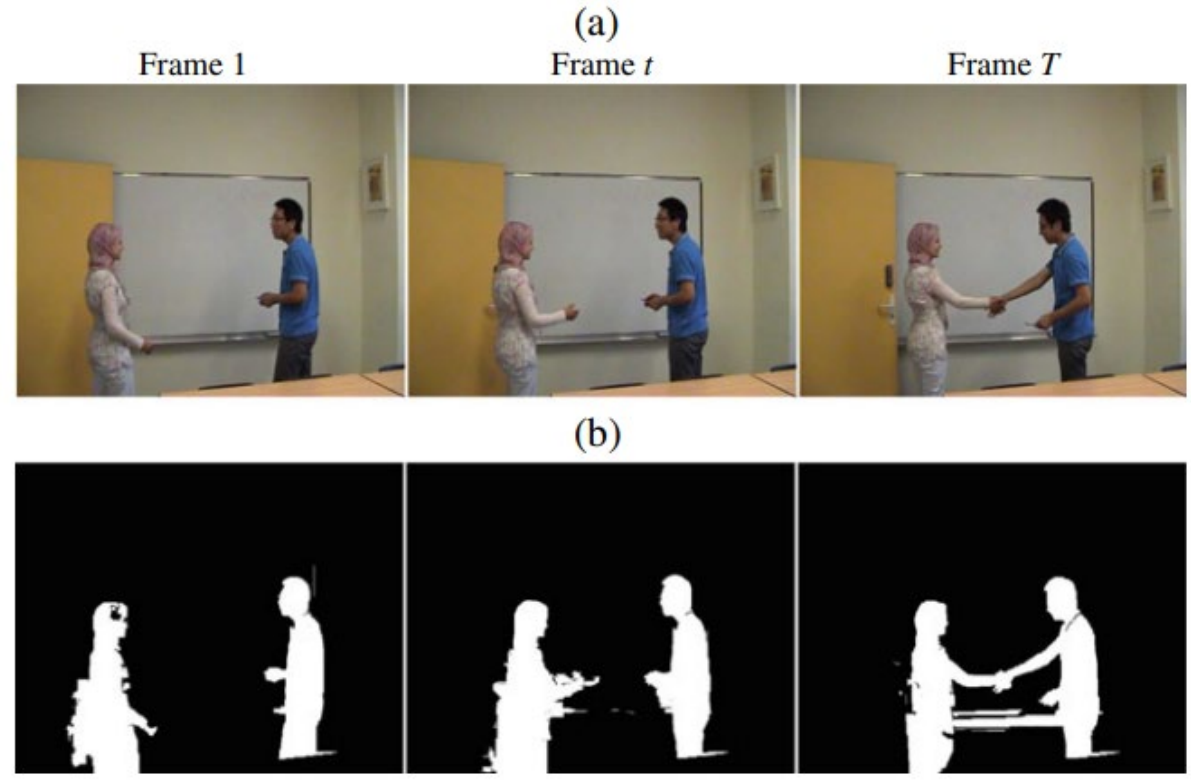


Fig. 8.6 a Three sample input video frames showing a "handshaking" action from a test video clip of LIRIS HARL dataset [26]. **b** The human action foreground-background segmentation results are shown for these three frames

8.3.1.2 Proposal Based on Selective Search

We use two competing approaches to generate region proposals for action detection. The first is based upon Selective Search [24], and the second approach is presented in Sect. 8.3.1.1. Whilst using the Selective Search based method for both training and testing, we only use the motion segmentation based method for testing since it does not provide good negative proposals to use during training. Having a sufficient number of negative examples is crucial to train an effective classifier. At test time, the human motion segmentation (Sect. 8.3.1.1) allows us to extract pixel-level action instance segmentation which is superior to what we may obtain by using Selective Search. We validate our action detection pipeline using both algorithms - the results are discussed in Sect. 8.4.

Measuring "Actionness" of Selective Search Proposals. The selective-search region-merging similarity score is based on a combination of colour (histogram intersection), and size properties, encouraging smaller regions to merge early, and avoid holes in the hierarchical grouping. Selective Search (SS) generates on average 2,000 region proposals per frame, most of which do not contain human activities. In order to rank the proposals with an

"actionness" score and prune irrelevant regions, we compute dense optical flow between each pair of consecutive frames using the state-of-the-art algorithm in [2]. Unlike Gkioxari and Malik [8], we use a relatively smaller motion threshold value to prune SS boxes, (Sect. A.4 of [19]) to avoid neglecting human activities which exhibit minor body movements exhibited in the LIRIS HARL [26] such as "typing on keyboard", "telephone conversation" and "discussion" activities. In addition to pruning region proposals, the 3-channel optical flow values (i.e., flow-x, flow-y and the flow magnitude) are used to construct 'motion images' from which CNN motion features are extracted [8].

8.3.2 Appearance- and Motion-Based Detection Networks

In the second stage of the pipeline, we use the "actionness" ranked region proposals (Sect. 8.3.1) to select image patches from both the RGB (original video frames) and flow images. The image patches are then fed to a pair of fine-tuned Convolutional Neural Networks (Fig. 8.2d) (which encode appearance and local image motion, respectively) from which appearance and motion feature vectors were extracted. As a result the first network learns static appearance information (both lower-level features such as boundary lines, corners, edges and high level features such as object shapes), while the other encodes action dynamics at frame level. The output of the Convolutional Neural Network may be seen as a highly nonlinear transformation(.) from local image patches to a high-dimensional vector space in which discrimination may be performed accurately even by a linear classifier. We follow the AlexNet [16] and [29]'s network architectures.

8.3.2.1 Pretraining

We adopt a CNN training strategy similar to [7]. Indeed, for domain-specific tasks on relatively small scale datasets, such as LIRIS HARL [26], it is important to initialise the CNN weights using a model pre-trained on a larger-scale dataset, in order to avoid over-fitting [8]. Therefore, to encode object "context" we initialise the appearance-based CNN's weights using a model pre-trained on the PASCAL VOC 2012s object detection dataset. To encode typical motion patterns over a temporal window, the optical motion-based CNN is initialised using a model pre-trained on the UCF101 dataset (split 1) [23]. Both appearance- and motion-based pre-trained models are publicly available online at https://github.com/gkioxari/ActionTubes.

8.3.2.2 Fine Tuning

We use deep learning software tool Caffe [14] to fine-tune pretrained domain-specific appearance- and motion-based CNNs on LIRIS HARL training set. For training CNNs, the Selective Search region proposals (Sect. 8.3.1.2) with an IoU overlap score greater than 0.5 with respect to the ground truth bounding box were considered as positive examples, the rest as negative examples. The image patches specified by the pruned region proposals were randomly cropped and horizontally flipped by the Caffe's WindowDataLayer [14] with a crop dimension of 227 × 227 and a flip probability of 0.5 (Fig. 8.2c). Random cropping and flipping were done for both RGB and flow images. The pre-processed image patches along with the associated ground truth action class labels are then passed as inputs to the appearance and motion CNNs to fine-tune (i.e. updating only the weights of the fully

connected layers, in this case, fc6 and fc7 layers, and keeping the weights of the other layers untouched during training) for action classification (Fig. 8.2d). A mini batch of 128 image patches (32 positive and 96 negative examples) are processed by the CNNs at each training forward-pass. Note that the number of batches varies frame-to-fame as per the number of ranked proposals per frame. It makes sense to include fewer positive examples (action regions) as these are relatively rare when compared to background patches (negative examples).

8.3.2.3 Feature Extraction from CNN Layers

We extract the appearance- and motion-based features from the fc7 layer of the the two networks. Thus, we get two feature vectors (each of dimension 4096): appearance feature 'xa = a(r)' and motion feature 'x f = f (r)'. We perform L2 normalisation on the obtained feature vectors, to then, scale and merge appearance and motion features (Fig. 8.2d) in an approach similar to that proposed by [8]. This yields a single feature vector x for each image patch r. Such frame-level region feature vectors are used to train an SVM classifier (Sect. 8.3.3).

8.3.3.1 Class Specific Positive and Negative

Examples In the original RCNN-based one-versus-rest SVM training approach [7], only the ground truth bounding boxes are considered as positive training examples. In contrast, due to extremely high inter- and intra-class variations in LIRIS HARL dataset [26], we use those bounding boxes as positive training examples which have an IoU overlap with the ground truth greater than 75%. In addition, we also consider the ground truth bounding boxes as positives. We believe, our this training data sampling scheme is more intuitive for complex datasets to train SVMs with more positive examples rather than only ground truths. We have achieved almost 5% gain over SVMs classification accuracy with this training strategy. In a similar way, we consider as negative examples only those features vectors whose associated region proposal have an overlap smaller than 30% with respect to the ground truth bounding boxes (possibly several) present in the frame.

8.3.3.2 Training with Hard Negative Mining

We train the set of class specific linear SVMs using hard negative mining [6] to speed up the training process. Namely, in each iteration of the SVM training step we consider only those negative features which fall within the margin of the decision boundary. We use the publicly available toolbox Liblinear [http://www.csie.ntu.edu.tw/~cjlin/liblinear/] or SVM training and use L2 regularizer and L1 hinge-loss with the following parameter values to train the SVMs: positive loss weight WLP = 2; SVM regularisation constant C = 10−3; bias multiplier B = 10.

8.3.4 Testing Region Proposal Classifiers

With our actionness-ranked region proposals $\mathbf{r}_i$ (Sect. 8.3.1) we can extract a cropped image patch and pass it to the CNNs for feature extraction in a similar fashion as described in Sect. 8.3.2.3. A prediction takes the form:

$$\mathbf{S}_c(\mathbf{b}) = \mathbf{w}_c^T \phi(\mathbf{r}) + b_c^{svm},$$

where, $\phi(\mathbf{r}) = \{\phi_a(\mathbf{r}); \phi_f(\mathbf{r})\}$ is combination of appearance and motion features of **r** , $\mathbf{w}_c^T$ and b_c^{svm} are the hyperplane parameter and the bias term of the learned SVM model of class *c*. The confidence measure $\mathbf{s_c}(\mathbf{b})$ that the action '*c*' has happened within the bounding-box region '**b**' is based on the appearance and motion features. Here **b** denotes the associated bounding box for a region proposal **r**.

After SVM prediction,each region proposal '**r**' has been assigned a set of classspecific scores $\mathbf{s}_c$, where *c* denotes the action category label, $c \in \{1,...,C\}$. Once a region proposal has been assigned classification scores $\mathbf{s_c}$, we call it as a detection bounding-box and denote it as **b**. Due to the typically large number of region proposals generated by the Selective Search algorithms (Sect. 8.3.1.2), we further apply non-maximum suppression to prune the regions.

8.3.5 Action Tube Generation and Classification

Once we extract the frame-level detection boxes bt (Sect. 8.3.4) for an entire video, we would like to identify sequences of detections most likely to form action tubes. Thus, to extract final detection tubes, linking of these detection boxes in time is essential to generate tubes. We use our two-pass dynamic programming approach as in [21] to formulate the action tube generation problem as a labelling problem where: (i) we link detections bt into temporally connected action paths for each action, and (ii) we perform a piece-wise constant temporal labelling on the action paths. A detailed formulation of the tube generation problem can be found in the Appendix A.5 [19].

8.4 Experimental Results

We evaluate two region proposal methods with our pipeline, one based on human motion segmentation (HMS) (Sect. 8.3.1.1) and another one based on selective search (SS) (Sect. 8.3.1.2). We will use "HMS" and "SS" abbreviations in tables and plot to show the performance of our pipeline based on each region proposal technique. Our results are also compared to the current state-of-the-art: VPULABUAM-13 [22] and IACAS-51 [11].

8.4.1 Instance Classification Performance—No Localisation (NL)

This evaluation strategy ignores the localisation information (i.e. the bounding boxes) and only focuses on whether an action is present in a video or not. If a video con tains multiple actions then system should return the labels of all the actions present correctly. Even though our action detection framework is not specifically designed for this task, we still outperform the competition, as shown in Table 8.1.

Table 8.1 Quantitative measures precision and recall on LIRIS HARL dataset

Method	Recall	Precision	F1-score
VPULABUAM-13-NL	0.36	*0.66*	0.46
IACAS-51-NL	0.3	0.46	0.36
SS-NL (ours)	*0.5*	0.53	*0.52*
HMS-NL (ours)	*0.5*	0.63	*0.56*
VPULABUAM-13-10%	0.04	0.08	0.05
IACAS-51-NL-10%	0.03	0.04	0.03
SS-10% (ours)	*0.5*	0.53	0.52
HMS-10% (ours)	*0.5*	*0.63*	*0.56*

8.4.2 Detection and Localisation Performance

This evaluation strategy takes localisation (space and time) information into account [27]. We use a 10% threshold quality level for the four thresholds (Sect. 4.2.5 of [19]), which is the same as that used in the LIRIS-HARL competition. In Table 8.1, we denote these results as “method-name-NL” (NL for no localisation) and “methodname-10%”. In both cases (without localisation and with 10% overlap), our method outperforms existing approaches, achieving an improvement from 46% [22] to 56%, in terms of F1 score without localisation measures, and a improvement from 5% [22] to 56% (11.2 times better) gain in the F1-score when 10% localisation information is taken into account. In Table 8.2 we list the results we obtained using the overall integrated performance scores (Sect. 4.2.5 of [19])—our method yields significantly better quantitative and qualitative results with an improvement from 3% [22] to 43% (14.3% times better) in terms of F1 score, a relative gain across the spectrum of measures. Samples of qualitative instance segmentation results are shown in Fig. 8.7.

Table 8.2 Qualitative thresholds and integrated score on LIRIS HARL dataset

Method	I_{sr}	I_{sp}	I_{tr}	I_{tp}	IQ
VPULABUAM-13-IQ	0.02	0.03	0.03	0.03	0.03
IACAS-51-IQ	0.01	0.01	0.03	00.0	0.02
SS-IQ (ours)	0.52	0.22	0.41	0.39	0.38
HMS-IQ (ours)	0.49	0.35	0.46	0.43	*0.44*

The pure classification accuracy of the HMS- and SS-based approaches are reflected in the Confusion Matrices shown in Fig. 8.9. Confusion matrices show the the complexity of the dataset. Some of the actions are wrongly classified, e.g., “telephone-conversation” is classified as “put/take object to/from box/desk”, same can be observed for action “unlock enter/leave room” in SS approach.

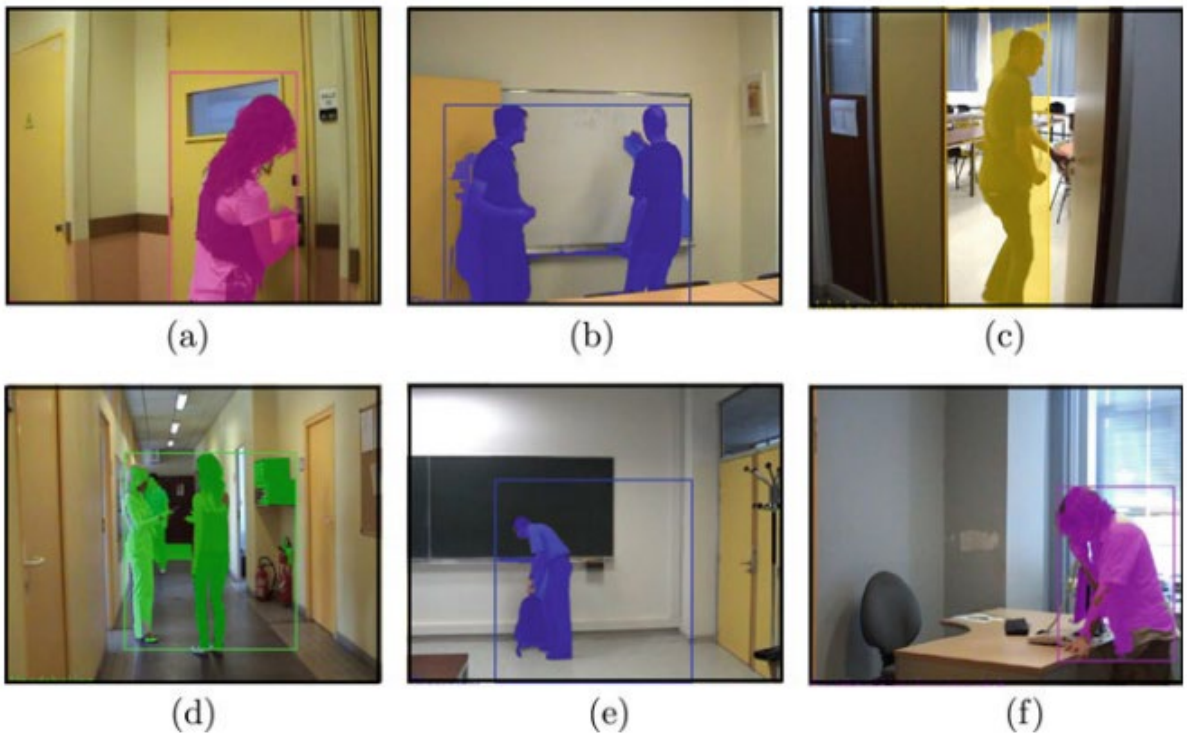


Fig. 8.7 Correct (**a–c**) and incorrect (**d–f**) instance segmentation results on the LIRIS-HARL dataset [26], the correct category is shown in brackets. **a** 'Try enter room unsuccessfully'. **b** 'Discussion'. c 'Unlock enter/leave room'. **d** 'Handshaking' (Give take object from person). **e** 'Discussion' (Leave bag unattended). **f** 'Put take object into/from desk' (Telephone conversation)

8.4.3 Performance Versus Detection Quality Curves

The plots in Fig. 8.8 attest the robustness of our method, as they depict the curves corresponding to precision, recall and F1-score over varying quality thresholds. When the threshold t_{tr} for temporal recall is considered (see Fig. 8.8 plot a) we achieved a highest recall of 50% for both HMS- and SS-based approaches and a highest precision of 65% for HMS-based approach at threshold value of $t_{tr} = 0$. As the threshold increases towards $t_{tr} = 1$, SS-based method shows a robust performance, with highest recall = 50% and precision = 52%, HMS-based method shows promising results with an acceptable drop in precision and recall. Note that when $t_{tr} = 1$, we assume that all frames of an activity instance need to be detected in order for the instance itself to be considered as detected. As for the competing methods, IACAS-51 [11] yields the next competing recall of 2.4% and a precision of 3.7% with a threshold value of $t_{tr} = 1$.

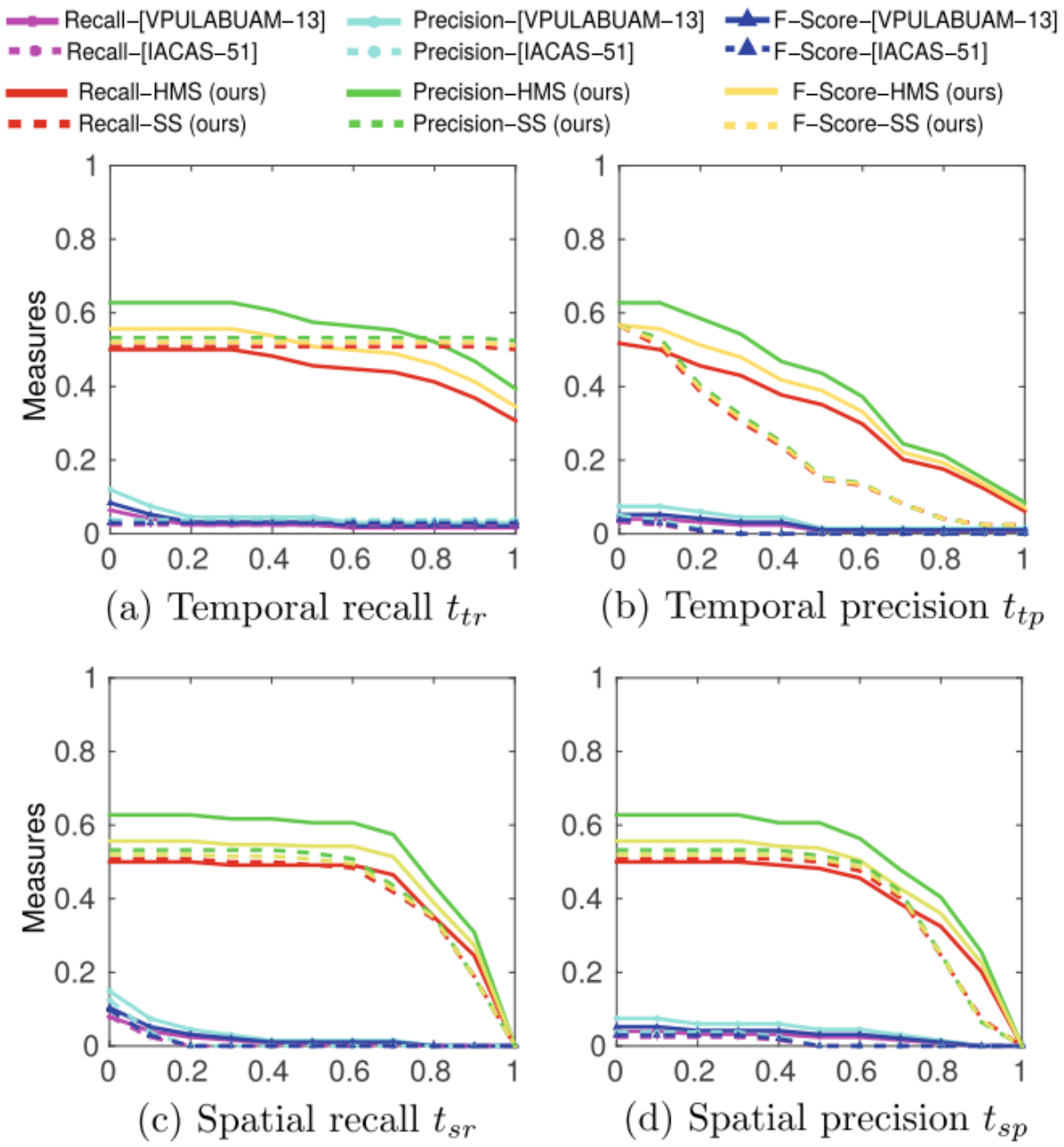
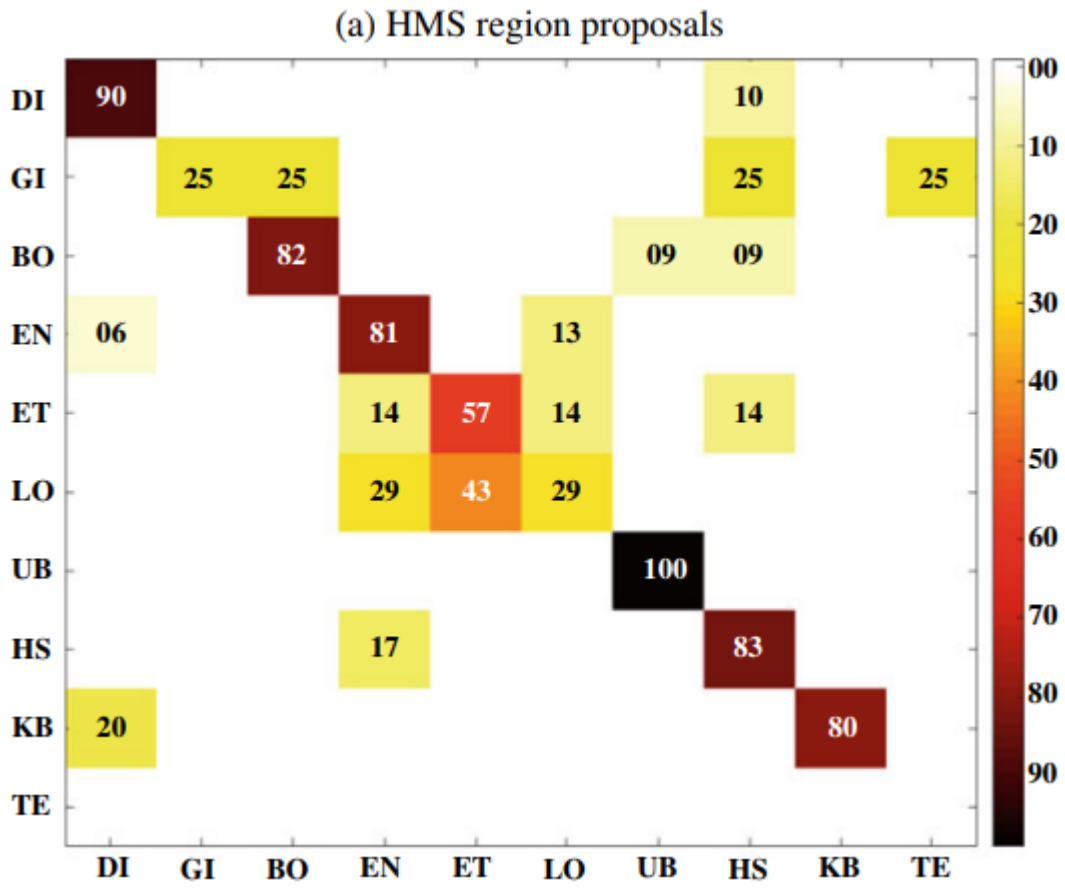


Fig. 8.8 Performance versus detection quality curves

When acting on the value of the temporal frame-wise precision threshold tt p (see Fig. 8.8 plot b) we can observe that at tt p = 1, when we assume that not a single spurious frame outside the ground truth temporal window is allowed, our HMS-based region proposal approach gives highest recall of 8% and precision 10.7%, where, as SSbased approach has significantly lower recall = 2% and precision = 2.4%, which is still significantly higher than the performance of the existing methods. Indeed, at tt p = 1, VPULABUAM-13 has recall = 0.8% and precision = 1% where IACAS-51 yields both zero precision and zero recall. This results tell us that HMS-based approach performs superior in detecting temporal extent of an action and thus is suitable for action localisation in temporally untrimmed videos. The remaining two plots c, d of Fig. 8.8 illustrate the overall performance when spatial overlap is taken into account. Both plots show metrics approaching zero when the corresponding spatial thresholds (pixel-wise recall tsr and pixel-wise precision tsp) approach 1. Note that it is highly unlikely for a ground truth activity to be consistently (spatially) included in the

corresponding detected activity over all the consecutive frames (spatial recall), as indicated in the plot c. It is also rare for a detected activity to be (spatially) included in the corresponding ground truth activity over all the frames (spatial precision) as indicated in plot d. For the pixel-wise recall (plot c), our HMS based method shows consistent recall between 45 and 50% and precision between 59 and 65.5% up to a threshold value of tsr = 0.7, where as, SS-based region proposal approach gives comparable recall between 48.3 and 50.8%, but relative lower precision between 43.5 and 53.2% up to tsr = 0.7. For the pixel-wise precision (plot d), HMS and SS-based approaches give similar recall between 39 and 50%, where as HMS-method again outperforms in precision with 48–63% up to a threshold value of tsp = 0.7, where as SS has precision 41–53% up to a threshold value tsp = 0.7. Finally, we draw conclusion that our HMSbased region proposal approach shows superior qualitative and quantitative detection performance on the challenging LIRIS HARL dataset.



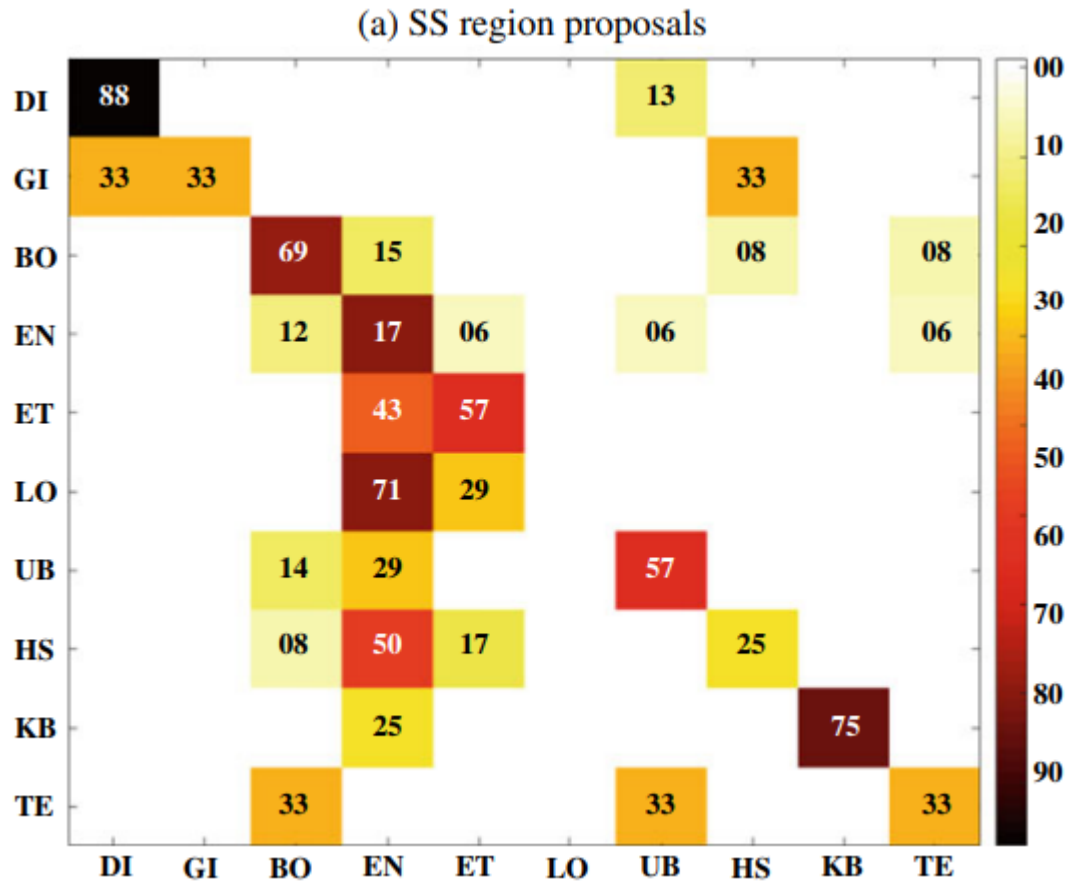


Fig. 8.9 Confusion matrix obtained by human motion segmentation (HMS) and selective search (SS) region proposal approach. They show the classification accuracy of HMS- and SS-based methods on LIRIS HARL human activity dataset. HMS region proposal based method provides better classification accuracy on the the complex LIRIS dataset [26]

8.4.4 Qualitative Action Instance Segmentation and Localisation Results

8.4.4.1 LIRIS HARL Dataset

Figure 8.10 shows additional qualitative action instance segmentation and localisation results on LIRIS HARL dataset [26]. In particular, Fig. 8.10a, d show that the proposed approach can successfully detect action instances belonging to a same class or different classes at finer pixel-level. In (a), two action instances of a single action class (i.e. “typing on keyboard”) are present, whereas in (d) two action instances belonging to two different action classes (i.e. “handshaking” and “leave baggage unattended”) are present.

8.4.4.2 UCF-101-24 Dataset

To demonstrate that the proposed instance segmentation method generalises well on other datasets, we present here some sample instance segmentation results on UCF101-24. We compute the binary segmentation masks for some selected UCF-101-24 test video clips, and apply the bounding-boxes predicted by our proposed action detection model [21] on the top of the binary masks to generate the final instance segmentation results which are

shown in Figs. 8.11 and 8.12. Note that, the proposed approach can successfully localise multiple instances of the “biking” (Fig. 8.11b), “fencing” (Fig. 8.12a), and “ice dancing” (Fig. 8.12c) actions at finer pixel level in space and time.

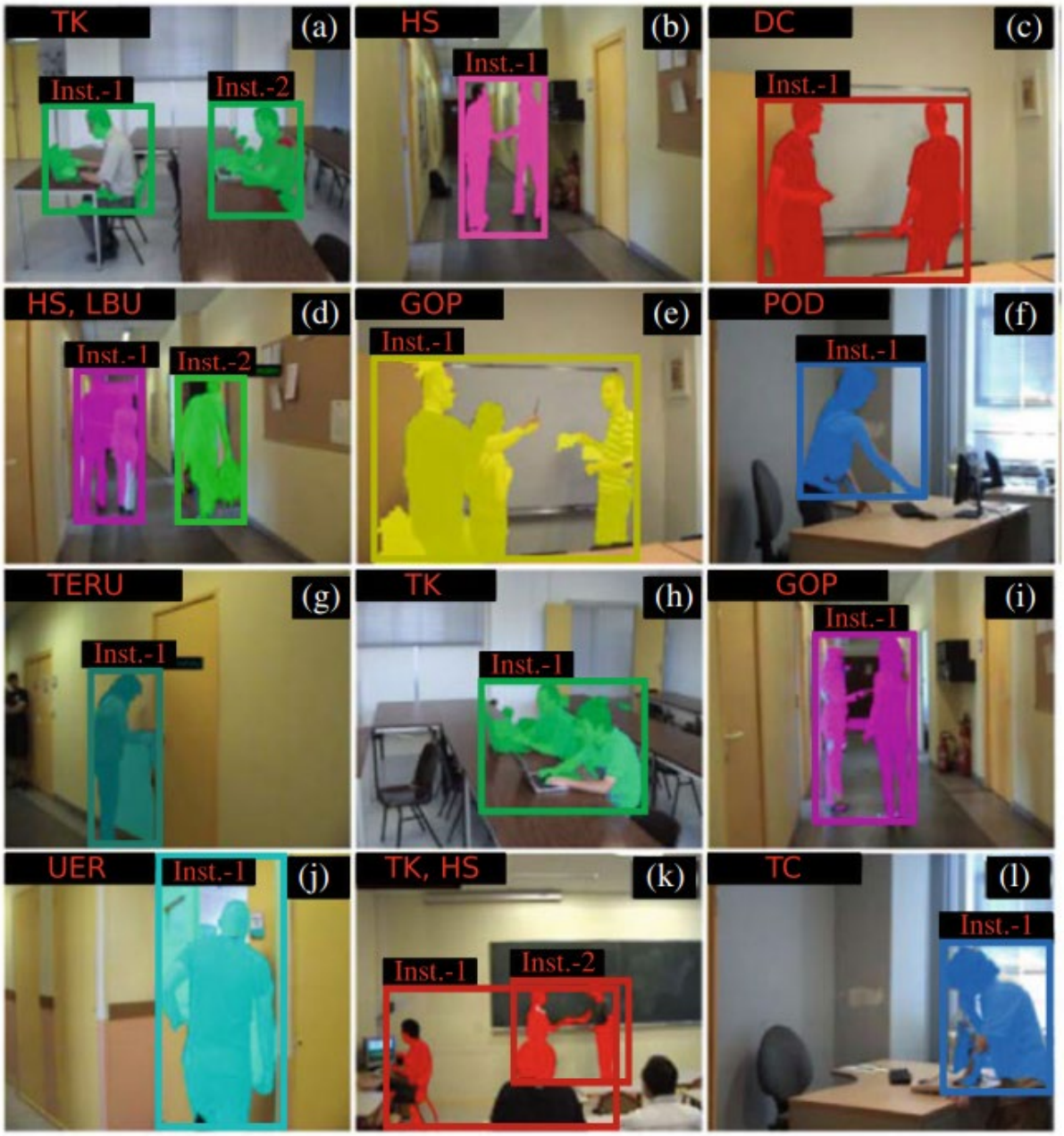


Fig. 8.10 Qualitative action instance segmentation and localisation results on LIRIS HARL dataset. Ground-truth action labels: **TK**—typing on keyboard, **HS**—handshaking, **DC**—discussion, **LBU**— leave baggage unattended, **GOP**—give object to person, **POD**—put object into desk, **TERU**—try enter room unsuccessfully, **UER**—unlock enter room, **TC**—telephone conversation. Correct results: **a, b, c, d, e, f, g, h, j**; incorrect results: **h, i, k, l.** In **h**, out of two instances of **TK** action class, only one instance has been successfully detected. In **i,** the ground truth action class **GOP** has been misclassified as HS class. In **k**, the ground truth action classes **TK** and **HS** have been misclassified as **DC** class. In **l**, the ground truth action class **TC** has been misclassified as **POD** class

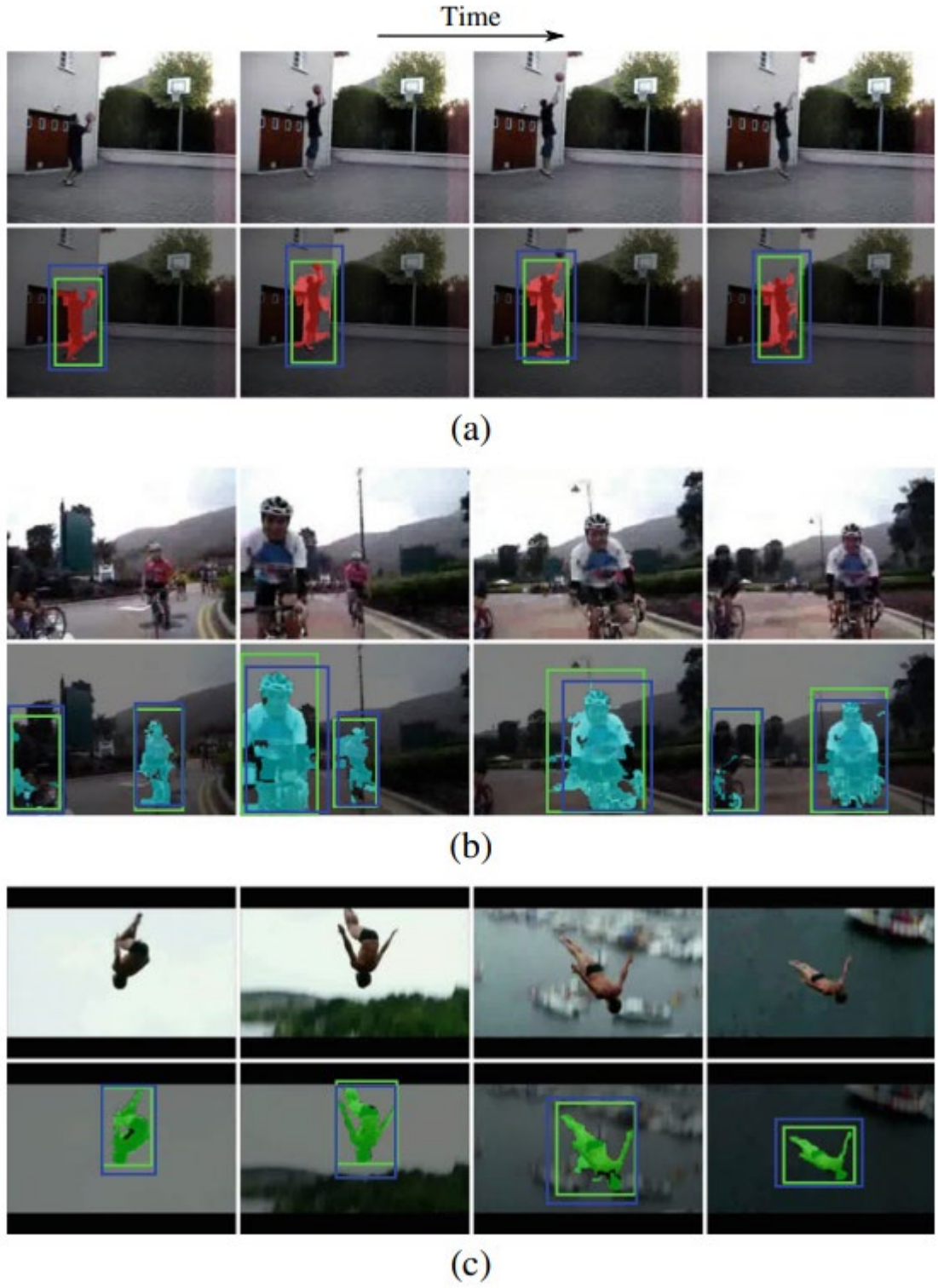


Fig. 8.11 Qualitative action instance segmentation and localisation results on UCF-101-24 test videos. The green boxes represent ground truth annotations, whereas the blue boxes denote the frame-level detections. Each row represents an UCF-101-24 test video clip where the 1st and 2nd rows in each set (i.e. set **a–c**) are the input video frames and their corresponding outputs respectively. From each clip 4 selected frames are shown. Predicted action labels: **a** “basketball”; **b** “biking”; **c** “cliffdiving”

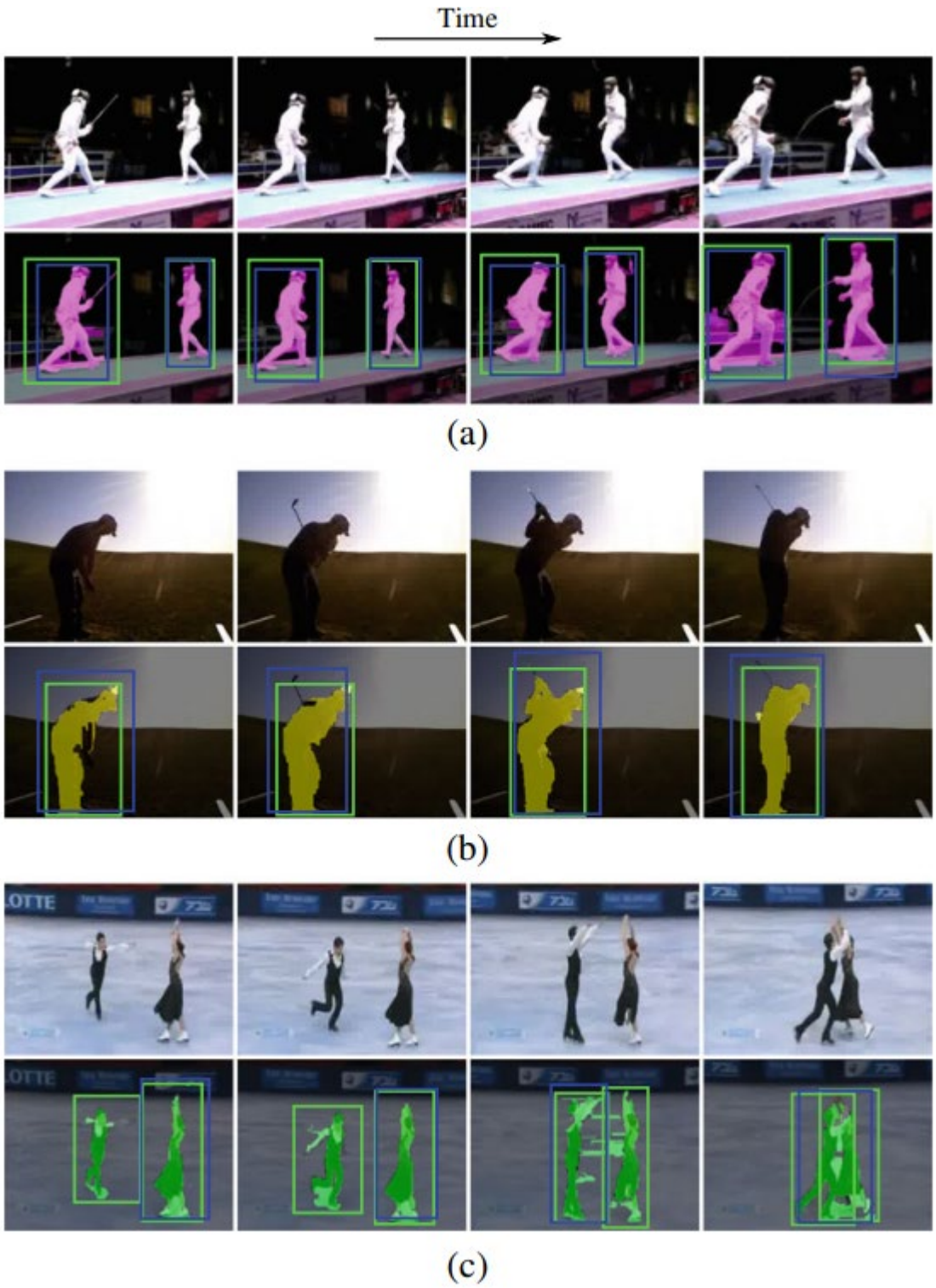


Fig. 8.12 Qualitative action instance segmentation and localisation results on UCF-101-24 test videos. The green boxes represent ground truth annotations, whereas the blue boxes denote the frame-level detections. Each row represents an UCF-101-24 test video clip where the 1st and 2nd rows in each set (i.e. set **a–c**) are the input video frames and their corresponding outputs respectively. From each clip 4 selected frames are shown. Predicted action labels: **a** "fencing"; **b** "golfswing"; **c** "icedancing"

8.5 Discussion

Unlike state-of-the-art supervised instance segmentation approaches (for objects) [9, 10] which require expensive ground-truth segmentation (i.e. per pixel class- and instance-aware labelling) to train their networks, the proposed framework does not require such expensive ground-truth annotations. Thanks to the human action segmentation [17] algorithm which computes human action binary masks using unsupervised learning, thus, does not require expensive ground-truth labels. However, the major drawback of [17] is that it is computationally expensive. For example, it takes several days to compute the binary masks for all frames in LIRIS HARL dataset. Another limitation is that the HMS (human motion segmentation) based region proposals fail to generate accurate bounding box proposals in cases where the action segmentations of two or multiple actors get merged into one 2D connected component, e.g., see Fig. 8.10 (8) in which out of two instances of "typing on keyboard" action class, only one instance has been successfully detected. We empirically found that in such instances Selective Search based region proposals work more effectively. Lastly, as there are no ground truth instance segmentation annotations available for LIRIS HARL and UCF-101-24 datasets, we could not perform an quantitative evaluation of the instance segmentation results. Also note, the J-HMDB-21 dataset has a single action instance per video, and thus, not suitable for evaluating instance segmentation methods.

Acknowledgements

This work was partly supported by ERC grant ERC-2012-AdG 321162- HELIOS, EPSRC grant Seebibyte EP/M013774/1 and EPSRC/MURIgrant EP/N019474/1.

References

1. Boykov, Y., Veksler, O., & Zabih, R. (2001). Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(11), 1222–1239.

2. Brox, T., Bruhn, A., Papenberg, N., &Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. Computer Vision-ECCV, 2004, 25–36.

3. Brox, T., & Malik, J. (2011). Large displacement optical flow: Descriptor matching in variational motion estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(3), 500–513.

4. Chen, W., Xiong, C., Xu, R., & Corso, J. J. (2014). Actionness ranking with lattice conditional ordinal random fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 748–755).

5. Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2007). The PASCAL visual object classes challenge (VOC2007) results. Available at: http://www.pascalnetwork.org/challenges/VOC/voc2007/workshop/index.html.

6. Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9), 1627–1645.

7. Girshick, R., Donahue, J., Darrel, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In IEEE International Conference on Computer Vision and Pattern Recognition.

8. Gkioxari, G., & Malik, J. (2015). Finding action tubes. In IEEE International Conference on Computer Vision and Pattern Recognition.

9. Hariharan, B., Arbeláez, P., Girshick, R., & Malik, J. (2014). Simultaneous detection and segmentation. In European Conference on Computer Vision (pp. 297–312). Springer.

10. He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask r-cnn. In The IEEE International Conference on Computer Vision (ICCV).

11. He, Y., Liu, H., Sui, W., Xiang, S., & Pan, C. (2012). Liris harl competition participant. Institute of Automation, Chinese Academy of Sciences, Beijing. http://liris.cnrs.fr/harl2012/results.html.

12. Jain, S. D., & Grauman, K. (2014). Supervoxel-consistent foreground propagation in video. In European Conference on Computer Vision (pp. 656–671). Springer.

13. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., & Black, M. J. (2013). Towards understanding action recognition. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) (pp. 3192–3199).

14. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R. B., et al. (2014). Caffe: Convolutional architecture for fast feature embedding. http://arxiv.org/abs/1408.5093.

15. Kohli, P., Torr, P. H., et al. (2009). Robust higher order potentials for enforcing label consistency. International Journal of Computer Vision, 82(3), 302–324.

16. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 1097–1105.

17. Lu, J., Xu, R., & Corso, J. J. (2015). Human action segmentation with hierarchical supervoxel consistency. In IEEE International Conference on Computer Vision and Pattern Recognition.

18. Peng, X., & Schmid, C. (2016). Multi-region two-stream r-cnn for action detection. In European Conference on Computer Vision (pp. 744–759). Springer.

19. Saha, S. Spatio-temporal human action detection and instance segmentation in videos. Ph.D. thesis. Available at: https://tinyurl.com/y4py79cn.

20. Saha, S., Singh, G., Sapienza, M., Torr, P. H., & Cuzzolin, F. (2017). Spatio-temporal human action localisation and instance segmentation in temporally untrimmed videos. arXiv:1707.07213.

21. Saha, S., Singh, G., Sapienza, M., Torr, P. H. S., & Cuzzolin, F. (2016). Deep learning for detecting multiple space-time action tubes in videos. In British Machine Vision Conference.

22. SanMiguel, J. C., & Suja, S. (2012). Liris harl competition participant. Video Processing and Understanding Lab, Universidad Autonoma of Madrid, Spain, http://liris.cnrs.fr/harl2012/ results.html.

23. Soomro, K., Zamir, A. R., & Shah, M. (2012). UCF101: A dataset of 101 human action classes from videos in the wild. Technical Report, CRCV-TR-12-01.

24. Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. International Journal of Computer Vision, 104(2), 154–171.

25. Weinzaepfel, P., Harchaoui, Z., & Schmid, C. (2015). Learning to track for spatio-temporal action localization. IEEE International Conference on Computer Vision and Pattern Recognition.

26. Wolf, C., Mille, J., Lombardi, E., Celiktutan, O., Jiu, M., Baccouche, M., et al. The LIRIS Human activities dataset and the ICPR 2012 human activities recognition and localization competition. Technical Report, LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/École Centrale de Lyon (2012). http://liris.cnrs.fr/ publis/?id=5498.

27. Wolf, C., Mille, J., Lombardi, E., Celiktutan, O., Jiu, M., Dogan, E., et al. (2014). Evaluation of video activity localizations integrating quality and quantity measurements. Computer Vision and Image Understanding, 127, 14–30.

28. Xu, C., Xiong, C., & Corso, J. J. (2012). Streaming hierarchical video segmentation. In European Conference on Computer Vision (pp. 626–639). Springer.

29. Zeiler, M. D., & Fergus, R. (2013). Visualizing and understanding convolutional networks.