

Doctoral Dissertation

**Actor-identified Spatiotemporal
Action Detection —
Detecting Who Is Doing What in Videos**

Fan Yang

March 3, 2021

Graduate School of Science and Technology
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Thesis Committee:

Professor Satoshi Nakamura	(Supervisor, NAIST)
Professor Takeo Kanade	(Carnegie Mellon University)
Professor Kiyoshi Kiyokawa	(Co-supervisor, NAIST)
Associate Professor Sakriani Sakti	(Co-supervisor, NAIST)
Senior Lecturer Yang Wu	(Co-supervisor, Kyoto University)

Actor-identified Spatiotemporal Action Detection — Detecting Who Is Doing What in Videos*

Fan Yang

Abstract

The success of deep learning on video Action Recognition (AR) has motivated researchers to progressively promote related tasks from the coarse level to the fine-grained level. Compared with conventional AR that only predicts an action label for the entire video, Temporal Action Detection (TAD) has been investigated for estimating the start and end time for each action in videos. Taking TAD a step further, Spatiotemporal Action Detection (SAD) has been studied for localizing the action both spatially and temporally in videos. However, who performs the action, is generally ignored in SAD, while identifying the actor could also be important. To this end, we propose a novel task, Actor-identified Spatiotemporal Action Detection (ASAD), to bridge the gap between SAD and actor identification.

In ASAD, we not only detect the spatiotemporal boundary for instance-level action but also assign the unique ID to each actor. To approach ASAD, Multiple Object Tracking (MOT) and Action Classification (AC) are two fundamental elements. By using MOT, the spatiotemporal boundary of each actor is obtained and assigned to a unique actor identity. By using AC, the action class is estimated within the corresponding spatiotemporal boundary. Since ASAD is a new task, it poses many new challenges that cannot be addressed by existing methods: i) no dataset is specifically created for ASAD, ii) no evaluation metrics are designed for ASAD, iii) current MOT performance is the bottleneck to obtain satisfactory

*Doctoral Dissertation, Graduate School of Science and Technology, Nara Institute of Science and Technology, March 3, 2021.

ASAD results. To address those problems, we contribute to i) annotate a new ASAD dataset, ii) propose ASAD evaluation metrics by considering multi-label actions and actor identification, iii) improve the data association strategies in MOT to boost the MOT performance, which leads to better ASAD results. We believe considering actor identification with spatiotemporal action detection could promote the research on video understanding and beyond.

Keywords:

Object Identification, Multiple Object Tracking, Action Recognition, Computer Vision

Contents

1	Introduction	1
2	Related Works	5
2.1.	Video Action Recognition	5
2.2.	Multiple Object Tracking	8
2.3.	Action Classification	11
3	Proposed ASAD Dataset and Evaluation Metrics	13
3.1.	Dataset for ASAD	13
3.2.	Evaluation Metrics for ASAD	16
3.3.	Spatial Detection Evaluation	17
3.4.	Actor Identification Evaluation	19
3.5.	Multi-label Action Classification Evaluation	20
4	Addressing the Bottleneck of ASAD by Improving Multiple Object Tracking	23
4.1.	Overview	23
4.2.	Methodology	27
4.2.1	The Appearance Encoder	27
4.2.2	The Online Data Association	29
4.2.3	Offline Data Association	33
4.3.	Experiments on MOT Datasets	37
4.3.1	Implementation Details	37
4.3.2	Experimental Datasets	38
4.3.3	Evaluation Metrics	39

4.3.4	Online Data Association Evaluation	39
4.3.5	Offline Data Association Evaluation	43
4.3.6	Evaluation with Oracle Detection	45
4.4.	Discussion	47
5	ASAD Benchmark	49
5.1.	ASAD Framework	49
5.2.	ASAD Experiments	50
5.3.	Discussion	53
6	Conclusions and Future Work	55
6.1.	Conclusions	55
6.2.	Future Works	57
6.2.1	Short-term Future Work	57
6.2.2	Middle-term Future Work	59
6.2.3	Long-term Future Work	60
	Acknowledgement	61
	References	62
	Publication List	90
	Appendix	92
A.	92
A.1	Appendix Overview	92
A.2	Hand Pose/Skeleton Tracking	93
A.3	Skeleton-based Action Classification	107
A.4	3D Multiple Object Tracking	119
A.5	Actor-specified Spatiotemporal Action Detection (ASAD) for 4K-resolution Aerial Videos	129

List of Figures

1.1	Actor-identified Spatiotemporal Action Detection (ASAD) is Spatiotemporal Action Detection (SAD) plus actor identification. Part of this figure is from https://www.pinterest.jp/pin/130745195408112697/ .	2
1.2	The illustration of ASAD processing. Part of this figure is from https://www.pinterest.jp/pin/130745195408112697/ .	3
2.1	A comparison of action recognition works, which could be roughly divided into four categories: Action Recognition (AR), Temporal Action Detection (TAD), Spatiotemporal Action Detection (SAD), and our defined Actor-identified Spatiotemporal Action Detection (ASAD). Existing works (<i>i.e.</i>, AR, TAD, and SAD) ignore to identify actors while our ASAD addresses this issue. Parts of this graph credit to [1].	6
2.2	Categories of Action Classification (AC) models.	12
3.1	Comparison between SAD and ASAD spatiotemporal annotation by using UCF101+ROAD [2] as an example. The annotation in ASAD should complete the the spatiotemporal boundary for each actor in the entire video, no matter if the defined action is finished or not.	14
3.2	Comparison between SAD and ASAD actor ID annotation by using AVA [3] as an example. In a single video, while the existing SAD dataset may assign multiple actor IDs to the same actor, our ASAD assigns the unique actor ID the actor.	15
3.3	We create a new ASAD dataset based on existing AVA dataset [4], by assigning the unique actor identity to each actor.	16

3.4	Illustration of our Actor-identified AVA dataset.	17
3.5	A historical timeline overview of datasets intended for video action recognition studies.	18
3.6	Overview of our ASAD metrics, which evaluate the performance of spatial detection, action classification, and actor identification.	18
3.7	Illustration of matching pair between the ground-truth and the predicted samples.	21
3.8	An example of calculating HL@0.5. Only the first case with IoU=0.52 is considered as a positively detected sample, and therefore the overall HL@0.5=0.5.	22
4.1	A demonstration that objects could be tracked fully by the appearance feature but partially by the motion feature. ① The ideal scenarios that both red and blue observations can be correctly associated by using either the motion feature or the appearance feature. ② Due to the faster motion, red and blue observations obtain incorrect motion initialization by only using the motion feature. ③ Due to the unpredictable movement of the camera, blue and red observations obtain incorrect motion initialization and then lose tracking by only using the motion feature.	24
4.2	Appearance Encoder to learn appearance features with identities and classes. GAP and BN respectively represent Gap Average Pooling and Batch Normalization. Batched-normalized Embedding Features are selected as the appearance features for data association.	28

4.3	Illustration of our online data association. For the initialization, the previous-frame location is extended to the adjustable matching window by adding the maximum shifting distance obtained from the statistics of labeled data (Eq. 4.6). After the initialization, a rectangle, that bounds the expanded previous-frame box and the estimated current-frame box, is used as the adjustable matching window (Eq. 4.7). Only observations, covered by the adjustable matching window, are considered for data association by using appearance features, which makes a trade-off between excluding impossible matching candidates and including potential ones. . . .	31
4.4	The illustration of constructing inputs for offline self-supervised learning. Within the same video, we only gather triplets from temporally overlapped tracklets, as $\Pi_p \cap \Pi_q \neq \emptyset$ for pseudo tracklet T_p and T_q . In a mini-batch of input, the ratio between samples of labeled videos and unlabelled target videos is 1 : 1.	34
4.5	The histograms of cosine similarity for intra-frame and intra-tracklet observations. By approximating the histogram as a normal distribution, the boundary that maximally separates two histograms is selected as the offline appearance threshold $\theta_{offline}^{app}$, which minimizes the sum of the false positives and negatives of cosine similarity distributions..	35
4.6	The Hierarchical Clustering is accommodated to associate short-term tracklets into long-term tracklets. The cutting threshold is obtained from the statistical information of splitting tracklets. . .	36
4.7	Qualitative results of ReID-dominated data association. We performed online approach on BDD100K MOT dataset and offline approach on others. Red arrows indicate the identical instance, which shows that the targets are tracked robustly in diverse scenarios.	48
5.1	Overview of the basic ASAD framework.	50
5.2	Visualization of actor identification results in the A-AVA dataset (1/2). The identical actor is located by bounding boxes of the same color.	52

5.3	Visualization of actor identification results in the A-AVA dataset (2/2). The identical actor is located by bounding boxes of the same color.	53
5.4	The difference of motion consistency in static camera recording videos and non-static camera recording videos.	54
6.1	Roadmap to Actor-identified Spatiotemporal Action Detection (ASAD).	58
6.2	Integrating 3D MOT, fine-grained hand action detection, skeleton-based action classification, and drone-recorded videos in our ASAD framework.	59
A.1	Integrating 3D MOT, fine-grained hand action detection, skeleton-based action classification, and drone-recorded videos in our ASAD framework.	92
A.2	Examples of hand location in depth images, where the target hand is inside the red box. In the egocentric view case, the hand is not the nearest object to depth cameras, so that it cannot be separated from the background through depth threshold.	94
A.3	Normalizing hand scale after segmentation.	95
A.4	Obtaining hand segmentation ground truth by using gloves.	95
A.5	The architecture of SPS-Net. The \odot represents element-wise multiplication. H_i^d and H_i^t are soft proposals generated at frame i	96
A.6	The details of each components in SPS-Net. “2D Res-Bolock_64” denotes an 2D residual residual convolutional block with kernel size 64. “Max pool, /2” means maxpooling operation with stride = 2. The \oplus denotes element-wise summation.	97
A.1	Qualitative results of hand segmentation on the NYU Hand Dataset.	104
A.2	Qualitative results of 3D hand pose tracking on the Hand2017 Challenge.	105
A.3	Concerned skeleton sequence properties.	107

A.4	The network architecture of DD-Net. “ $2 \times \text{CNN}(3, 2^* \text{filters}), /2$ ” denotes two 1D ConvNet layers (kernel size = 3, channels = 2^*filters) and a Maxpooling (strides = 2). Other ConvNet layers are defined in the same format. GAP denotes Global Average Pooling. FC denotes Fully Connected Layers (or Dense Layers). We can change the model size by modifying <i>filters</i>	110
A.5	An example of Joint Collection Distances (JCD) feature at frame k , where the number of joints is N	111
A.6	Confusion matrix of SHREC dataset (14 hand actions) obtained by DD-Net.	117
A.7	Confusion matrix of SHREC dataset (28 hand actions) obtained by DD-Net.	118
A.8	Our framework for 3D panoramic multi-person localization and tracking.	120
A.9	Overview of our ASAD framework.	130
A.10	Illustration of our room-in detection.	132
A.11	Applicable model structures for ASAD when only RGB data is used. L denotes the number of frames used in the model. i) and ii) represent different models that share the same structure at the beginning.	133
A.12	Architecture of the proposed framework. Given each 4K-resolution aerial image of size 2160×3840 , C-RPN is utilized to select patches (608×608) that might contain actors. Based on selected patches, normal detectors are used to generate fine-grained bounding boxes for each actor. After that, fine-grained bounding boxes are further connected to be spatio-temporal tubes by a MOT algorithm. Next, we sample L frames from spatio-temporal tubes and obtain their corresponding 2D CNN features. STAM then takes 2D CNN features to generate attention maps that focus on target actors. In the end, the concatenation of 2D CNN features and their multiplication with attention maps, are used to estimate multi-label action classes by a 3D ConvNet.	135

A.13	The demonstration of generating patches by C-RPN. The downscale factor $s_1 \approx 1/4$, and the down-sampling factor $s_2 = 1/8$.	138
A.14	Visualizations of optical flow maps generated by PWC-Net [5], using Okutama-action Dataset. Due to the tiny size of actor and the drone camera movement, it is challenging to obtain actor motion information from the optical flow.	139
A.15	An illustration of proposed Attention Action Classification Network (AACN), with its Spatio-temporal Attention Module (STAM). Three frames are used in this illustration, where $\{x_{L-2}^n, x_{L-1}^n, x_L^n\}$ are RGB features sampled by Algorithm 3, and they are fed to 2D ConvNet to generate 2D CNN features $\{f_{L-2}^n, f_{L-1}^n, f_L^n\}$. STAM takes stacking 2D CNN features to obtain corresponding attention maps $\{a_{L-2}^n, a_{L-1}^n, a_L^n\}$. The multiplication results of 2D CNN features and attention maps are concatenated with 2D CNN features again, and then be used to estimate multi-label actions by a 3D ConvNet.	140
A.16	Patch proposals in Okutama-action testing sets, which are generated by C-RPN. Generated peak points are marked by red, and patches are enclosed by colorful rectangles. The first row shows three sequential frames (<i>i.e.</i> , 300, 400 and 500) in video 1.2.3. The second row shows three sequential frames (<i>i.e.</i> , 250, 300 and 350) in video 2.2.1. To efficiently cover target actors, clusters automatically merge and split, based on the relative distance within peak points.	143
A.17	Visualizations of dense map estimation and patch generation on VisDrone testing dataset (1/3). The size of above aerial images is 1080×1920 and the patch size is 608×608 . Proposal patches are bounded by colorful rectangles.	145
A.18	Visualizations of dense map estimation and patch generation on VisDrone testing dataset (2/3). The size of above aerial images is 1080×1920 and the patch size is 608×608 . Proposal patches are bounded by colorful rectangles.	146

A.19	Visualizations of dense map estimation and patch generation on VisDrone testing dataset (3/3). The size of above aerial images is 1080×1920 and the patch size is 608×608 . Proposal patches are bounded by colorful rectangles.	147
A.20	Examples of multi-label action classification results in our framework.	148
A.21	Visualization of attentions for the target actor. We assume that the target actor consistently appears in his/her spatio-temporal tube while others may not. The attention mask is learned in an unsupervised manner.	150

List of Tables

2.1	The related datasets and studies for AR, TAD, SAD, and ASAD. Note that, unlike other SAD datasets, actor ID is given in annotations of Okutama [6] but the ASAD evaluation has not been explored. Besides, the Okutama dataset consists of 4K-resolution drone videos, which may only cover very limited scenarios of ASAD. In addition, some SAD models, such as ROAD [2], AlphAction [7], and ACAM [8], may potentially generate ASAD results but were evaluated by the SAD protocol in the original works. That is, the consistency of actor identity is ignored in these works.	7
2.2	The role of MOT and AC in AR, TAD, SAD, and ASAD. For some evaluation protocols of SAD, there is no need to link detection to tubes and MOT may not be used.	8
2.3	A comparison of SAD, Actor Identification, and ASAD.	8
2.4	Corresponding datasets of single-class and multi-class MOT. Although MOT15-17 datasets are annotated with multi-class MOT labels, they are generally evaluated in single-class MOT protocol.	9
2.5	Online and offline data association works	10
2.6	Data association approaches.	10
2.7	The properties of action classification works	12
4.1	The top-3 results for BDD100K MOT Challenge of CVPR 2020 Workshop on Autonomous Driving (June-13th-2020).	39

4.2	Ablation studies for our online approach on BDD100K MOT Dataset. The identical detection and appearance encoder are utilized in each approach. In each column, Red and Blue represent the first and second results, respectively.	42
4.3	CVPR 2020 MOTS Track 1 Challenge (May 30th, 2020).	43
4.4	The ablation studies on MOTS20 testing set.	44
4.5	Compared with state-of-the-art methods on MOT15-17 testing sets with private detection, where $\uparrow(\downarrow)$ indicates that the larger(smaller) the value is, the better the performance.	45
4.6	KITTI-MOTS Results. Red and Blue represent the first and second results, respectively.	46
4.7	Evaluation performance on MOT15-17 train sets. All methods use the oracle detection of MOT15-17 train sets. The appearance encoder is trained on Market-1501 dataset [9].	46
5.1	Results of the default ASAD-adapted frameworks on our A-AVA dataset by using our ASAD evaluation metrics, where $\uparrow(\downarrow)$ indicates that the larger(smaller) the value is, the better the performance.	50
5.2	Comparison of using different MOT module in ADAD frameworks. We utilize our A-AVA dataset and ASAD evaluation metrics, where $\uparrow(\downarrow)$ indicates that the larger(smaller) the value is, the better the performance.	51
5.3	Results of actor identification on our A-AVA dataset with oracle actor detection, where $\uparrow(\downarrow)$ indicates that the larger(smaller) the value is, the better the performance.	52
A.1	Properties of experimental datasets.	101
A.2	Hand segmentation results on NYU Hand Dataset.	103
A.3	Hand segmentation results on CVAR Dataset.	103
A.4	The top-4 results of the Hand2017 Challenge - 3D Hand Pose Tracking Task. ALL denotes ADE of all joints; SEEN denotes ADE over visible joints; UNSEEN denotes ADE over occluded joints.	103
A.5	Properties of experimental datasets	114
A.6	Results on SHREC (Using 3D skeletons only)	115

A.7	Results on JHMDB (Using 2D skeletons only)	116
A.8	Properties of experimental datasets.	126
A.9	Monocular-camera-based localization precision on KITTI Dataset. If distance from predicted locations to ground-truth location is within a threshold, it is correctly predicted.	127
A.10	3D MOT Benchmark. $\uparrow(\downarrow)$ indicates that the larger(smaller) the value is, the better the performance. Multiple Object Tracking Accuracy (MOTA) is the dominant criterion. The details of the evaluation metrics were previously explained in [10].	127
A.11	Comparing using features of the whole body and the upper body on the MPLT dataset. We evaluate localization and tracking per- formance within 10 m of the coordinate center.	127
A.12	Actor spatial detection performance on Okutama-action dataset. The symbol $\uparrow(\downarrow)$ indicates that the larger(smaller) the value, the better the performance.	142
A.13	Multi-label ASAD results. Action Identification perfor- mance will be separately evaluated. The symbol $\uparrow(\downarrow)$ in- dicates that the larger(smaller) the value, the better the perfor- mance. Only RGB data is used in this test. Note, we choose $L = 5$ for Better-AVA due to computation memory limitation and it has to be an odd number. While other models utilize $L = 8$ since in- stantaneous actions are defined in ASAD. Except for Better-AVA, other action detection models use bounding boxes that are gener- ated by C-RPN + YOLOv3-tiny, which achieves $AP@0.5=85.2$.	149
A.14	Evaluation performance on actor identification by referring se- lected MOT metrics.	150

Nomenclature

Acronyms

AR: Action Recognition.

TAD: Temporal Action Detection.

SAD: Spatiotemporal Action Detection.

ASAD: Actor-identified Spatiotemporal Action Detection.

MOT: Multiple Object Tracking.

AC: and Action Classification.

CNN: Convolutional Neural Network.

RNN: Recurrent Neural Network.

RoI: Region of Interest.

GAP: Global Average Pooling.

FC layer: Fully Connected layer.

IoU: Intersection over Union.

Chapter 1

Introduction

Vision-based Action Recognition (AR) aims to detect human-defined actions from a sequence of data (*e.g.*, videos) and has a wide range of applications in our daily life. For instance, it has been applied for YouTube to recognize billions of video tags before recommending a video to us, or for the policemen to quickly retrieval a criminal from thousands-hours surveillance videos, or for a virtual game machine to interact with players, and many others.

In recent years, the success of deep learning on AR has motivated researchers to progressively promote the AR task from the coarse level to the fine-grained level. Compared with conventional AR that only predicts an action label for the entire video, Temporal Action Detection (TAD) has been investigated for estimating the start and end time for each action in videos. Taking TAD a step further, Spatiotemporal Action Detection (SAD) has been studied for localizing the action both spatially and temporally in videos. However, who performs the action is generally ignored in SAD studies. **We believe the actor identification should be considered together with SAD.** When multiple actors are involved in the target scenes (*e.g.*, basketball/soccer games), it is preferred to know “who is doing what”, and thus, identifying each actor with their actions is desired. Nonetheless, SAD and actor identification are treated as different tasks for a long time. To this end, we propose a novel task, Actor-identified Spatiotemporal Action Detection (ASAD), to bridge the gap between SAD and actor identification (Figure 1.1).

To approach ASAD, Multiple Object Tracking (MOT) [11] and Action Classi-

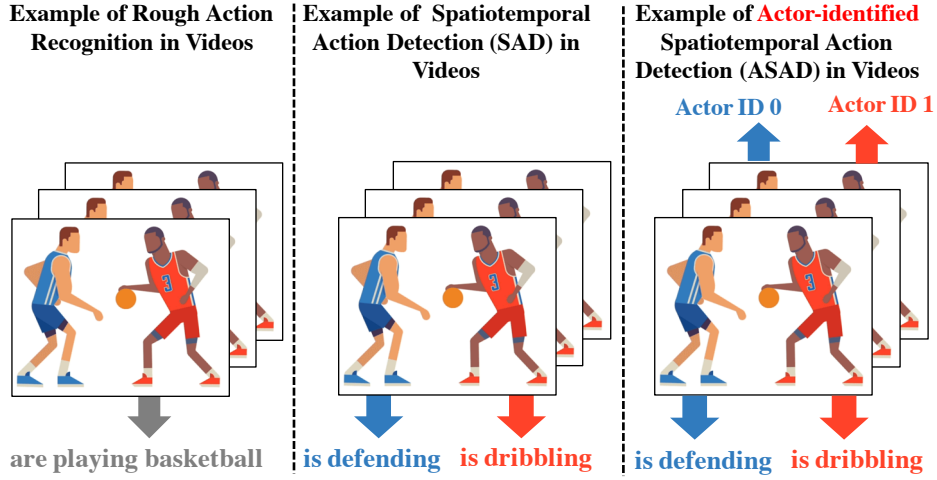


Figure 1.1: Actor-identified Spatiotemporal Action Detection (ASAD) is Spatiotemporal Action Detection (SAD) plus actor identification. Part of this figure is from <https://www.pinterest.jp/pin/130745195408112697/>.

fication (AC) [1] are two fundamental elements (Figure 1.2). By using MOT, the spatiotemporal boundary of each actor is obtained and assigned to a unique actor identity. By using AC, the action class is estimated within the corresponding spatiotemporal boundary. In general, they may work as independent modules by considering the model training flexibility.

Since ASAD is a new task, it poses many new challenges that cannot be addressed by existing methods: i) no dataset is specifically created for ASAD, ii) no evaluation metrics are designed for ASAD, iii) current MOT performance could be the bottleneck to obtain satisfactory ASAD results. To address those problems, we contribute to i) annotate a new ASAD dataset, ii) propose ASAD evaluation metrics by considering multi-label actions and actor identification, iii) improve the data association strategies in MOT to boost the MOT performance, which leads to better ASAD results.

We summarize the main contributions of this thesis as follows.

- We raise a new study task of video action recognition — Actor-identified Spatiotemporal Action Detection (ASAD). As far as we are aware, it has a great importance but has been historically overlooked. ASAD bridges the gap between the existing Spatiotemporal Action Detection (SAD) study

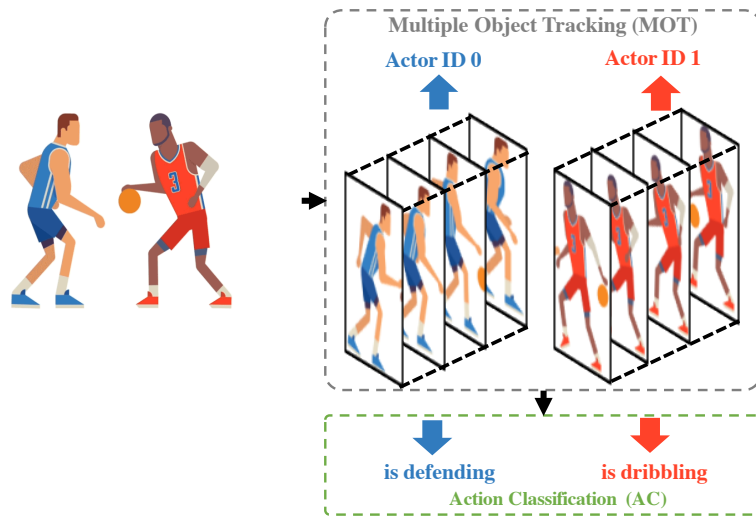


Figure 1.2: The illustration of ASAD processing. Part of this figure is from <https://www.pinterest.jp/pin/130745195408112697/>.

and the new demand for identifying actors.

- We specifically provided a novel dataset for ASAD study. It covers a rich action category and actor identities.
- We presented novel metrics for ASAD evaluation. To the best of our knowledge, existing metrics cannot be applied to ASAD, and we are the first to introduce such metrics.
- Since MOT is the main bottleneck for improving the ASAD performance, we proposed a new method to boost the MOT performance and therefore can promote the ASAD performance. We also demonstrate that our methods can achieve state-of-the-art results on other MOT datasets.

In the following chapters, we first review the literature on video action recognition, multiple object tracking (MOT), and action recognition (AC). In Chapter 2, by comparing with previous research, we explain the importance of proposing ASAD task. Then we introduce our efforts on constructing ASAD dataset and metrics. In Chapter 3, we introduce a new ASAD dataset — Actor-identified AVA (A-AVA), in where the spatiotemporal boundaries, actor identities, and

corresponding actions are all annotated. Besides, we propose ASAD metrics to evaluate all aspects of ASAD outputs. Next, we present our efforts on improving the MOT algorithm. In Chapter 4, we show how to improve the MOT performance by taking the appearance feature as the dominant factor. We demonstrate the effectiveness of our MOT solution on multiple MOT datasets. After that, we introduce the ASAD framework and experiments. In Chapter 5, we perform experiments on our A-AVA dataset and evaluate the results by our ASAD metrics. We prove that our MOT solution can improve actor identification performance and consequently obtain better ASAD results. In the end, we talk about several external works about action recognition that may be integrated into the ASAD task in the future (see Appendix).

Chapter 2

Related Works

2.1. Video Action Recognition

In general, video action recognition research can be divided into several categories (Figure 2.1). Normal Action Recognition (AR) takes an entire video, or, a video clip, as the input and generates a corresponding action class. It is used to understand the overall video concept without specifying the details in the spatial domain and temporal domain. Temporal Action Detection (TAD) gives temporal details to AR, by clarifying the start and end time of an action. Accordingly, one video could be segmented into several temporal components in TAD. Compared with TAD, Spatiotemporal Action Detection (SAD) not only detects the action boundary in the temporal domain but also locates the actor with bounding boxes (or instance masks) in the spatial domain. We generally call such a spatiotemporal boundary the action tube. In this work, we propose Actor-identified Spatiotemporal Action Detection (ASAD) from SAD, by incorporating the unique identity of each actor.

We summarize the related datasets and studies for AR, TAD, SAD, and ASAD in Table 2.1. To link bounding boxes to action tubes, Multiple Object Tracking (MOT) is also commonly applied in SAD. Some SAD works can also track the actor and assign them with unique IDs. However, **based on the evaluation protocol of SAD, the annotation of actor identity may not be provided and the actor identification has not been evaluated.** That means, there is no clear boundary between ASAD and SAD in terms of the method, their

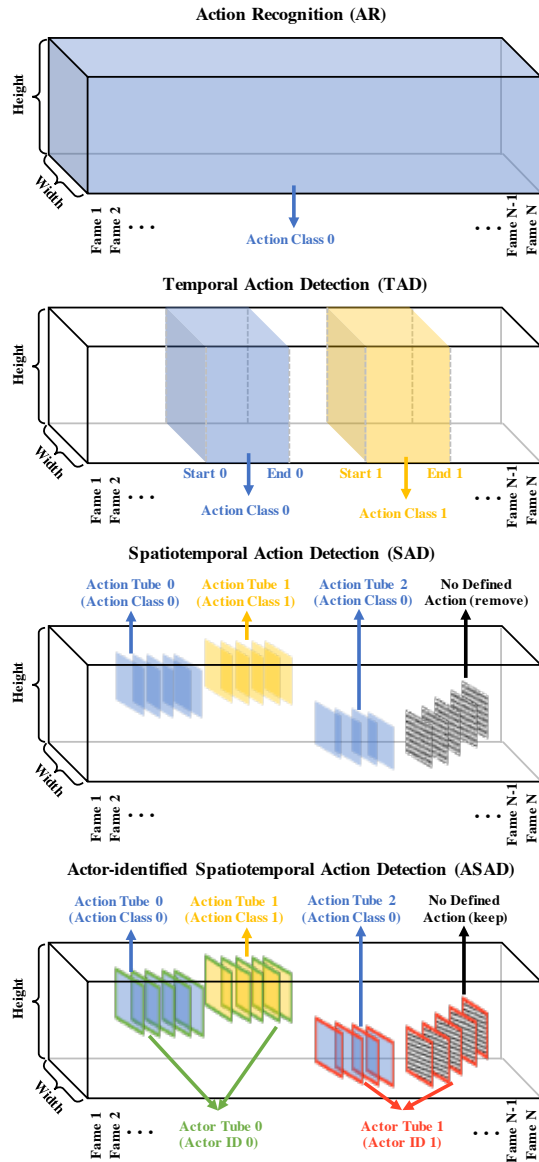


Figure 2.1: A comparison of action recognition works, which could be roughly divided into four categories: Action Recognition (AR), Temporal Action Detection (TAD), Spatiotemporal Action Detection (SAD), and our defined Actor-identified Spatiotemporal Action Detection (ASAD). **Existing works (*i.e.*, AR, TAD, and SAD) ignore to identify actors while our ASAD addresses this issue.** Parts of this graph credit to [1].

Action Recognition Category	Available Datasets	Related Works
AR	HMDB [12], UCF101 [13], Sports-1M [14], Kinetics-700 [15]	[16, 17, 18, 19]
TAD	ActivityNet [20], YouTube-8M [21], THUMOS [22], HACS [23]	[24, 25, 26, 27, 28, 29]
SAD	UCF101+ROAD [2], DALY [30], Hollywood2Tubes [31] AVA [3], AVA-Kinetics [32], ActEV [33]	[34, 35, 36, 37, 2, 38, 39, 40, 41, 8, 42, 43, 44, 7]
ASAD	Okutama [6] (available but not ideal)	Ours

Table 2.1: The related datasets and studies for AR, TAD, SAD, and ASAD. Note that, unlike other SAD datasets, actor ID is given in annotations of Okutama [6] but the ASAD evaluation has not been explored. Besides, the Okutama dataset consists of 4K-resolution drone videos, which may only cover very limited scenarios of ASAD. In addition, **some SAD models, such as ROAD [2], AlphAction [7], and ACAM [8], may potentially generate ASAD results but were evaluated by the SAD protocol in the original works. That is, the consistency of actor identity is ignored in these works.**

difference more lies in the data annotation and evaluation protocols. In detail, the action tube ID given in SAD may not be consistent with actor ID. For example, after the same actor changes his/her action, the corresponding action tube ID changed but the actor ID should remain the same. Unfortunately, such kind of actor ID is not available in most SAD datasets.

As we suppose that MOT and AC are two important components in ASAD, we take a look into the role of MOT and AC in AR, TAD, SAD, and ASAD (Table 2.2). The AC could be a necessary module for all action recognition categories. In SAD, MOT might be used (*e.g.*, on UCF101+ROAD dataset [2]), but not be necessary (*e.g.*, on AVA dataset [3]). However, both MOT and AC are needed in ASAD.

In addition, previous studies [45, 46, 47] focus on only identifying actors in videos, but without detecting their actions. In this manner, as a new task, ASAD has bridged the gap between the SAD and the actor identification (Table 2.3).

Action Recognition Category	Using MOT	Using AC
Action Recognition (AR)	Not Need	Need
Temporal Action Detection (TAD)	Not Need	Need
Spatiotemporal Action Detection (SAD)	May Not Need	Need
Actor-identified Spatiotemporal Action Detection (ASAD)	Need	Need

Table 2.2: The role of MOT and AC in AR, TAD, SAD, and ASAD. For some evaluation protocols of SAD, there is no need to link detection to tubes and MOT may not be used.

Approaches	Identifying Actors	Detecting Actions
Spatiotemporal Action Detection (SAD) [34, 35, 36, 37, 2, 38, 39, 40, 42, 43, 44, 7]	✗	✓
Actor Identification [45, 46, 47]	✓	✗
Actor-identified Spatiotemporal Action Detection (ASAD)	✓	✓

Table 2.3: A comparison of SAD, Actor Identification, and ASAD.

2.2. Multiple Object Tracking

Since Multiple Object Tracking (MOT) plays an important role in Actor-identified Spatiotemporal Action Detection (ASAD), we further provide an overview of MOT related works.

Depending on the number of object classes, we divide MOT tasks into the single-class MOT task and the multi-class MOT task. Single-class MOT, as the name suggests, only contains a unique class for the target object (*e.g.*, pedestrian) in one video. By contract, multi-class MOT involves multiple object classes (*e.g.*, pedestrian, bike, and car) per video. Compared with multi-class MOT, single-class MOT is favored by numerous MOT studies [11, 48] because it offers a relatively simple experimental environment. Recently, researchers have drawn attention to multi-class MOT due to the increased demand from real applications (*e.g.*, autonomous driving [49]). To conduct data association, the main difference between single-class and multi-class MOT lies in how to utilize the category labels: while only identity labels are considered in single-class MOT, category

labels could be leveraged in multi-class MOT.

	Corresponding Datasets
Single-class MOT	MOT15-17 Datasets [50, 51] & MOTS20 dataset [52]
Multi-class MOT	KITTI-MOTS Dataset [53, 52] & BDD100K MOT Dataset [49]

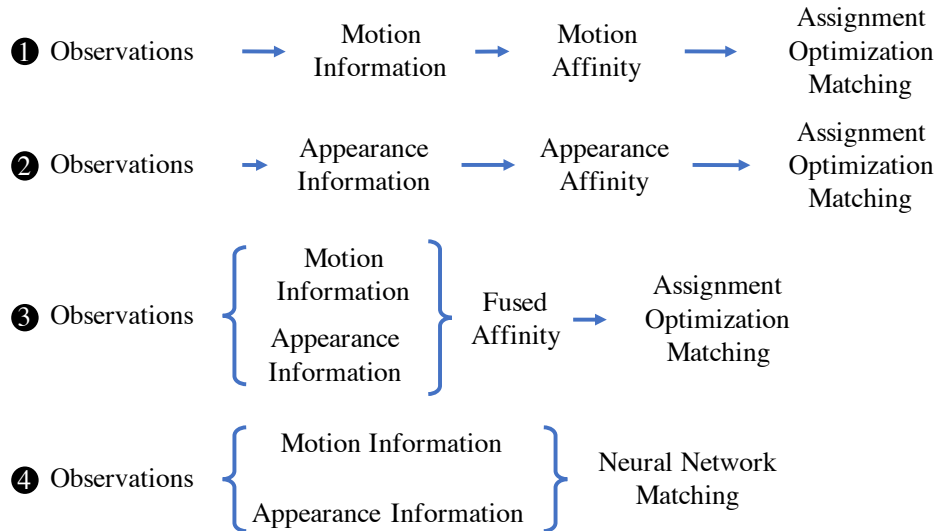
Table 2.4: Corresponding datasets of single-class and multi-class MOT. Although MOT15-17 datasets are annotated with multi-class MOT labels, they are generally evaluated in single-class MOT protocol.

Despite multi-class MOT and single-class MOT having the aforementioned difference, they essentially seek for associating identical observations crossing frames. We attempt to investigate their data association from online and offline perspectives. Related works are listed in Table 2.5. The online data association is performed on observations that are available up to the current frame. Typically, linear assignment algorithms [54, 55] are applied to associate consecutive-frame observations through a bipartite graph matching [56, 57]. Different from online approaches, offline data association takes global observations into consideration, which may not be applicable for real-time applications but be ideal for assisting MOT annotation works. Numerous offline approaches have been proposed in previous studies [11]. Among them, formulating MOT data association as a global clustering problem has achieved great successes since decades ago [58, 59, 60, 61, 62, 63]. We applied Hierarchical Clustering (HC) for global clustering in our offline approach. However, for the main challenge of applying HC in MOT, as determining the sensitive cutting threshold, [59, 64, 63] has to heuristically determine it.

Concerning how the motion feature and the appearance feature are utilized, we further group existing data association approaches (online & offline) into four categories, which are illustrated in Table 2.6. Particularly, approach **1** only applies the motion feature to calculate the motion affinity and then match cross-frame observations. Those approaches either applying the Kalman filter (*e.g.*, [66]) or training a neural network model (*e.g.*, [79]) to estimate the object mo-

Approaches	Online Data Association	Offline Data Association
[65, 66, 56, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80]	✓	✗
[81, 82, 83, 61, 62, 84, 85]	✗	✓

Table 2.5: Online and offline data association works



(a) The mechanism of using motion features and appearance features for data association.

	Approaches	Demand for Tracklet Annotation	Feature Fusion	Including Other ReID Datasets for Training
①	[66, 67, 79, 80]	Not Used / High	N/A	N/A
②	[61]	Medium / Low	N/A	Easy
③	[82, 83, 56, 68, 62, 73, 78]	Medium / Low	Difficult	Easy
④	[71, 72, 74, 75, 76, 77]	High	Easy	Difficult

(b) The properties of listed data association approaches.

Table 2.6: Data association approaches.

tion. **While the former case does not need any tracklet annotation, the later case heavily relies on the tracklet label to train the motion estimation network.** As opposed to ❶, ❷ represents a bunch of appearance-based methods. Generally, they adopt the appearance encoder from the object Re-identification task (*e.g.*, [86]) to learn strong discriminative appearance features for data association.

To train the appearance encoder, the tracklet annotation is preferred but not mandatory. Since the appearance encoder may work as an independent component, other ReID datasets (*e.g.*, [9]) can be used in the training. Approach ❸ is a combination of ❶ and ❷, which fuses the motion feature and the appearance feature for data association. However, it is non-trivial to make a trade-off between the motion feature and the appearance feature in diverse scenarios. Some defects, such as failing for fast movement, may exist in existing approaches (*e.g.*, [56]), and thus improved solutions are desired. Approach ❹ leverages neural networks to directly learn a data association strategy in an end-to-end manner. Consequently, the hand-craft feature fusion is avoided, but this, in turn, makes ❹ can only be trained with adequate annotated tracklet labels, which increases the annotation burdens. Besides, the trajectories, though, are smooth in the labeled data, heavy jitters could exist in the testing data due to the detection noise. This inconsistency arises from the generalization challenge for ❹, as unsatisfactory results might be generated when the detection quality is poor. Our approach leverages the procedure of ❸, which may require further exploration of the fusion strategy. But in compensation, it reduces the demand for annotated tracklets and also improves the model generalization by incorporating the priory knowledge to spatiotemporal constraints.

2.3. Action Classification

The Action Classification (AC) model plays such a role to map the spatiotemporal information to action categories. There are numerous AC studies considering the approaches of utilizing features and designing the model structure. In detail, Action Classification (AC) approaches could be divided into 5 categories, including RGB AC, RGB + Flow AC, Pose AC, RGB + Pose AC, and RGB + Flow

+ Pose AC, as shown in Figure 2.2. Based on these 5 categories, we list the corresponding studies in Table 2.7.

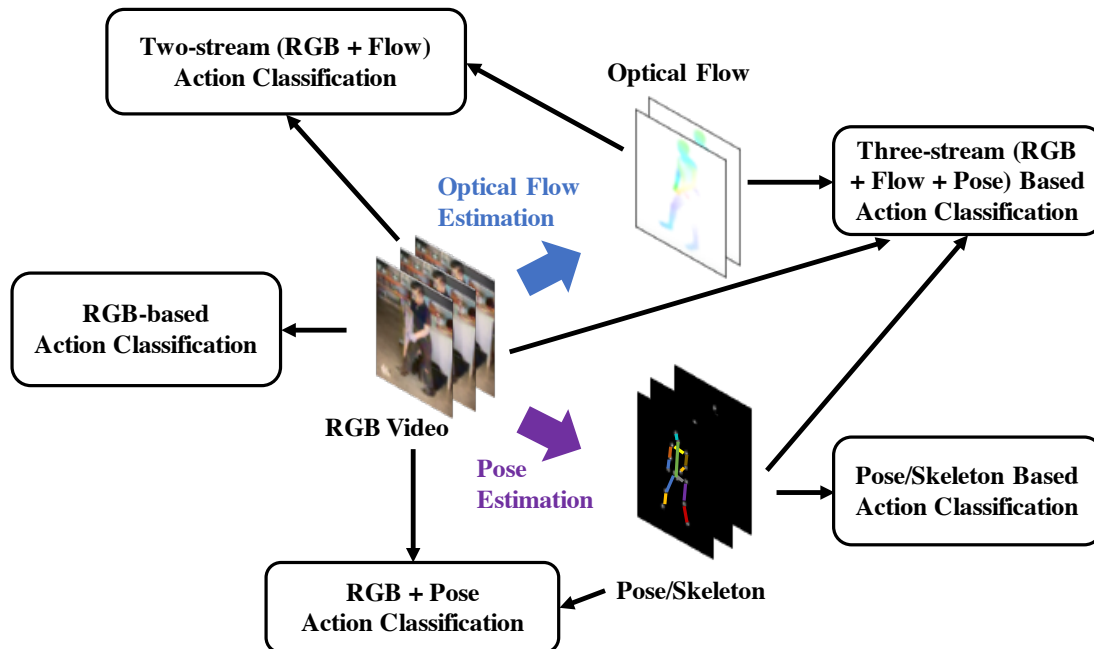


Figure 2.2: Categories of Action Classification (AC) models.

Approaches	RGB AC	RGB + Flow AC	Pose AC	RGB + Pose AC	RGB + Flow + Pose AC
Large-scale AC [14]	✓	✗	✗	✗	✗
Two-Stream [16]	✗	✓	✗	✗	✗
C3D [17], I3D [87], ECO [88], P3D [89], FastSlow [19]	✓	✓	✗	✗	✗
HCN [90], 2s-AGCN [91], DD-Net [92]	✗	✗	✓	✗	✗
Potion [93], PA3D [94]	✗	✗	✗	✓	✗
Chained AC [95]	✗	✗	✗	✗	✓

Table 2.7: The properties of action classification works

Chapter 3

Proposed ASAD Dataset and Evaluation Metrics

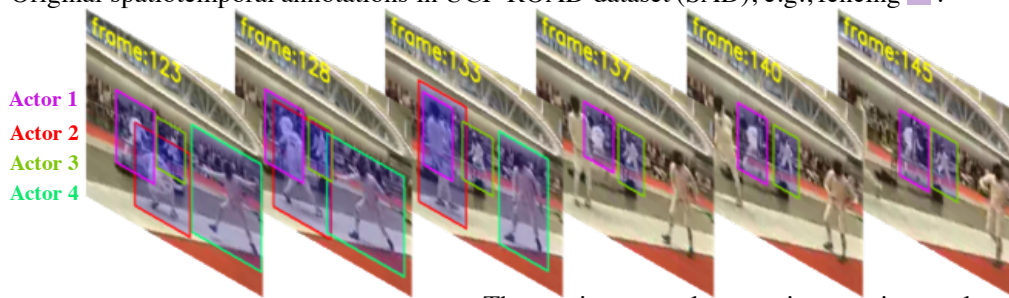
Given a video, Actor-identified Spatiotemporal Action Detection (ASAD) aims to detect the spatiotemporal boundaries (*i.e.*, tracklets/actor tubes) for each actor, assign each actor a unique identity, and obtain the actions of actors at each moment. Consequently, the ASAD dataset should include those factors and the ASAD metrics should verify the performance on those factors. We organized our proposed dataset and evaluation metrics scripts at GitHub <https://github.com/fandulu/ASAD>.

3.1. Dataset for ASAD


By reviewing existing action recognition datasets (Chapter 2), it can be noticed that a proper ASAD dataset may not be available. Although the existing Spatiotemporal Action Detection (SAD) dataset might be similar to our desired ASAD dataset, the actor identity is not properly annotated in the SAD dataset. We illustrate the annotation difference between SAD and ASAD data annotation by using UCF101+ROAD dataset [2] and AVA dataset [3] as examples (Figures 3.1 and 3.2). In the UCF101+ROAD dataset, the spatiotemporal boundaries are incomplete. Since actor identification is not the concern in SAD, after the predefined action is finished, the spatiotemporal annotation is not available. In contrast, the annotation in ASAD should complete the spatiotemporal boundary

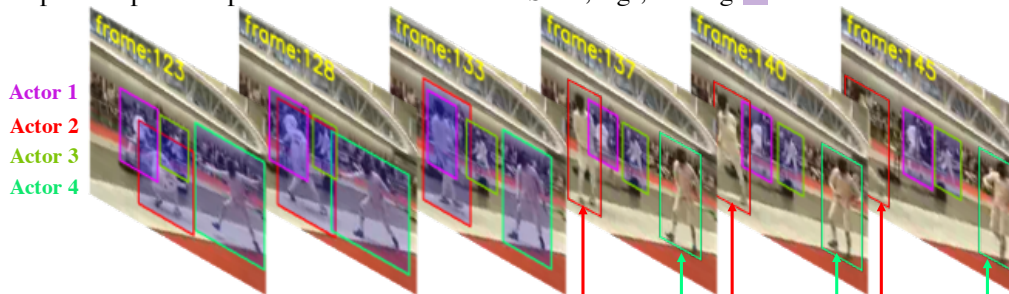
for each actor in the entire video, no matter if the defined action is finished or not. In the AVA dataset, despite the actor IDs being given, multiple actor IDs have been assigned to the same actor in a single video, which is incorrect for actor identification. For actor identification purposes, the unique actor ID should be assigned to each actor in one piece of video. Besides, while some remote surveillance video datasets equip with spatiotemporal boundaries, actor identities, and action classes, they focus on the special scene (*e.g.*, remote surveillance) and may not be suitable for the general ASAD study. For example, Okutama dataset [6] and PANDA [96] only record tiny scale actors and covers a small group of human daily activities.

Original spatiotemporal annotations in UCF-ROAD dataset (SAD), *e.g.*, fencing .



The spatiotemporal annotations are incomplete.

Expected spatiotemporal annotations for our ASAD, *e.g.*, fencing .



The spatiotemporal annotations are complete.

Figure 3.1: Comparison between SAD and ASAD spatiotemporal annotation by using UCF101+ROAD [2] as an example. The annotation in ASAD should complete the the spatiotemporal boundary for each actor in the entire video, no matter if the defined action is finished or not.

Due to the above reasons, we are motivated to annotate a new ASAD dataset. Compared with the SAD dataset, the ASAD dataset requires correct actor iden-

Actor ID annotation in the original AVA dataset (SAD). Actor IDs are fragmented in one video.



Expected actor ID annotation in our ASAD. Actor IDs are consistent in one video.



Figure 3.2: Comparison between SAD and ASAD actor ID annotation by using AVA [3] as an example. In a single video, while the existing SAD dataset may assign multiple actor IDs to the same actor, our ASAD assigns the unique actor ID the actor.

tities. As the AVA dataset [3] is a canonical SAD dataset and TAO dataset [97] offers some actor identity annotations on it, we create an ASAD dataset based on them (Figure 3.3). Due to the heavy annotation cost, among 430 AVA video clips, we only selected 77 of them to make our new dataset. Note that, we mainly selected video clips that have visible actors and multiple actors available. We named our ASAD dataset A-AVA, which represents the Actor-identified AVA dataset. A-AVA dataset contains 47 videos for training and 30 videos for testing. Be the same as the AVA dataset, there are 80 action categories in the A-AVA dataset, and, every 25 frames (*i.e.*, around 1 second), the annotation is given once. In the A-AVA dataset, the spatiotemporal boundaries, actor identities, and corresponding actions are all annotated. More examples are illustrated in Figure 3.4.

We present the historical role of our A-AVA dataset in Figure 3.5. As the



Training Set:	47 videos (part of AVA dataset)
Testing Set:	30 videos (part of AVA dataset)
Actions Category:	80 actions (the same as AVA dataset)
Annotation Frequency:	Every 25 frames (the same as AVA dataset)

Figure 3.3: We create a new ASAD dataset based on existing AVA dataset [4], by assigning the unique actor identity to each actor.

first dataset that is specifically designed for the ASAD study, the A-AVA dataset covers a rich diversity of video scenes, as indoor and outdoor, different times of the day, various actor scales, and more. Those properties are not available in the previous dataset (*i.e.*, Okutama dataset). A-AVA dataset has bridged the gap between the SAD dataset and actor identification dataset.

3.2. Evaluation Metrics for ASAD

When considering the multi-label action, ASAD evaluation could be a complicated task. Unlike single-label SAD evaluation [4], it is challenging to simultaneously evaluate multi-label action classification and actor identification with spatial detection. To address this issue, we suggest evaluating ASAD from three aspects and then consider their overall performance. The three aspects include Spatial Detection Evaluation, Actor Identification Evaluation, and Multi-label Action Classification Evaluation (Figure 3.6).



Figure 3.4: Illustration of our Actor-identified AVA dataset.

3.3. Spatial Detection Evaluation

We take the object detection metrics [98, 99, 100] to evaluate the spatial detection performance. First, we calculate Intersection over Union (IoU), which is defined by

$$IoU = \frac{bbox^{pred} \cap bbox^{true}}{bbox^{pred} \cup bbox^{true}} \quad (3.1)$$

where $bbox^{pred}$ and $bbox^{true}$ represent the predicted bounding box and the ground-truth box, respectively.

Second, based on the IoU value, True Positive (TP), False Positive (FP), and False Negative (FN) are defined by

- True Positive (TP): A correct detection with an IoU greater the threshold.
- False Positive (FP): A wrong detection with an IoU smaller than the threshold.
- False Negative (FN): A ground truth not detected.

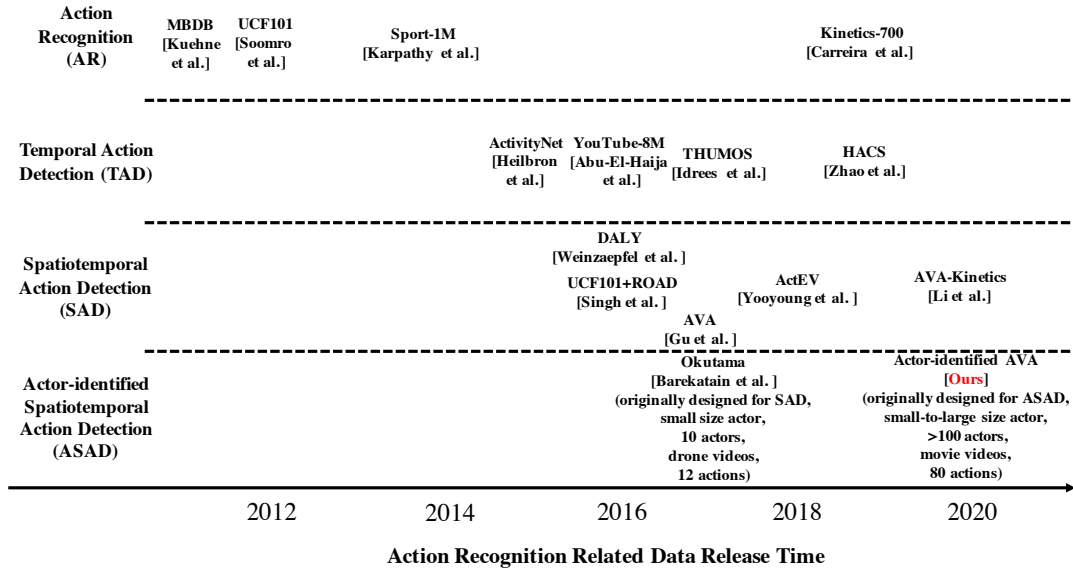


Figure 3.5: A historical timeline overview of datasets intended for video action recognition studies.

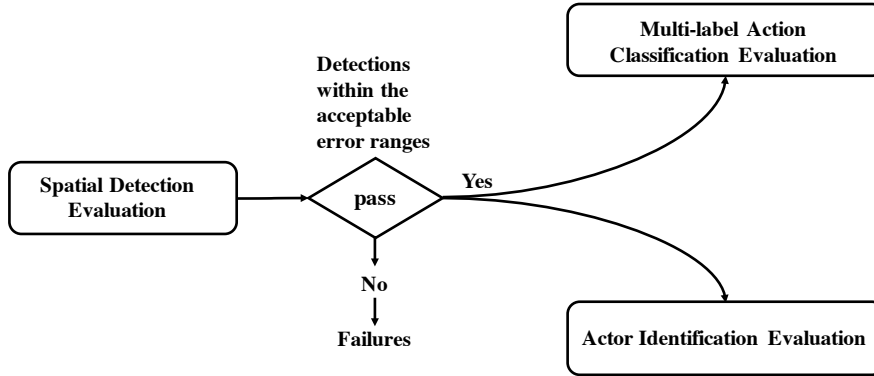


Figure 3.6: Overview of our ASAD metrics, which evaluate the performance of spatial detection, action classification, and actor identification.

and the corresponding Precision and Recall are

$$\begin{aligned}
 Precision &= \frac{TP}{TP + FP} \\
 Recall &= \frac{TP}{TP + FN}
 \end{aligned}
 \tag{3.2}$$

By traversing through all thresholds for detection confidence, different pairs

of precision and recall can generate the precision-recall curve, which indicates the association between precision and recall. To reduce the affect of the wiggles in the curve, the precision-recall curve is interpolated as p_{interp} . The p_{interp} at recall score r is assigned with the highest precision for $r > r'$:

$$p_{interp}(r) = \max_{r > r'} p(r') \quad (3.3)$$

Since we only treat human as actor, there is only one class for the detection, and therefore we utilize Average Precision (AP), other than Mean Average Precision (mAP), for the spatial detection evaluation. AP is the area under the interpolated precision-recall curve, which can be calculated using the following formula:

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{interp}(r_{i+1}) \quad (3.4)$$

In this thesis, we assume any spatial detection with an IoU value larger than 0.5 is True Positive, and the corresponding metrics are represented as AP@0.5.

3.4. Actor Identification Evaluation

While actor classification has the pre-defined actor identities, actor identification assigns each actor a unique identity and the number of actor identities is non-parametric. Therefore, we utilized part of Multiple Object Tracking (MOT) evaluation metrics for actor identification evaluation, as IDF1 (ratio of correctly identified detections), MT (mostly tracked targets), ML (mostly lost targets), and ID Switches. Those identification metrics were introduced by [10, 51] and have been popularly utilized for a while. More specifically, the IDF1, MT, and ML are respectively defined by

$$IDF1 = \frac{2IDTP}{2IDTP + IDFP + IDFN} \quad (3.5)$$

where IDTP, IDFP, IDFN respectively represent the True Positive ID, the False Positive ID, the False Negative ID.

$$\begin{aligned}
MT &= \sum_{i \in N_{T_{true}}} \mathbb{1}\left\{\frac{\text{len}(T_i^{pred})}{\text{len}(T_i^{true})} \geq 0.8\right\} \\
ML &= \sum_{i \in N_{T_{true}}} \mathbb{1}\left\{\frac{\text{len}(T_i^{pred})}{\text{len}(T_i^{true})} \leq 0.2\right\}
\end{aligned} \tag{3.6}$$

where T_i^{pred} and T_i^{true} respectively denote the predicted and the ground-truth Tracklet i , the number of T_i^{true} is $N_{T_{true}}$. If the prediction matches for the ground truth more than 80% of its life span, it is regraded as mostly tacked (MT). If the prediction only matches for the ground truth less than 20% of total length, it is regraded as mostly lost (ML).

3.5. Multi-label Action Classification Evaluation

It is intuitive to consider that each actor could take several actions simultaneously, which are corresponding to multi-label actions. For instance, an actor could be making a phone call and walking at the same time. Due to the lack of evaluation metrics, conventional Action Recognition studies have been evaluated with only the single-label action for a while [6, 4]. Therefore, we provide metrics for multi-label ASAD evaluation, which considers the evaluations of multi-label multi-class action classification and actor identification.

The evaluation metrics for actor detection and multi-label classification have been well-studied separately [98, 101], but the problem remains on how to associate them together for multi-label ASAD evaluation.

A simple approach could be evaluating the ‘‘actor’’ actor detection performance for all detected samples and then evaluating the multi-label action recognition performance for positively detected samples. For instance, assuming that a predicted sample is positive when $\text{IoU} \geq 0.5$ for the predicted and ground-truth bounding boxes, we can apply $HL@0.5$, which corresponds to Hamming Loss associated with $\text{IoU} \geq 0.5$, to measure its multi-label classification performance.

Note that, due to the object occlusions, the IoU value between multiple actors could be larger than 0.5. To remove such ambiguity, we apply the Hungarian Algorithm for bipartite matching between the predicted bounding boxes and the

ground-truth bounding boxes before comparing their classification results. Meanwhile, a pair that has IoU < 0.5 will be excluded before calculating their Hamming Loss. We illustrate these cases in Figure 3.7.

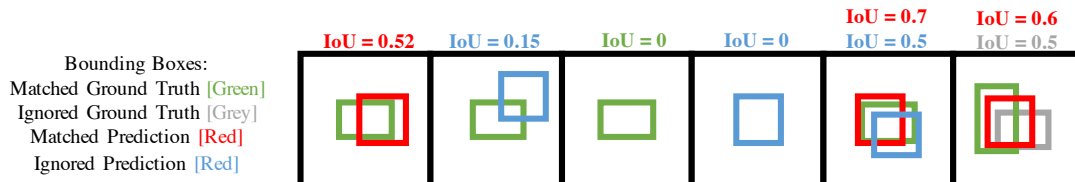


Figure 3.7: Illustration of matching pair between the ground-truth and the predicted samples.

In detail, we utilize matrix $\mathcal{D}_{i,j}$ to represent the matching distance between each ground-truth bounding box (denoted by i) and predicted bounding box (denoted by j), and we obtain $\mathcal{D}_{i,j}$ by

$$\mathcal{D}_{i,j} = \begin{cases} 1, & \text{if } IoU_{i,j} < 0.5; \\ 1 - IoU_{i,j}, & \text{otherwise.} \end{cases} \quad (3.7)$$

Next, we employ linear assignment [102] to obtain the optimal assignment \mathcal{M}^* with

$$\mathcal{M}^* = \arg \min_{\mathcal{M}} \sum_i \sum_j \mathcal{D}_{i,j} \mathcal{M}_{i,j}, \quad (3.8)$$

where \mathcal{M} is a Boolean matrix. When row i (*i.e.*, ground truth box i) is assigned to column j (*i.e.*, predicted box j), we have $\mathcal{M}_{i,j} = 1$. Each row can be assigned to at most one column and each column to at most one row.

Since matching pairs that have IoU value less than 0.5, we further process \mathcal{M}^* by

$$\mathcal{M}_{i,j}^* = \begin{cases} 0, & \text{if } \mathcal{D}_{i,j} = 1; \\ \mathcal{M}_{i,j}^*, & \text{otherwise.} \end{cases} \quad (3.9)$$

Referring to \mathcal{M}^* , we select matched pairs to evaluate the corresponding action labels with Hamming Loss. The number of matching pairs are represented by $N_{actors@0.5}$ (*i.e.*, $\mathcal{M}^* = 1$). Below, we show how the $HL@0.5$ is extended from the

original Hamming Loss.

$$HL@0.5 = \frac{1}{N_{actors@0.5}} \frac{1}{N_{labels}} \sum_{i=1}^{N_{actor@0.5}} \sum_{l=1}^{N_{labels}} Y_{true}^{i,l} \mathbf{XOR} Y_{pred}^{i,l}, \quad (3.10)$$

where **XOR** is an exclusive-or operation and N_{labels} stands for the number of action categories. Y_{true} and Y_{pred} are boolean arrays that denote the ground truth and predicted labels, respectively. To help understand the above metrics, we illustrate how $HL@0.5$ is calculated by a toy example in Figure 3.8.

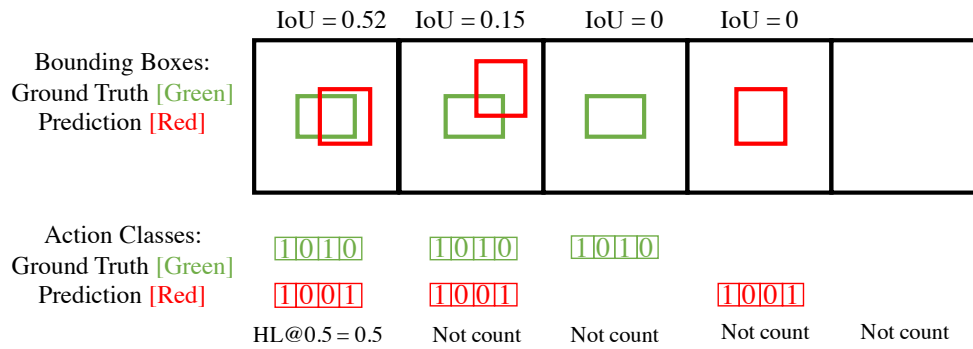


Figure 3.8: **An example of calculating HL@0.5.** Only the first case with IoU=0.52 is considered as a positively detected sample, and therefore the overall HL@0.5=0.5.

Chapter 4

Addressing the Bottleneck of ASAD by Improving Multiple Object Tracking

To approach ASAD, a simple pipeline could be performing Multiple Object Tracking (MOT) to obtain the Actor IDs and corresponding spatiotemporal boundaries and then applying Action Classification (AC) to generate actions within those spatiotemporal boundaries. In a simple pipeline, the MOT and the AC can work independently, but MOT might be the bottleneck to improve the overall performance in ASAD.

Multiple Object Tracking (MOT) is the key element to acquire actor IDs and the corresponding spatiotemporal boundaries in videos. In this Chapter, we focus on systematically exploring the data association strategies in MOT, aiming to boost the MOT performance on various MOT datasets. In the next chapter (*i.e.*, Chapter 5), we will discuss how to apply our MOT method in the ASAD framework and have experiments on our A-AVA dataset.

4.1. Overview

How would it be for an autonomous driving system to obtain trajectories of surrounding objects and make a safe path planning? Or for a survey system to track and count a herd of moving animals in the wild? Or for a virtual

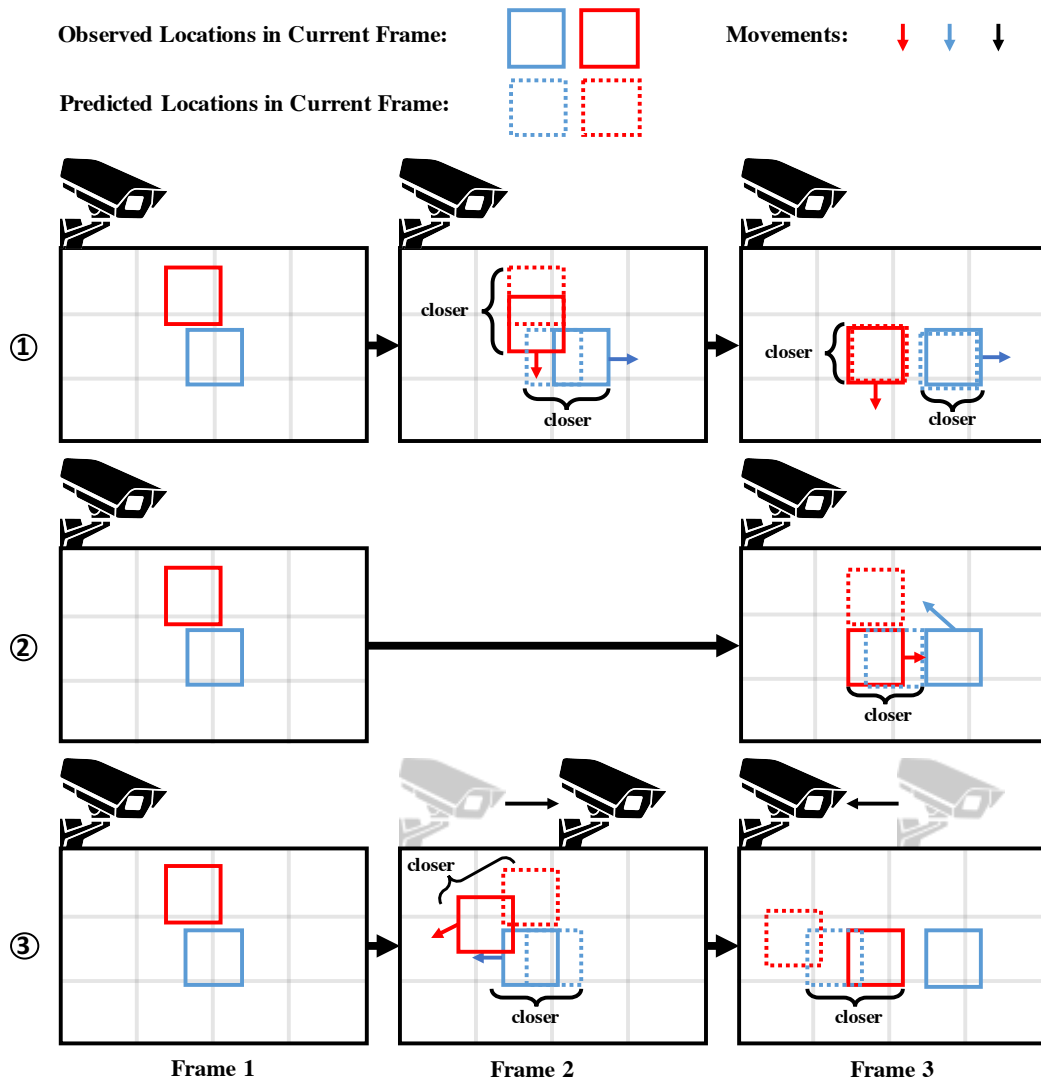


Figure 4.1: A demonstration that objects could be tracked fully by the appearance feature but partially by the motion feature. ① The ideal scenarios that both red and blue observations can be correctly associated by using either the motion feature or the appearance feature. ② Due to the faster motion, red and blue observations obtain incorrect motion initialization by only using the motion feature. ③ Due to the unpredictable movement of the camera, blue and red observations obtain incorrect motion initialization and then lose tracking by only using the motion feature.

game system to follow multiple players' commands? Multiple Object Tracking (MOT) is a core element to make these applications available and has drawn a lot of attention from autonomous driving, animal survey, and human-computer interaction.

Under the popular tracking-by-detection paradigm, MOT algorithms can be roughly divided into two phases: object detection and data association. Given a video data, object detection phase generates observations on each frame, with the format of bounding boxes (*e.g.*, [50]), instance masks (*e.g.*, [52]), or object key points (*e.g.*, [103]). In the data association phase, observations that correspond to the identical object are associated with a consistent set of trajectories, which are also referred to as tracklets. In this thesis, **we define our data association phase also includes feature extraction and matching graph construction processes**, so that we do not introduce extra phases in this work.

To accomplish data association in MOT, the motion feature and the appearance feature are generally used to distinguish a target object from others. The motion feature represents the tendency of object movement, which could be the estimated object position and scale in the future frame. In general assumption, the motion feature of the identical object should be consistent within adjacent frames, and thereby the affinity of consecutive-frame motion feature can be used for determining the connection between observations. Among existing works, the motion feature could be obtained from the recurrent neural network (RNN) [104], optical flow [105], and Kalman or Particle filters [106]. They take the advantage of being robust to the presence of occlusion or extremely similar object appearance, but fails when objects move unpredictably, due to **the camera movements, detection errors, or the deceptive movement in sports** (Figure 4.1). **Not that, we assume that the motion feature is initialized after associating at least two-frame observations, and therefore the estimated location for the second frame is the same as the first frame's.**

Since different objects may have distinct visual appearances, the appearance feature extracted from the object image is used as another important cue for data association. The appearance feature is extremely useful to handle irregular movement and long-term data association, where solely using the motion feature is insufficient. The recent successes of utilizing deep-learning generated appear-

ance feature in data association [61, 62, 56, 68, 73, 78, 71, 75, 76, 77] raise the question to what degree does appearance feature contribute to data association. Is the appearance feature alone sufficient for data association, and, in what kind of scenarios it may fail? Since the appearance feature learning is generally formulated as a Re-identification (ReID) problem [107, 108, 86, 109, 110, 111], we propose a ReID-dominated data association with a focus on answering the above questions.

We acknowledge that there are considerable discussions about applying ReID in MOT tasks among existing works, but what is the proper way to integrate ReID in MOT tasks is still an open problem. As a new practice, our ReID-dominated data association covers both online and offline data association. In particular, the online data association employs the motion feature to generate adaptive temporal-spatial constraints, and only observations fitting constraints are considered for consecutive-frame bipartite matching with the appearance feature. Note that, although DeepSORT [56] and JED [73] took advantage of a similar mechanism, our method greatly alleviates their defects in tracking initialization and forming motion-based constraints. On the other hand, online data association is often incapable of handling detection failures and occlusions by only utilizing the local spatio-temporal information. Thus, offline data association is commonly applied to leverage global spatio-temporal information. Our offline data association takes the Hierarchical Clustering (HC) [112] to associate tracklets generated by our online data association. Compared with methods that use Random Field [83] or Min-cost Flow [68], **HC allows us to add association constraints flexibly and obtain the corresponding hyper-parameter from the statistical information automatically.**

To our knowledge, however, the potentiality of HC was underrated in MOT tasks for a while. Because the clustering decision is made by a sensitive distance cutting threshold, on which heuristically selecting may lead to clustering errors in MOT tasks [59, 82]. We revive the renaissance of HC in offline data association, by proposing a novel method that automatically selects the cutting threshold referring to the statistical information of tracklets.

Our contributions are two-fold:

- We analyzed some ignored defects in existing data association approaches,

including motion initialization failures and losing tracking when objects move unpredictably. Since those defects could be caused by the over-reliance on the motion feature, we proposed a ReID-dominated data association to alleviate them (Section 4.2).

- From various perspectives, our experimentally demonstrated that our ReID-dominated data association can achieve better robustness on multiple visual MOT datasets (Section 4.3). The related mechanism also demonstrated the effectiveness by winning two championships in recent MOT challenges: BDD100K MOT of CVPR’20 WAD Workshop¹ and Track 1 of CVPR’20 MOTS Workshop².

4.2. Methodology

4.2.1 The Appearance Encoder

In our ReID-dominated data association method, only the appearance encoder is a trainable neural network. Our appearance encoder is partially inherited from a ReID work [86] and with necessary modifications, which is shown in Figure 4.2.

Specifically, a ResNet-50 [113] is used as the backbone, and its global-average-pooling output, which is a 2048-dimension vector, is employed as the appearance feature after an unbiased batch-normalization [114]. Refereeing to track IDs, triplets are sampled and used in contrastive learning [115] (Eq. 4.1). The appearance features are mapped to corresponding one-hot identify label by a single fully connected (FC) layer, in where the dimension of the output is equal to the number of identities. And the cross-entropy loss is applied for the identity classification (Eq. 4.2). Note that, when multi-class objects are included, the entire track IDs is the union of track IDs in each category. The identity predictions are further mapped to object categories by another fully connected layer, in where the dimension of the output is equal to the number of object categories. We also apply the cross-entropy loss for the category classification (Eq. 4.3). The

¹<https://bdd-data.berkeley.edu/wad-2020.html>

²<https://motchallenge.net/workshops/bmtt2020/tracking.html>

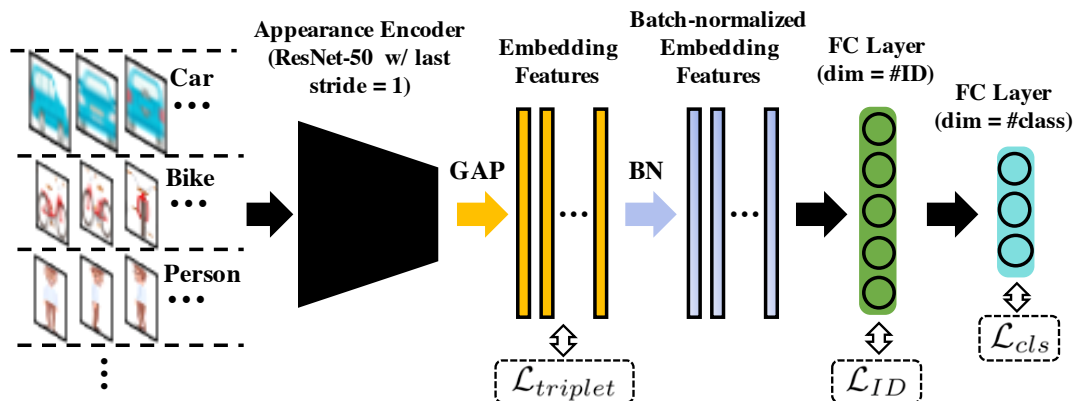


Figure 4.2: Appearance Encoder to learn appearance features with identities and classes. GAP and BN respectively represent Gap Average Pooling and Batch Normalization. Batched-normalized Embedding Features are selected as the appearance features for data association.

corresponding equations are organized as follows.

$$\mathcal{L}_{triplet} = \max \left[\|f_a - f_p\|_2^2 - \|f_a - f_n\|_2^2 + \alpha, 0 \right], \quad (4.1)$$

where f_a , f_p , and f_n denote the embedding of the anchor, positive, and negative samples, respectively; E is the appearance encoder; α indicates the margin between the positive and negative pairs and we empirically set $\alpha = 0.3$ as [116] suggested.

$$\mathcal{L}_{ID} = - \sum_{i=1}^{N_{id}} k_i \log(\hat{k}_i), \quad (4.2)$$

where k_i and \hat{k}_i respectively represent the ground-truth identity label and estimated identity label; N_{id} denotes the total number of unique object identities.

$$\mathcal{L}_{cls} = - \sum_{i=1}^{N_{cls}} c_i \log(\hat{c}_i), \quad (4.3)$$

where c_i and \hat{c}_i respectively represent the ground-truth category label and estimated category label; N_{cls} denotes the number of object categories.

For the labeled data, the \mathcal{L}_{sup} is applied to train the appearance encoder, and,

the optimization goal is to minimize

$$\mathcal{L}_{sup} = \mathcal{L}_{triplet} + \mathcal{L}_{ID} + \mathcal{L}_{cls}. \quad (4.4)$$

4.2.2 The Online Data Association

In online data association, the relationship of consecutive-frame observations could be formulated as a bipartite graph. Specifically, the node of the bipartite graph represents the observation and the edge signifies the distance between two consecutive-frame observations. The goal is to find the optimal matching solution in the bipartite graph. **Such an approach is commonly applied in recent MOT works [56, 73, 78] due to its simplicity and efficiency, though it was proposed for MOT tasks around 40 years ago [117].** However, even applying the same bipartite graph matching, **how to form the bipartite graph edges is still an open problem.** And more importantly, **a tiny difference of bipartite graph edge setting may lead to totally disparate results.**

As we discussed in Chapter 2, there are different combinations of utilizing MOT features. Our approach brings the merit from both features — the appearance feature plays the dominant role to form graph edges while the motion feature is used as adaptive temporal-spatial constraints. The value of our bipartite graph edge is the pair-wise appearance distance between two consecutive-frame observations under spatio-temporal constraints (Figure 4.3). Letting a matrix \mathcal{D} represent bipartite graph edges and $\mathcal{D}_{i,j}^{t,t+1}$ denote the association distance between observation i (at frame t) and j (at frame $t + 1$), we define

$$\mathcal{D}_{i,j}^{t,t+1} = \begin{cases} inf, & \text{if } c_i \neq c_j; \\ inf, & \text{if } IoU(\pi_i^{t+1}, b_j^{t+1}) = 0; \\ 1 - \frac{f_i^t f_j^{t+1}}{\|f_i^t\| \|f_j^{t+1}\|}, & \text{otherwise,} \end{cases} \quad (4.5)$$

where f_i^t and f_j^{t+1} are respectively the appearance feature of observation i and j ; c_i and c_j are the object class; π_i^{t+1} is the adjustable matching window of observation i and b_j^{t+1} is the bounding box of observation j , when their IoU value equals 0 (*i.e.*, no overlap), we suppose the likelihood for their matching is low and thus set the distance value as infinite.

For the motion initialization, to be robust to the fast movement, we define the transformation from $b_i^t = \{x_b^t, y_b^t, w_b^t, h_b^t\}$ to $\pi_i^{t+1} = \{x_\pi^{t+1}, y_\pi^{t+1}, w_\pi^{t+1}, h_\pi^{t+1}\}$ as

$$\begin{aligned} x_\pi^{t+1} &= \max(0, x_b^t - w_b^t(r_w - 1)/2), \\ y_\pi^{t+1} &= \max(0, y_b^t - h_b^t(r_h - 1)/2), \\ w_\pi^{t+1} &= \min(r_w w_b^t, W_{img} - x_\pi^{t+1}), \\ h_\pi^{t+1} &= \min(r_h h_b^t, H_{img} - y_\pi^{t+1}), \end{aligned} \tag{4.6}$$

where r_w and r_h respectively are the expanding scales for w and h , and their default values are from the largest shifting scales in labeled data; W_{img} and H_{img} are the width and the height of a video frame respectively.

After the motion initialization, we can estimate the location $\hat{b}^{t+1} = \{\hat{x}_b^{t+1}, \hat{y}_b^{t+1}, \hat{w}_b^{t+1}, \hat{h}_b^{t+1}\}$ in future frame $t + 1$ by leveraging Kalman Filter or other motion models. We conjecture that \hat{b}^{t+1} may not be robust to unpredictable motions and instead we apply π_i^{t+1} , with a new transformation from \hat{b}^{t+1} to π_i^{t+1} as

$$\begin{aligned} x_\pi^{t+1} &= \max\left(0, \min\left(x_b^t - w_b^t(r_w - 1)/2, \hat{x}_b^{t+1}\right)\right), \\ y_\pi^{t+1} &= \max\left(0, \min\left(y_b^t - h_b^t(r_h - 1)/2, \hat{y}_b^{t+1}\right)\right), \\ w_\pi^{t+1} &= \min\left(W_{img}, \max\left(x_b^{t+1} + r_w w_b^t, \hat{x}_b^t + \hat{w}_b^t\right)\right) - x_\pi^t, \\ h_\pi^{t+1} &= \min\left(H_{img}, \max\left(y_b^{t+1} + r_h h_b^t, \hat{y}_b^t + \hat{h}_b^t\right)\right) - y_\pi^{t+1}. \end{aligned} \tag{4.7}$$

Introducing the adjustable matching window is our main contribution, as distinct from previous methods that have a similar approach (*e.g.*, DeepSORT [56] and JDE [73]). DeepSORT [56] and JDE [73] take 9 default Mahalanobis gating thresholds of Kalman Filter for spatio-temporal constraints, but the unpredictable object shifting (*e.g.*, camera movement) has not been considered. **Our adjustable matching window is designed for compensating unpredictable object shifting.** As shown in Figure 4.3, identical observations could be excluded when the estimated motion acting as an incorrect spatio-temporal constraint. Nonetheless, removing the spatio-temporal constraint may increase the likelihood of including more objects that have a similar appearance, which may lead to association failures. Our proposed adjustable matching window attempts

to make a trade-off between them. Supposing labeled videos share homogeneous properties with the target unlabeled videos. When observed maximum shifting is small in labeled videos, the value of r_h and r_w are closed to 1 and thus π_i^{t+1} has a similar value as b_i^t in the motion initialization. After the motion initialization, π_i^{t+1} will be flexibly adjusted by referring to the online motion updating. We will further demonstrate the effectiveness of using our adjustable matching window in ablation studies.

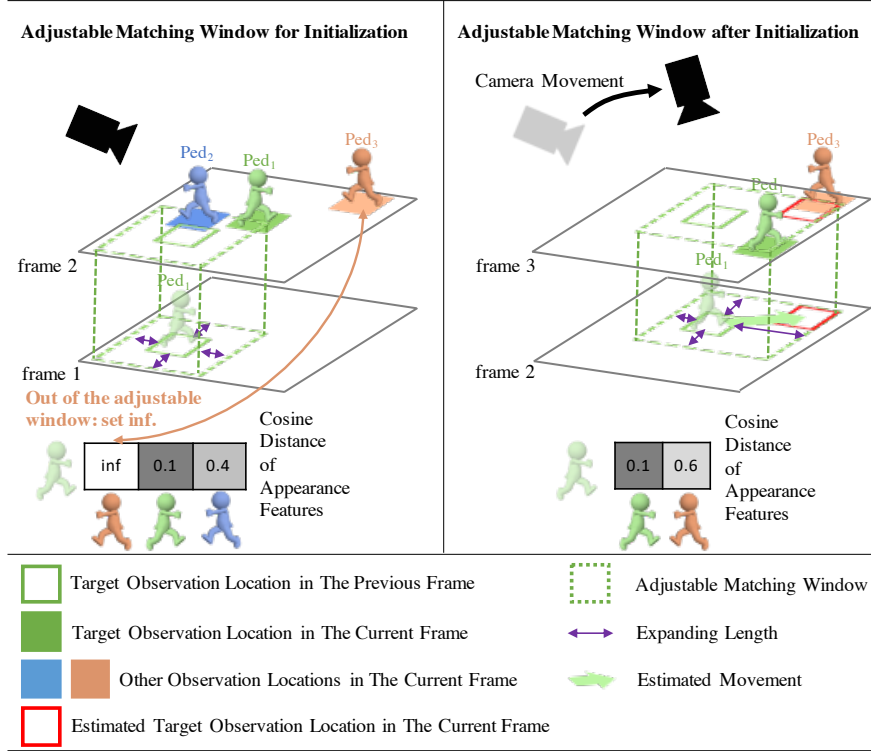


Figure 4.3: Illustration of our online data association. For the initialization, the previous-frame location is extended to the adjustable matching window by adding the maximum shifting distance obtained from the statistics of labeled data (Eq. 4.6). After the initialization, a rectangle, that bounds the expanded previous-frame box and the estimated current-frame box, is used as the adjustable matching window (Eq. 4.7). Only observations, covered by the adjustable matching window, are considered for data association by using appearance features, which makes a trade-off between excluding impossible matching candidates and including potential ones.

Only applying spatio-temporal constraints may not filter all unwanted observations out in the data association. For instance, a new object, that just enters the video scene, may have large appearance distances to tracked observations, though, some times it may not be filtered out by spatio-temporal constraints. Generally, we may be able to exclude such a new object if it has a large appearance distance to existing tracklets. However, it is challenging to heuristically determine how large the appearance distance should be to exclude a new object. We tackle this issue by proposing an adaptive appearance threshold, which is denoted as $\theta_{online}^{app}[t+1]$ when associating observations between frame t and frame $t+1$. Without introducing heuristic parameters, $\theta_{online}^{app}[t+1]$ can be inferred from the obtained tracklets, the corresponding formula is

$$\theta_{online}^{app}[t+1] = \frac{1}{N^{t+1}(N^{t+1}-1)} \sum_{i \neq j}^{N^{t+1}N^{t+1}} \left(1 - \frac{f_i^{t+1} f_j^{t+1}}{\|f_i^{t+1}\| \|f_j^{t+1}\|} \right), \quad (4.8)$$

where N^{t+1} represents the number of observations at frame $t+1$.

In Eq. 4.8, we assume that intra-frame observations at frame $t+1$ belong to different objects, and therefore their appearance distances should be larger than identical observations'. In fact, $\theta_{online}^{app}[t+1]$ is the mean of appearance distances for intra-frame observations at frame $t+1$. Whenever the appearance distance between a pair of cross-frame observations is lower than $\theta_{online}^{app}[t+1]$, we suppose they are different objects. Consequently, after applying Eq. 4.5, we further process $\mathcal{D}_{i,j}^{t,t+1}$ by letting

$$\mathcal{D}_{i,j}^{t,t+1} = \begin{cases} inf, & \text{if } \mathcal{D}_{i,j}^{t,t+1} < \theta_{online}^{app}[t+1]; \\ \mathcal{D}_{i,j}^{t,t+1}, & \text{otherwise.} \end{cases} \quad (4.9)$$

Unlike many MOT studies that focus on proposing new graph optimization solutions, **we concentrate on improving graph edges in our ReID-dominated data association**. Using the carefully designed association distance matrix \mathcal{D} (*i.e.*, graph edges), without bells and whistles, we apply the linear assignment [102] to obtain the optimal assignment \mathcal{M}^* with

$$\mathcal{M}^* = \arg \min_{\mathcal{M}} \sum_i \sum_j \mathcal{D}_{i,j} \mathcal{M}_{i,j}, \quad (4.10)$$

where \mathcal{M} is a Boolean matrix. When row i (*i.e.*, observation i) is assigned to column j (*i.e.*, observation j), we have $\mathcal{M}_{i,j} = 1$. Each row can be assigned to at most one column and each column to at most one row.

Since objects may dynamically move outside, or, into the video, we have to aware that the bijective matching only happens when one-to-one correspondence existing for cross-frame observations. While in other cases, we need to exclude matching with small likelihood (*i.e.*, with value infinite). Accordingly, we further process the optimal assignment \mathcal{M}^* with

$$\mathcal{M}_{i,j}^* = \begin{cases} 0, & \mathcal{D}_{i,j} = inf; \\ \mathcal{M}_{i,j}^*, & \text{otherwise.} \end{cases} \quad (4.11)$$

To this end, based on \mathcal{M}^* , we keep updating our tracklet records and move forward to the next frame. To avoid being misled by unpredictable motion noise, some related works [118, 105] utilized optical flow to guide their data association. To some extent, the optical flow conducts cross-frame point-to-point matching, it is essentially similar to cross-frame observation-to-observation matching in our ReID-dominated approach. However, since the tracking target could be non-rigid objects (*e.g.*, pedestrians), it might be challenging to make a satisfactory optical flow estimation in unseen videos. While performing observation-to-observation matching could partially alleviate such a problem.

4.2.3 Offline Data Association

Offline data association can be used in MOT processing when latency is allowed. Compared with the online data association, the offline data association not only can access the global information of observations, but also fine-tune the appearance encoder on target videos with pseudo tracklet labels.

We utilize the unlabeled target videos for self-supervised learning in our offline data association. Our online data association results can be employed as pseudo labels to refine the appearance encoder on unlabeled target videos. Such a practice may not be applicable for real-time applications but **be ideal to be utilized in assisting MOT annotation works**. Due to the estimation errors in our online data association, different pseudo tracklets may have the same identity, especially when they are separated in the temporal domain. Unlike the previous work [63] ignores this issue when sampling pseudo labels for ReID training, we attempt to alleviate it by porpoising a new sampling strategy: within the same video, we only gather triplets from temporally overlapped tracklets, witch

means $\Pi_p \cap \Pi_q \neq \emptyset$ for pseudo tracklet T_p and T_q . By doing this, we can reduce the possibility of treating identical samples as negative pairs. The procedure of constructing inputs is illustrated in Figure 4.4. During the training process, only Eq. 4.1 is applied for pseudo-label samples, while Eq. 4.4 is applied for the labeled samples.

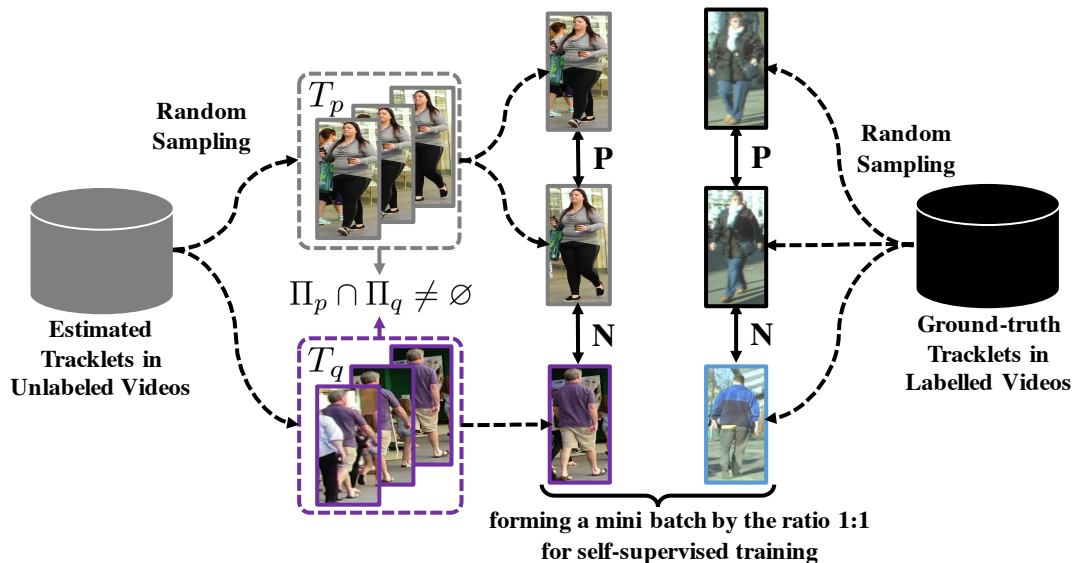


Figure 4.4: The illustration of constructing inputs for offline self-supervised learning. Within the same video, we only gather triplets from temporally overlapped tracklets, as $\Pi_p \cap \Pi_q \neq \emptyset$ for pseudo tracklet T_p and T_q . In a mini-batch of input, the ratio between samples of labeled videos and unlabelled target videos is 1 : 1.

Previous offline MOT works [82, 83, 68] created short-term tracklets first and then associated them to long-term tracklets with the global information. Given that we now had the short-term tracklets obtained from our online data association, our offline data association can directly work on them. With improved appearance features, associating online-obtained tracklets is cast as a Hierarchical Clustering (HC) [112] problem by optimizing an undirected graph \mathcal{W} . Each

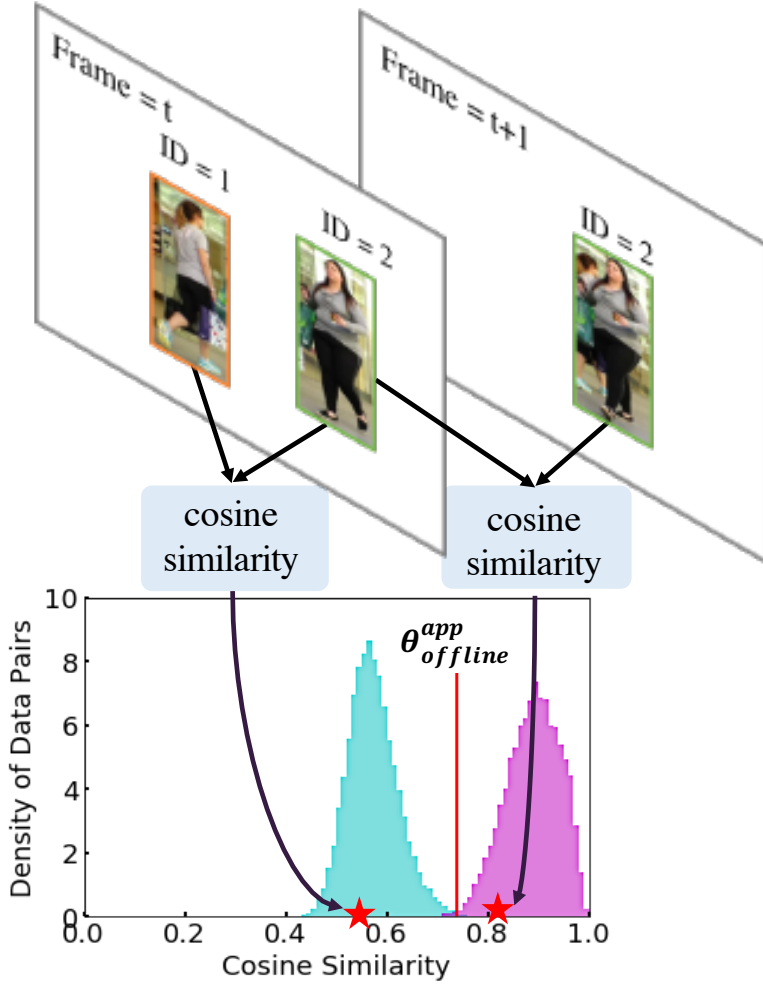


Figure 4.5: The histograms of cosine similarity for intra-frame and intra-tracklet observations. By approximating the histogram as a normal distribution, the boundary that maximally separates two histograms is selected as the offline appearance threshold $\theta_{offline}^{app}$, which minimizes the sum of the false positives and negatives of cosine similarity distributions..

node of this undirected graph represents a tracklet and an edge is defined as

$$\mathcal{W}_{p,q} = \begin{cases} inf, & \text{if } c_i \neq c_j; \\ inf, & \text{if } p = q, \\ inf, & \text{if } \Pi_p \cap \Pi_q \neq \emptyset, \\ \frac{1}{N_p N_q} \sum_{i \in \Pi_p} \sum_{j \in \Pi_q} \left(1 - \frac{f_i^p f_j^q}{\|f_i^p\| \|f_j^q\|} \right), & \text{otherwise,} \end{cases} \quad (4.12)$$

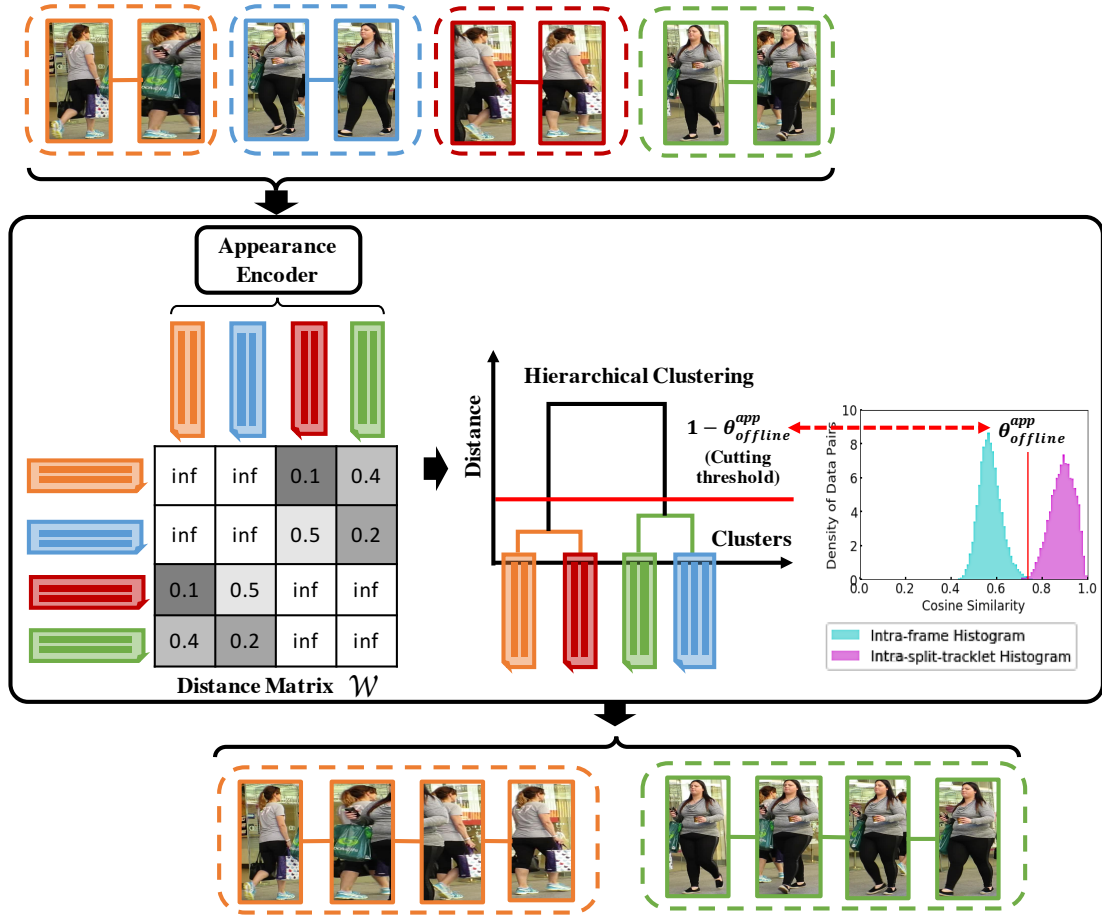


Figure 4.6: The Hierarchical Clustering is accommodated to associate short-term tracklets into long-term tracklets. The cutting threshold is obtained from the statistical information of splitting tracklets.

where respectively to tracklets T_p and T_q , $\mathcal{W}_{p,q}$ is the distance between them; c_p and c_q are their corresponding object classes; Π_p and Π_q are their corresponding frame sets, f_i^p and f_j^q are their appearance features at frame i and j ; N_p and N_q are the number of observations within a tracklets.

We impose spatio-temporal constraints and appearance constraints to construct undirected graph \mathcal{W} . In Eq. 4.12, whenever the matching condition violates given constraints, we set their distance value to be infinite. Based on the distance matrix, we accommodate HC to merge the split tracklets. Nonetheless, the main challenge of applying HC in the MOT task is how to set the proper

cutting threshold. We do not give a heuristic value but let the data speak for itself. After pseudo-tracklet self-supervised learning, we generate cosine similarity histograms for intra-frame and intra-split-tracklet, respectively. In the ideal case, if the cosine similarity between two observations falls into the intra-frame histogram, it represents these two observations have different identities, and vice versa when it falls into the inter-frame histogram. We suppose that intra-frame and intra-split-tracklet cosine similarity histograms can be maximally separated at $\theta_{offline}^{app}$ (Figure 4.5), which minimizes the sum of the false positives and negatives of cosine similarity distributions. Without access to the ground-truth, such a threshold may distinguish objects adaptively based on appearance features. Therefore, we set $1 - \theta_{offline}^{app}$ as the cutting threshold in HC (Figure 4.6). An advantage of our HC approach over [59, 82, 63] is its simplicity and the fact that we can automatically generate an adaptive cutting threshold, without heuristically setting a static one. Note that, while we can directly obtain the threshold from the statistical information in HC, it is challenging to apply such strategy in multi-cut approaches [61], since its clustering is performed in the Laplacian space. This could be another advantage of our method.

4.3. Experiments on MOT Datasets

We experimented on multiple MOT datasets from diverse perspectives. First, we performed online data association evaluation on BDD100K MOT dataset [49]. Next, we conducted offline data association on MOTS20 dataset [52]. Finally, we trained our appearance encoder on Market-1501 dataset [9] and explored both online and offline MOT on MOT15-17 [50, 51] with their oracle detection.

4.3.1 Implementation Details

We implemented our proposed method by Pytorch. Since **our contributions do no depend on the object detector**, we utilized the off-the-shelf object detector — Mask R-CNN X152 [119] of Detectron2 [120], to generate bounding boxes and masks. We ran Detectron2 with its default settings on 4 NVIDIA GTX1080Ti GPUs. Our appearance encoder was modified from [86], which can be trained and inferred on a single NVIDIA GTX1080Ti GPU. Referring to boxes/masks

generated by the Mask R-CNN X152 of Detectron2, the cropped image of each observation is resized to be 128×256 as the input of our appearance encoder. We applied data augmentation strategies that include Random horizontal flipping, random color jittering, and random affine transformation in the training. For the appearance encoder optimization, we choose the Adam optimizer [121] with a learning rate of 1×10^{-4} . During the merging process, we applied the centroid linkage criteria to determine the distance between newly merged tacklets in Hierarchical Clustering.

4.3.2 Experimental Datasets

BDD100K MOT dataset [49] is a large-scale MOT dataset, which covers 2,000 fully annotated 40-second sequences under different weather conditions, time of the day, and scene types. BDD100K MOT dataset contains 8 category of objects, which are divided into three super-categories: “person” (with classes “pedestrian” and “rider”), “vehicle” (“car”, “bus”, “truck”, and “train”), and “bike” (“motor-cycle” and “bicycle”). The evaluation requires correctly identify the instance ID and class ID simultaneously. The evaluation of the testing set is performed on the CodaLab website.

MOTS20 dataset [52] are Multi-Object Tracking and Segmentation (MOTS) datasets. The evaluation of the testing set is performed on the MOTChallenge website. As the name suggested, MOTS studies aim to track the object with the instance segmentation. Since different objects may stay in the same bounding box for MOT, MOTS utilizes masks to decrease the ambiguity for data association. Although previous works [68, 69] attempted to work on MOTS, their evaluations are based on MOT metrics. Specific MOTS evaluations metrics are proposed in [52].

KITTI-MOTS datasets [53, 52] contain 29 videos in the training set, in which instance masks are given. Compared with the MOTS20 dataset that only has pedestrian objects, the object category of the KITTI-MOTS dataset covers pedestrians and cars. The evaluation of the testing set is performed on the KITTI website.

MOT15-17 datasets [50, 51] are a series of MOT datasets released from 2015 to 2017, where the testing set evaluation is performed on the MOTChallenge

official website. The video length ranges from 3 seconds to 2 minutes, and each dataset contains around 10 videos for training and testing, respectively. Besides, video covers a rich diversity of scenes, as indoor and outdoor, different times of the day, different image resolutions, and more. Although multi-class annotations can be found in the training sets of MOTChallenge15-17 datasets, we generally treat them as single-class MOT tasks since only the “pedestrian” class is evaluated on the MOTChallenge website.

Challenge Rank	sMOTA(%) \uparrow
1 st place (Our Online Approach)	33.67
2 nd place	31.64
3 rd place	26.40

Table 4.1: The top-3 results for BDD100K MOT Challenge of CVPR 2020 Workshop on Autonomous Driving (June-13th-2020).

4.3.3 Evaluation Metrics

We evaluated the data association performance on box-based MOT tasks with such widely recognized CLEAR MOT metrics as sMOTA (box-based soft MOT accuracy), MOTA (MOT accuracy), IDF1 (ratio of correctly identified detections), MT (mostly tracked targets), ML (mostly lost targets), and ID Switches [50, 51]. For the mask-based MOT, we replaced sMOTA and MOTA with sMOTSA (mask-based soft MOT accuracy) and MOTSA (mask-based MOT accuracy) [52], respectively. In the BDD100K MOT dataset, there are 8-categories objects, and therefore mMOTA (Mean MOT accuracy), which is the average of MOTA values for the 8 categories, is employed as the dominant metric.

4.3.4 Online Data Association Evaluation

Table 4.1 reports the ranking of BDD100K MOT Challenge. By using our online approach, we obtained the 1st place with the sMOTA score as 33.67%, which outperformed the 2nd place by 2.03%. Our online approach was not proposed by

one shot, instead, we improved it step-by-step, which is illustrated in Table 4.2a. For all our experiments on the BDD100K MOT dataset, we trained the Mask R-CNN X152 of Detectron2 to estimate bounding boxes. And our appearance encoder, which is described in Section 4.2.1, was also trained to generate appearance features. That means, we applied identical features for all ablation settings, so that we can focus on how to construct bipartite graph edges.

We first tried SORT [66] (*i.e.*, i) in Table 4.2) to associate bounding boxes generated by Detectron2. Since BDD100K MOT dataset was captured by a vehicle-mounted camera, the object movement could be **much faster** than those of other MOT datasets. It brings the challenge of motion initialization when only the motion feature is considered, because identical cross-frame observations may have the Intersection over Union (IoU) values as 0. Besides, due to the occlusion between observations, only comparing the motion similarity may insufficient for correct data association. To leverage the appearance feature, we utilized DeepSORT [56] in our following practice. We equipped DeepSORT with our appearance encoder (*i.e.*, ii) in Table 4.2), which is stronger than the original one in DeepSORT. Compared with setting i), we reached an improvement in setting ii), from sMOTA value 19.50% to 22.42%. After investigating a vast of failure cases, we supposed the tracklet initialization failure could be the main reason that prevented us to obtain satisfactory results. It can be noticed that both i) and ii) of Table 4.2 share the identical tracklet initialization strategy, which might be unsuitable for fast-moving objects. To improve the tracklet initialization strategy, we attempted to utilize the method proposed by JDE [73] and FairMOT [78] (*i.e.*, iii) in Table 4.2). The data association of JDE and FairMOT, though, were inherited from DeepSORT, their tracklet initialization incorporated both motion features and appearance features. Such a tiny chance significantly improved the sMOTA value from 22.42% to 30.53%. When the object movement is complicated, we had verified the importance of utilizing appearance features in the motion initialization process.

By looking at the results of settings i), ii), and iii), we realized the appearance feature may play a key role to improve the MOT performance. To figure out to what degree does appearance feature solely contribute to data association, we designed setting iv), which only utilized the appearance feature for data as-

sociation. Surprisingly, the sMOTA score of iv) is even higher than iii)’s. This result could be attributed to that iv) avoided the side effect of inaccurate motion features. **Even object occlusion and various illumination existing, only applying ReID in data association seemed to be more powerful than we had expected.** Nonetheless, removing the spatio-temporal constraint may increase the likelihood of including more objects that have a similar appearance, which may also lead to association failures. Thus, we were motivated to propose an adjustable matching window (AMW) for utilizing motion features. Before the BDD100K MOT challenge deadline, our approach could be summarized as setting v). Referring to Eq. 4.6, the AMW was simply expanded from the observed box and the predicted box in the tracklet initialization process and after the tracklet initialization process, respectively. Compared with setting iii), setting v) essentially leverages a more flexible spatio-temporal constraint in terms of using motion features. Consequently, setting v) achieved an improved sMOTA value of 33.67%. We did not stop our exploration after the BDD100K MOT challenge deadline. Due to the motion estimation errors, we detected the desired observations could locate outside of its corresponding AMW after the tracklet initialization. To address this issue, we compensated the object movement in AMW after the tracklet initialization (*i.e.*, vi) of Table 4.2a), which was formulated in Eq. 4.7. Compared with our winner solution v), a superior result was achieved in setting vi), with the sMOTA value of 34.36%. In addition, We illustrate the qualitative results in Figure 4.7. Our method shows its effectiveness from diverse perspectives, including variant camera movements, illumination conditions, and object categories.

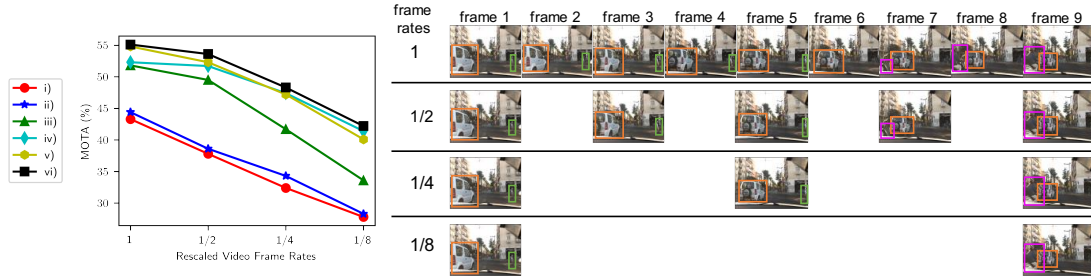
Through our investigation of the above results, we also found that an object with a slower speed could be relatively easier to be tracked, because both its location and appearance changes may be smaller at each frame. As it lacks a systematic study on how the object speed affects the MOT performance in existing works, we started to investigate it by conducting ablation studies in Table 4.2c. We simulated the speed change by modifying the Frame Rates in videos. The smaller Frame Rates leads to relatively faster object speed and vice versa. On the validation set of BDD100K MOT dataset, we sampled original videos and the ground-truth label with the Frame Rates of 1, 1/2, 1/4, 1/8. We mainly

Data Association Approach	Ablation Study Settings			
	Object Detection	Appearance Encoder	Association for Track Init.	Association after Track Init.
i): from SORT [66] (ICIP'16)	Detectron2 [120]	-	IoU Matching	IoU Matching for KF Predicted Locations
ii): from DeepSORT [56] (ICIP'17)	Detectron2 [120]	Section 4.2.1	IoU Matching	Step1: KF gating + App. Matching, Step2: IoU Matching for Unmatched
iii): from JDE [73] (ECCV'20)	Detectron2 [120]	Section 4.2.1	KF gating + App. Matching	Step1: KF gating + App. Matching, Step2: IoU Matching for Unmatched
iv)	Detectron2 [120]	Section 4.2.1	App. Matching	App. Matching
v): our winner solution for BDD100K MOT challenge)	Detectron2 [120]	Section 4.2.1	App. Matching + Adjustable Matching Window of Eq. 4.6	App. Matching + Adjustable Matching Window of Eq. 4.6
vi): our improved solution after the challenge deadline	Detectron2 [120]	Section 4.2.1	App. Matching + Adjustable Matching Window of Eq. 4.6	App. Matching + Adjustable Matching Window of Eq. 4.7

(a) Ablation study settings for our online approach. App. and KF represent the appearance feature and Kalman Filter, respectively.

Approach	sMOTA(%) \uparrow	MOTA(%) \uparrow				# MT \uparrow				# ML \downarrow				# ID Sw. \downarrow			
		all	Person	Vehicle	Bike	all	Person	Vehicle	Bike	all	Person	Vehicle	Bike	all	Person	Vehicle	Bike
i)	19.50	36.55	21.34	42.14	11.16	2509.00	170.00	2366.00	11.00	12394.00	7781.00	7408.00	582.00	44800.00	8869.00	49118.00	389.00
ii)	22.42	38.93	25.85	47.36	15.81	2877.00	421.00	3584.00	38.00	8948.00	5694.00	6032.00	517.00	43262.00	8792.00	48978.00	391.00
iii)	30.53	53.75	42.05	60.82	27.83	16097.00	1843.00	15545.00	120.00	5123.00	1347.00	2896.00	287.00	43049.00	8271.00	43723.00	323.00
iv)	31.83	55.57	42.60	62.83	28.12	16739.00	1917.00	15178.00	123.00	5045.00	1353.00	2837.00	285.00	43762.00	7943.00	41982.00	347.00
v)	33.67	59.76	44.59	67.18	29.77	16774.00	1922.00	15191.00	123.00	5004.00	1344.00	2785.00	278.00	42901.00	7912.00	38558.00	277.00
vi)	34.36	60.44	45.63	67.83	30.74	16769.00	1922.00	15184.00	123.00	5009.00	1346.00	2786.00	280.00	36841.00	6743.00	33557.00	162.00

(b) Online MOT ablation study results on BDD100K MOT **testing set**. The original video frame rates is utilized.



(c) Online MOT ablation study results on BDD100K MOT **validation set**. We adjust the video frame rates to verify the robustness of different online data association settings.

Table 4.2: Ablation studies for our online approach on BDD100K MOT Dataset. The identical detection and appearance encoder are utilized in each approach. In each column, Red and Blue represent the first and second results, respectively.

compared the MOTA values for each approach. The result is quite revealing in several ways. First, the performance of settings i), ii) and iii) considerably drop

when the Frame Rates is decreased (*i.e.*, the object speed is increased). Second, although the performance of settings iv), v), and vi) is also decreased, they hold a superiority compared with settings i), ii) and iii), which could be attributed to ReID-dominated data association are more robust to fast speed. The result is consistent with our assumption.

4.3.5 Offline Data Association Evaluation

Rank	Method	sMOTSA \uparrow	IDF1($\%$) \uparrow	MOTSA($\%$) \uparrow	MOTSP($\%$) \uparrow	MODSA($\%$) \uparrow	# MT \uparrow	# ML \downarrow	# ID Sw. \downarrow
1 st place	ReMOTS [122]	69.9	75.0	83.9	84.0	85.1	248	12	388
2 nd place	PTPM	68.8	68.5	82.6	84.1	83.7	244	19	368
3 rd place	PT	68.4	64.9	82.6	83.9	84.4	248	10	569

Table 4.3: CVPR 2020 MOTS Track 1 Challenge (May 30th, 2020).

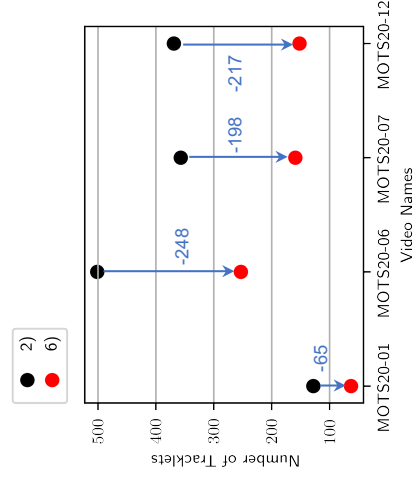
We applied our offline approach to win the 1st on CVPR 2020 MOTS Track 1 Challenge (Table 4.3). Our results outperformed the 2nd place result 1.1% and 6.5% for sMOTSA and IDF1, respectively. The notable margin of IDF1 indicates that our approach can correctly associate a larger ratio of obtained masks.

After the challenge deadline, MOTS20 evaluation has been re-opened and therefore we formed ablation studies on the testing set. Our ablation settings are described in Table 4.4. Through utilizing the global information, the results of our offline approaches are substantially better than our online ones, *i.e.*, 3) *v.s.* 1) and 4) *v.s.* 2). By respectively comparing settings 2), 5) and 6) to settings 1), 3) and 4), we verified that using Adjustable Matching Window (AMW) can achieve better results. This is consistent with what has been found in the ablation studies of the BDD100K MOT dataset. In some application scenarios, for instance, automatic MOT labeling, fine-tuning the appearance encoder on unlabeled videos is practicable. Our results demonstrated that the fine-tuning process could further improve the performance, as shown in 4) *v.s.* 3) and 6) *v.s.* 5).

The short-term tracklets, generated by our proposed online approach as introduced in Section 4.2.3, can be linked to long-term tracklets in our offline approach. In TABLE 4.4b, we explicitly illustrate the change of tracklet number after applying offline data association on the online-estimated results. It can

Approach	sMOTSA \uparrow	IDFI (%) \uparrow	MOTSA (%) \uparrow	# MT \uparrow	# ML \downarrow	# ID Sw. \downarrow
1) Online w/o AMW	65.0	69.8	77.4	214	32	470
2) Online w/ Adjustable Matching Window	66.4	70.4	80.0	231	28	468
3) Offline w/o Adjustable Matching Window	69.1	73.4	83.4	248	12	431
w/o Fine-tuning Appearance Encoder on the Testing Set						
4) Offline (<i>i.e.</i> , our winner solution ReMOTS [122]) w/o Adjustable Matching Window	69.9	75.0	83.9	248	12	388
w/ Fine-tuning Appearance Encoder on the Testing Set						
5) Offline w/ Adjustable Matching Window	70.0	73.6	84.0	248	12	309
w/o Fine-tuning Appearance Encoder on the Testing Set						
6) Offline w/ Adjustable Matching Window	70.4	75.0	84.0	248	12	231
w/ Fine-tuning Appearance Encoder on the Testing Set						

(a) The overall performance comparison for various settings.



(b) The number of tracklets from online to offline data association.

Table 4.4: The ablation studies on MOTS20 testing set.

be noticed that our offline data association significantly decreased the traklet number. Consequently, some fragment tracklets, which could be caused by false-negative detection, were successfully connected to complete ones. The qualitative result is shown in Fig. 4.7.

We also evaluated on the MOT15-17 testing set and compared with state-of-the-art methods to show the superiority of our ReID-dominated data association (Table 4.5). Note that, our offline approach may not generate state-of-the-art results for all cases. For instance, we experimented the KITTI-MOTS dataset with our offline approach, and the corresponding results are shown in Table 4.6. Since the quality of our estimated masks may not be compatible with other methods, we only obtained acceptable performance on the KITTI-MOTS dataset.

Testing Datasets	Approach	MOTA(%) \uparrow	IDF1(%) \uparrow	# MT \uparrow	# ML \downarrow	# ID Sw. \downarrow
MOT15 Test Set	Tube_TK [123] (CVPR 2020)	58.4	53.1	283	130	854
	FairMOT [78] (ArXiv 2020)	60.6	64.7	343	79	591
	Our Offline	63.6	67.0	382	96	445
MOT16 Test Set	Chained-Tracker [124] (ECCV 2020)	67.6	57.2	250	175	1897
	FairMOT [78] (ArXiv 2020)	69.3	72.3	306	127	815
	Our Offline	76.9	73.2	390	94	742
MOT17 Test Set	Chained-Tracker [124] (ECCV 2020)	66.6	57.4	759	570	5529
	FairMOT [78] (ArXiv 2020)	73.7	72.3	1017	408	3303
	Our Offline	77.0	72.0	1218	324	2853

Table 4.5: Compared with state-of-the-art methods on MOT15-17 testing sets with private detection, where $\uparrow(\downarrow)$ indicates that the larger(smaller) the value is, the better the performance.

4.3.6 Evaluation with Oracle Detection

In the above experiments, we applied the ReID-dominated data association to imperfect detection. Corresponding to real scenarios, the perfect detection is challenging to be obtained and we need to make our data association work on noisy detection. However, to focus on the data association study itself, we also would like to further experience our data association on oracle detection, to verify the upbound performance of our data association proposal.

Approach	sMOTA (%) \uparrow		MOTSA (%) \uparrow		MT (%) \uparrow		ML (%) \downarrow		# ID Sw. \downarrow	
	car	pedestrian	car	pedestrian	car	pedestrian	car	pedestrian	car	pedestrian
TrackR-CNN [52] (CVPR'19)	67.0	47.3	79.6	66.1	74.9	45.6	2.3	13.3	692	481
MOTSFusion [125] (ICRA'20)	75.0	58.7	84.1	72.9	66.1	47.4	6.2	15.6	201	279
PointTrack [126] (ECCV'20)	78.5	61.5	90.9	76.5	90.8	48.9	0.6	9.3	346	176
Our offline data association + Detectron2	78.0	66.6	90.4	81.9	90.8	61.5	0.6	5.2	533	150

Table 4.6: KITTI-MOTS Results. Red and Blue represent the first and second results, respectively.

Datasets	Approach	MOTA (%) \uparrow	IDF1 (%) \uparrow	MT (%) \uparrow	ML (%) \downarrow	# ID Sw. \downarrow
MOT15-17 Train	Oracle Tracklets	100.0	100.0	100.0	0.0	0
MOT15 Train	DeepSORT [56]	95.1	86.5	98.4	0.8	162
	Our Online	97.6	90.1	100.0	0.0	98
	Our Offline	98.7	92.2	100.0	0.0	50
MOT16 Train	DeepSORT [56]	98.2	88.8	99.1	0.2	653
	Our Online	99.1	91.5	100.0	0.0	540
	Our Offline	99.6	95.8	100.0	0.0	268
MOT17 Train	DeepSORT [56]	98.4	89.4	99.2	0.2	1934
	Our Online	99.1	92.4	100.0	0.0	1698
	Our Offline	99.8	96.3	100.0	0.0	809

Table 4.7: Evaluation performance on MOT15-17 train sets. All methods use the oracle detection of MOT15-17 train sets. The appearance encoder is trained on Market-1501 dataset [9].

We utilized the training sets of MOT15-17 datasets to access their ground-truth detection and tracklets. To treat the original training sets as our new testing sets, we trained our appearance encoder on an extra ReID dataset — Market-1501 dataset [9]. There could be domain gaps between the Market-1501 dataset and MOT15-17 datasets, but we ignore them here. By applying both online and offline ReID-dominated data association on the oracle detection of MOT15-17

datasets, we obtained the results as shown in Table 4.7. We supposed the oracle tracklets (*i.e.*, MOT ground truth) perfected all MOT metrics (*e.g.*, MOTA). When oracle detection was given, both of our online and offline data associations can generate nearly perfect results. Nonetheless, unlike the experiment on the BDD100K MOT dataset, the margin between our solutions and DeepSORT [56] is considerably diminished when oracle detection is given in MOT15-17. This result indicates, when high-quality object detection can be easily achieved, we may not increase the complicity of the data association for just improving a limited performance. However, when the object detection is noisy and the object movement is complicated, a more powerful data association could be considered. This gives insight to properly select a data association strategy in real practice.

4.4. Discussion

We proposed a ReID-dominated data association to handle complicated object movements in MOT tasks. In our online approach, the appearance feature generated by a ReID model dominates the matching process, while the motion feature is cast to adaptive temporal-spatial constraints. In our offline approach, by utilizing improved appearance features, a modified Hierarchical Clustering is applied to complete broken tracklets generated in our online approach. On multiple MOT/MOTS datasets, our experimental results cast a new light on fusing the appearance feature and the motion feature: using the ReID-dominated data association has decisive advantages over previous works, it copes much better with more complicated object movements and a better online data association performance can be achieved. Currently, since only the RGB visual feature is applied in our approach, it may not be robust to the poor illumination condition. For future research, we recommend looking into how to include more visual features (*e.g.*, the infrared feature) to our ReID-dominated data association, when object movements are complicated and the illumination condition is poor. The impact of applying such MOT methods in ASAD framework will be discussed in the next chapter.

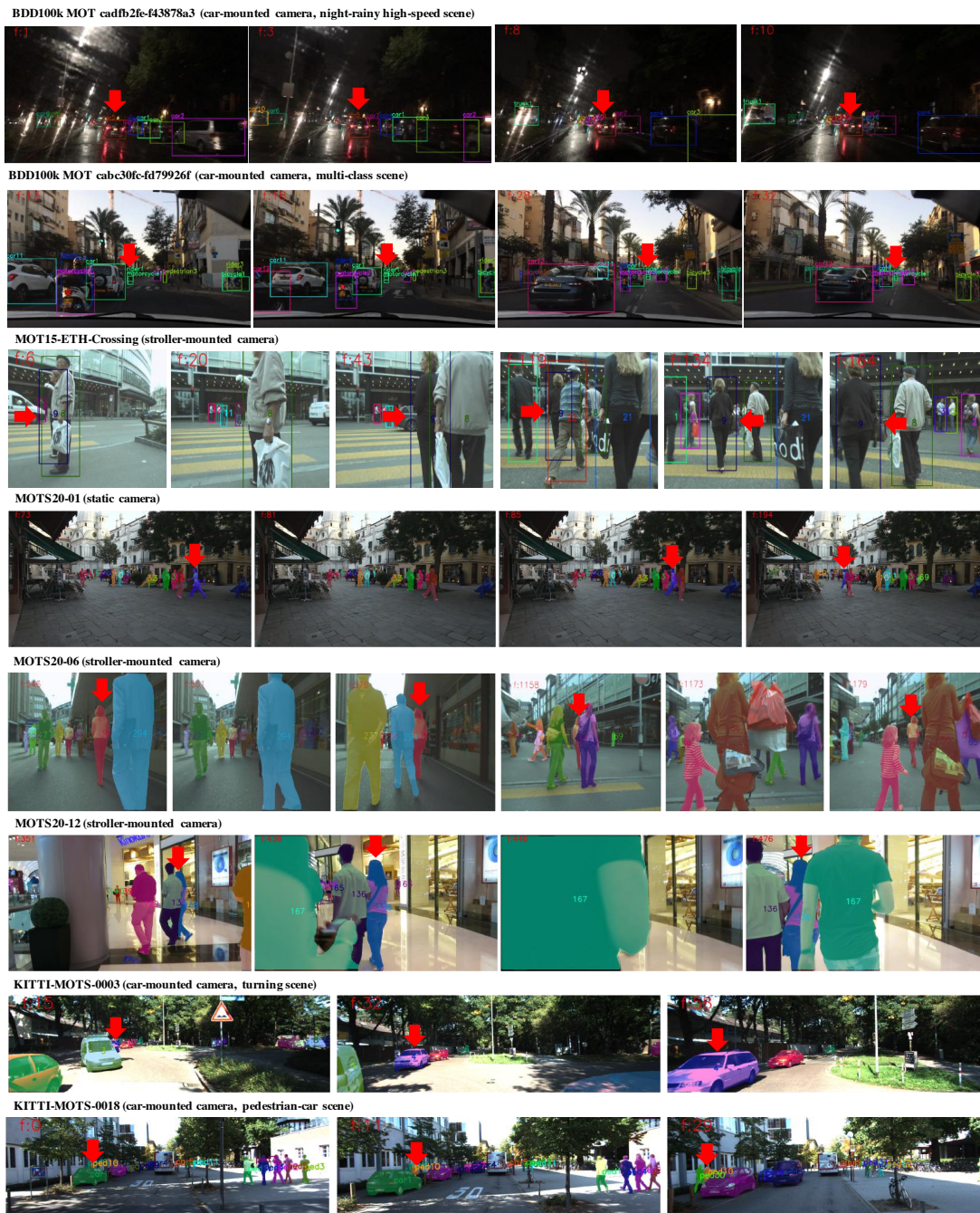


Figure 4.7: Qualitative results of ReID-dominated data association. We performed online approach on BDD100K MOT dataset and offline approach on others. Red arrows indicate the identical instance, which shows that the targets are tracked robustly in diverse scenarios.

Chapter 5

ASAD Benchmark

We set up the first benchmark for the ASAD study. Experiments are conducted on our A-AVA dataset and evaluated by our ASAD metrics.

5.1. ASAD Framework

As we have discussed in Chapter 2, some SAD models, such as ROAD [2], AlphAction [7], and ACAM [8], are consist of Multiple Object Tracking (MOT) and Action Classification (AC) modules. They could generate ASAD results but were evaluated by the SAD protocol in the original works. Based on the evaluation protocol of SAD, the annotation of actor identity may not be provided and the actor identification has not been evaluated. In other words, there is no clear boundary between ASAD and SAD in terms of the method, their difference more lies in the data annotation and evaluation protocols.

Without changing the basic structure, letting the above SAD methods to output actor identities with their original outputs can make ASAD frameworks. In this study, we let the off-the-shelf SAD methods, as AlphAction [7] and ACAM [8], to output actor identities that are generated by their MOT module. In this manner, they can perform as ASAD frameworks. In Figure 5.1, we summarize the basic structure of ASAD framework that is adapted from AlphAction [7] and ACAM [8]. In this chapter, an ASAD framework takes RGB videos as the input and outputs the bounding boxes, unique actor identity, and actions of each actor.

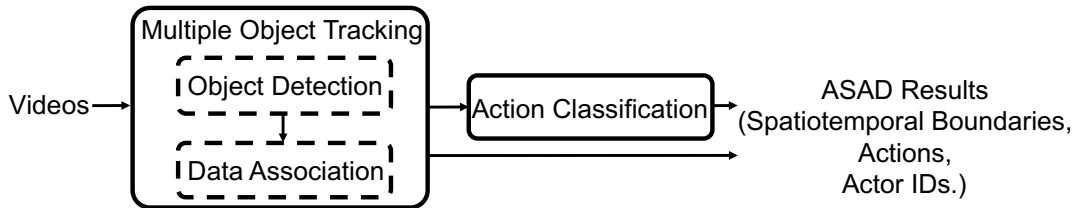


Figure 5.1: Overview of the basic ASAD framework.

5.2. ASAD Experiments

By using our A-AVA dataset, the performance of spatiotemporal detection, action classification, and actor identification can be jointly evaluated. We first evaluate the off-the-shelf SAD methods AlphAction [7] and ACAM [8] with their actor identification generation. We show the result of using our ASAD evaluation metrics in Table 5.1. It can be noticed that the action identification performance is unsatisfactory and become the bottleneck to obtain satisfactory ASAD results.

Approaches	Actor Detection Evaluation	Action Classification Evaluation	Actor Identification Evaluation			
	AP@0.5 (%) \uparrow	HL@0.5 (0~1) \downarrow	IDF1 (%) \uparrow	MT (%) \uparrow	ML (%) \downarrow	# ID Sw. \downarrow
AlphAction [7]	72.4	0.06	60.4	67.3	10.5	413
ACAM [8]	70.1	0.07	56.7	58.0	15.8	597

Table 5.1: Results of the default ASAD-adapted frameworks on our A-AVA dataset by using our ASAD evaluation metrics, where $\uparrow(\downarrow)$ indicates that the larger(smaller) the value is, the better the performance.

To solve this issue, we replace the default MOT module of AlphAction [7] and ACAM [8] by our MOT methods (Chapter 4). In Table 5.2, we report two of our MOT approaches that led to better actor identification result on our A-AVA dataset. The first one is the adoption of our online MOT method (Section 4.2.2) to AlphAction [7] and ACAM [8]. We found that the values of IDF1 and MT slightly increase while the values of ML and ID Sw. lightly dropped down. It shows our online MOT method is more robust in our A-AVA dataset. The second one is the adoption of our offline MOT method (Section 4.2.2) to AlphAction [7] and ACAM [8], which gave us a further gain in IDF1 and ML over their original MOT module, and reduced the ML and ID Sw. Such results demonstrate the

effectiveness of applying our offline MOT in the ASAD framework.

Approaches	Actor Detection Evaluation	Action Classification Evaluation	Actor Identification Evaluation			
	AP@0.5 (%) \uparrow	HL@0.5 (0~1) \downarrow	IDF1 (%) \uparrow	MT (%) \uparrow	ML (%) \downarrow	# ID Sw. \downarrow
AlphAction [7] w/ the original MOT	72.4	0.06	60.4	67.3	10.5	413
AlphAction [7] w/ our online MOT	72.4	0.06	60.8	69.5	10.5	459
AlphAction [7] w/ our Offline MOT	72.4	0.06	71.4	88.4	5.2	273
ACAM [8] w/ the original MOT	70.1	0.07	56.7	58.0	15.8	597
ACAM [8] w/ our online MOT	70.1	0.07	58.4	67.4	11.6	520
ACAM [8] w/ our Offline MOT	70.1	0.07	70.2	86.3	6.3	288

Table 5.2: Comparison of using different MOT module in ADAD frameworks. We utilize our A-AVA dataset and ASAD evaluation metrics, where $\uparrow(\downarrow)$ indicates that the larger(smaller) the value is, the better the performance.

In the above experiment, the actor detection is obtained from an object detector, which inevitably contains noise and error. Next, we would like to experience the ASAD framework with the oracle actor detection, to verify the upbound performance of our actor identification proposal. The results are shown in Table 5.3. Compared with the results of Table 5.2, the actor identification improvement is not significant by using the oracle actor detection. Such results indicate that actor detection may not be the main factor that hampers the actor identification performance. To generate a better actor identification result, we may need to focus on the data association strategy in MOT, as we have explored in Chapter 4.

By using the oracle actor detection, we illustrate some of actor identification result for visualization in Figures 5.2 and 5.3. In those figures, the identical actor is located by bounding boxes of the same color crossing frames. Whenever the viewpoint suddenly changed in videos, it is challenging to track the correct actor identities by the original MOT module in ACAM [8]. Our online MOT solution is more robust in such scenarios but cannot handle all cases. Our offline MOT approach can be used in MOT processing when the latency is allowed. Compared with the online approach, our offline MOT approach not only can access the global information of observations but also fine-tune the appearance encoder on

Approaches	Actor Identification Evaluation			
	IDF1 (%) \uparrow	MT (%) \uparrow	ML (%) \downarrow	# ID Sw. \downarrow
ACAM MOT module	62.1	61.1	12.6	463
AlphaAction MOT module	64.8	91.6	4.2	386
Our Online MOT	65.1	94.7	1.1	425
Our Offline MOT	75.3	100.0	0.0	237

Table 5.3: Results of actor identification on our A-AVA dataset with oracle actor detection, where $\uparrow(\downarrow)$ indicates that the larger(smaller) the value is, the better the performance.

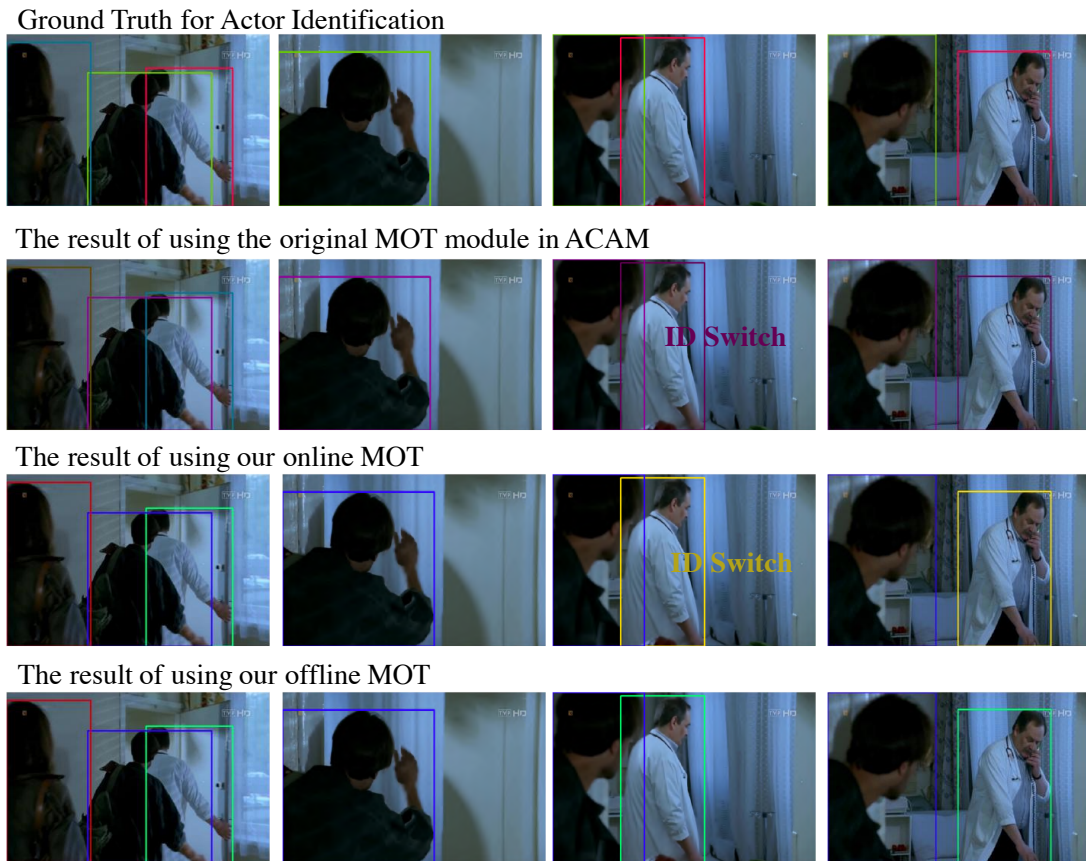


Figure 5.2: Visualization of actor identification results in the A-AVA dataset (1/2). The identical actor is located by bounding boxes of the same color.

Ground Truth for Actor Identification



The result of using the original MOT module in ACAM



The result of using our online MOT



The result of using our offline MOT



Figure 5.3: Visualization of actor identification results in the A-AVA dataset (2/2). The identical actor is located by bounding boxes of the same color.

target videos with pseudo tracklet labels. Consequently, the tracking performance can be significantly improved. Although it's noteworthy to achieve better actor identification results with our offline MOT approach, it made mistakes in actor occlusion scenarios where the actor can be easily tracked by humans (Figure 5.3). We still have space to improve the MOT method for better actor identification.

5.3. Discussion

For static camera recording, motion consistency is an important cue for data association. In contrast, for non-static camera recording, the motion consistency

assumption could be failed. This issue frequently happens in the movies and phone-recorded videos (Figure 5.4). The original MOT strategies that were applied in AlphAction [7] and ACAM [8] may over rely on the motion consistency and therefore cause failure cases in our A-AVA dataset. Our online MOT solution alleviates this issue by determining the correspondence between observations more by their appearance similarity. Moreover, our offline MOT solution utilizes the global information to further reduce ID switches and generate robust actor identification results.

Static camera recording:



Non-static camera recording:



Figure 5.4: The difference of motion consistency in static camera recording videos and non-static camera recording videos.

Chapter 6

Conclusions and Future Work

6.1. Conclusions

In this thesis, we introduced a novel task, Actor-identified Spatiotemporal Action Detection (ASAD), which marks the first effort in the computer vision community to jointly study spatiotemporal boundaries, actor identities, and corresponding actions. ASAD is ideal for action recognition applications when multiple actors are included, such as Human-computer Interaction, basketball/soccer games, and grocery operations monitoring, etc. We believe considering actor identification with spatiotemporal action detection could promote the research on video understanding and beyond. We are excited to engage with the research community to explore ASAD deeper.

In summary, we have made following contributions:

- To study ASAD, we are excited to offer a corresponding A-AVA dataset. A-AVA dataset contains 47 videos for training and 30 videos for testing. Be the same as the AVA dataset [3], there are 80 action categories in the A-AVA dataset, and, every 25 frames (*i.e.*, around 1 second), the annotation is given once. In the A-AVA dataset, the spatiotemporal boundaries, actor identities, and corresponding actions are all annotated. As the first dataset that is specifically designed for the ASAD study, the A-AVA dataset covers a rich diversity of video scenes, as indoor and outdoor, different times of the day, various actor scales, and more. Those properties are not available

in the previous dataset (*i.e.*, Okutama dataset [6]). Our A-AVA dataset has bridged the gap between the SAD dataset and the actor identification dataset.

- In existing SAD evaluation metrics, the evaluation of multi-label action classification and actor identification are not available. To evaluate the performance of ASAD, we also proposed ASAD evaluation metrics by considering multi-label actions and actor identification. We suggest evaluating ASAD from three aspects and then consider their overall performance. The three aspects include Spatial Detection Evaluation, Actor Identification Evaluation, and Multi-label Action Classification Evaluation. We provided the first evaluation metrics in ASAD such a complicated task.
- Since current MOT performance could be the bottleneck to obtain satisfactory ASAD results. In Chapter 4, we improved the data association strategies in MOT to boost the MOT performance. In Chapter 5, we proved that applying our MOT method to the ASAD framework can significantly improve the actor identification result and may slightly improve the action classification performance. Except for the ASAD dataset, our MOT method also achieved the state-of-the-art performance on multiple public MOT datasets and demonstrated its effectiveness by winning two MOT-related challenges, *i.e.*, BDD100K MOT of the CVPR’20 WAD Workshop, and Track 1 of the CVPR’20 MOTS Workshop.

Besides the above success, it is important to note that our ASAD study also suffers some limitations:

- Considering the high annotation cost, the size of our proposed ASAD dataset is still relatively small. Meanwhile, since the definition of action labels could be ambiguous, the action annotation may not be accurate. For instance, it is difficult to judge the boundary between “walk” and “running” in the continuous temporal domain. In addition, without including the audio information, it is challenging to decide who is speaking, and, whether actors are chatting or quarreling. Such issues may impair the ASAD study. To cope with this issue, it is necessary to perform high-quality annotations

with more annotators involved. Furthermore, encouraging the community to join the ASAD study may help to improve the ASAD dataset.

- Because evaluating the ASAD result is complicated, we separately evaluated spatial detection, actor identification, and multi-label action classification. Consequently, the overall ASAD performance is represented by multiple metric values. However, in an ideal case, we hope to utilize a single metric value to represent the overall ASAD performance. Considering that each of our ASAD metrics (*e.g.*, HL@0.5) is obtained from a complex formula, it is challenging to integrate them into a single metric value. To find a solution, further exploration is needed. Since we have raised this question in the ASAD task, it might be solved in future works.
- Although we have made a significant improvement in the actor identification by using our MOT method, the result still has a large margin to be perfect. Therefore, further detailed investigation of the remaining errors and further exploration on MOT usage shall be performed to find a better solution and future advancement in MOT research.

6.2. Future Works

This thesis is not the end, but rather the starting steps, there are more potential works worth exploring. Besides the achievements we summarized in the above section, we would like to introduce some remaining issues in our ASAD and discuss the possible solutions. The roadmap to Actor-identified Spatiotemporal Action Detection (ASAD) is outlined in Figure 6.1.

6.2.1 Short-term Future Work

6.2.1.1 Addressing the Limitations in Our Contributions

In the above section, we have analyzed the limitations in our contributions. For the next step, we will start to find solutions to alleviate them and construct a better ASAD benchmark.

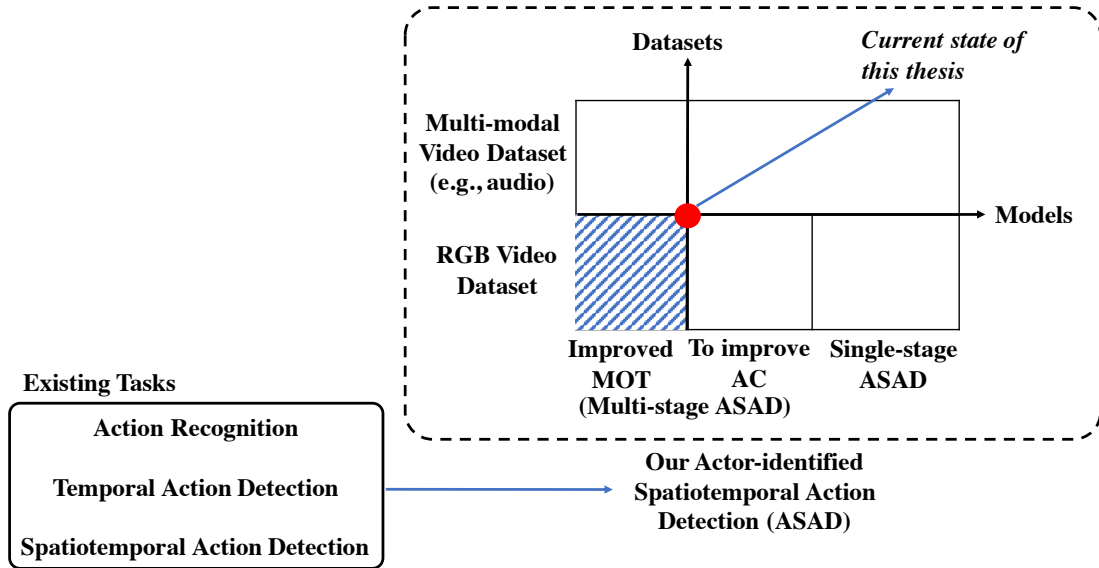


Figure 6.1: Roadmap to Actor-identified Spatiotemporal Action Detection (ASAD).

6.2.1.2 Improving the Action Classification

We supposed that Multiple Object Tracking (MOT) and Action Classification (AC) are two fundamental elements to approach ASAD. In this thesis, although we have significantly improved the MOT module, it can be observed that the action classification performance may not be satisfactory in Chapter 5. To improve the AC module in the ASAD framework, the difficulties may arise from two aspects: i) how to model spatiotemporal interaction without exponentially increasing the computation costs as time goes; ii) how to learn the interaction representations since the interaction labels could be sparse in a huge spatiotemporal domain.

For potential solutions, we may explore the adaptation of Transformer [127] in the action classification of ASAD. Using the Transformer to model the sparse action interaction in the huge spatiotemporal domain might be efficient.

6.2.2 Middle-term Future Work

6.2.2.1 Integrating MOT and AC

In our current ASAD framework, multiple object tracking (MOT) and action classification (AC) are independent, although they work together to approach ASAD. Setting MOT and AC independently enables the flexible training strategy, which reduces the annotation cost and simplifies the data augmentation. However, it might be more efficient by unifying the MOT and AC into a single network model and jointly training them. In the possible solution, integrating MOT and AC may avoid redundant computation and may significantly improve the speed of the ASAD framework.

6.2.2.2 Integrating 3D MOT and Fine-grained Hand Action Detection

While we have shown some encouraging works (*e.g.*, 3D MOT) in Chapter Appendix, much work remains to be done to adapt those works to our ASAD framework. In the future, we may integrate 3D MOT, fine-grained hand action detection, skeleton-based action classification, and drone-recorded videos in our ASAD framework (Figure 6.2).

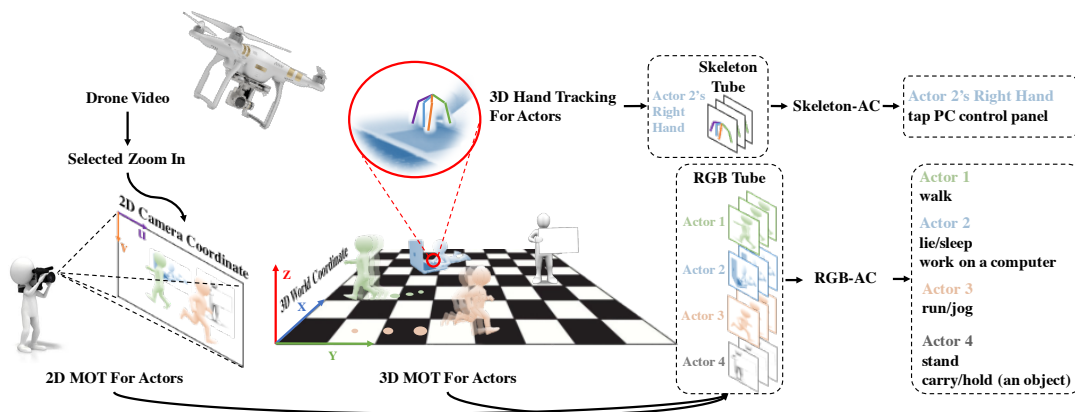


Figure 6.2: Integrating 3D MOT, fine-grained hand action detection, skeleton-based action classification, and drone-recorded videos in our ASAD framework.

6.2.3 Long-term Future Work

In this work, only RGB data is utilized in the ASAD study, however, to make ASAD more useful, richer multi-modal features could be incorporated in the ASAD task. In movies, for instance, the audio may also be included. The audio not only can be used to assist actor identification [128], but also benefit for the action classification [129, 130]. For example, we can utilize the audio to decide who is speaking, and, whether actors are chatting or quarreling. Besides, after getting the permission of users, the mobile data (*e.g.*, inertial measurement unit data) can be employed in action classification [131]. Since ASAD is a new task, the corresponding multi-modal data does not exist, and therefore, we need to create a multi-modal ASAD dataset by considering the audio data and mobile data. For the evaluation, our proposed ASAD metrics can be directly applied since the outputs of multi-modal ASAD are identical to the RGB-based ASAD dataset.

Acknowledgement

I am sincerely grateful to all of my advisers, as Prof. Satoshi Nakamura, Assoc. Prof. Sakriani Sakti, and Senior Lecturer Yang Wu. I want to express my gratitude for the constant support and encouragement from them, despite my submissions got plenty of rejections. I very much appreciate the help from my co-authors, Asst. Prof. Zheng Wang and Prof. Norimichi Ukita. I consider it an honor to work with them on the path towards my Ph.D. degree. I would like to extend my sincere thanks to Prof. Kiyoshi Kiyokawa. As my co-supervisor, he had spent much time listening to my research and gave me useful feedback. I also want to say a lot of thanks to Prof. Takeo Kanade. Since my master course, Prof. Takeo Kanade took a long trip to NAIST and had meetings with us, I very much appreciate his insightful comments and encouragement.

Besides, I would like to thank all my lab members, who always take their pleasure to help me in my life and study. In addition, I would like to thanks the JASSO scholarship, the Donghua Education Scholarship, and the tuition exemption of NAIST. These financial supports significantly help my study in Japan. Without them, I may not be able to focus on my research. Many thanks again.

Above ground, My biggest thanks to my family. This thesis is dedicated to my wife and parents who have always stood by me and showered me with their love.

References

- [1] Matthew Hutchinson and Vijay Gadepally. Video action understanding: A tutorial. *arXiv preprint arXiv:2010.06647*, 2020.
- [2] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3637–3646, 2017.
- [3] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6047–6056, 2018.
- [4] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. *CoRR*, *abs/1705.08421*, 4, 2017.
- [5] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.
- [6] Mohammadamin Barekatin, Miquel Martí, Hsueh-Fu Shih, Samuel Murray, Kotaro Nakayama, Yutaka Matsuo, and Helmut Prendinger. Okutama-action: An aerial view video dataset for concurrent human action detection.

- In *1st Joint BMTT-PETS Workshop on Tracking and Surveillance, CVPR*, pages 1–8, 2017.
- [7] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu. Asynchronous interaction aggregation for action detection. In *Proceedings of the European conference on computer vision (ECCV)*, 2020.
- [8] Oytun Ulutan, Swati Rallapalli, Mudhakar Srivatsa, Carlos Torres, and BS Manjunath. Actor conditioned attention maps for video action detection. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 527–536, 2020.
- [9] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015.
- [10] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016.
- [11] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, Xiaowei Zhao, and Tae-Kyun Kim. Multiple object tracking: A literature review. *arXiv preprint arXiv:1409.7618*, 2014.
- [12] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011.
- [13] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.
- [14] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolu-

- tional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.
- [15] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019.
- [16] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.
- [17] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.
- [18] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016.
- [19] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 6202–6211, 2019.
- [20] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.
- [21] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.
- [22] H. Idrees, A. R. Zamir, Y. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah. The thumos challenge on action recognition for videos “in the wild”. *Computer Vision and Image Understanding*, 155:1–23, 2017.

- [23] Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. Hacs: Human action clips and segments dataset for recognition and temporal localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8668–8678, 2019.
- [24] Khurram Soomro, Haroon Idrees, and Mubarak Shah. Action localization in videos through context walk. In *Proceedings of the IEEE international conference on computer vision*, pages 3280–3288, 2015.
- [25] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.
- [26] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE international conference on computer vision*, pages 5783–5792, 2017.
- [27] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Cascaded boundary regression for temporal action detection. In *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*. BMVA Press, 2017.
- [28] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1130–1139, 2018.
- [29] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3889–3898, 2019.
- [30] Philippe Weinzaepfel, Xavier Martin, and Cordelia Schmid. Human action localization with sparse spatial supervision. *arXiv preprint arXiv:1605.05197*, 2016.

- [31] Pascal Mettes, Jan C Van Gemert, and Cees GM Snoek. Spot on: Action localization from pointly-supervised proposals. In *European conference on computer vision*, pages 437–453. Springer, 2016.
- [32] Ang Li, Meghana Thotakuri, David A Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. The ava-kinetics localized human actions video dataset. *arXiv preprint arXiv:2005.00214*, 2020.
- [33] Yooyoung Yooyoung, Jon Fiscus, Afzal Godil, David Joy, Andrew Delgado, and Jim Golden. Actev18: Human activity detection evaluation for extended videos. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 1–8. IEEE, 2019.
- [34] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 759–768, 2015.
- [35] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *Proceedings of the IEEE international conference on computer vision*, pages 3164–3172, 2015.
- [36] Hongyuan Zhu, Romain Vial, and Shijian Lu. Tornado: A spatio-temporal convolutional regression network for video action proposal. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5813–5821, 2017.
- [37] Rui Hou, Chen Chen, and Mubarak Shah. Tube convolutional neural network (t-cnn) for action detection in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 5822–5831, 2017.
- [38] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4405–4413, 2017.
- [39] Fan Yang, Sakriani Sakti, Yang Wu, and Satoshi Nakamura. A framework for knowing who is doing what in aerial surveillance videos. *IEEE Access*, 7:93315–93325, 2019.

- [40] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. A better baseline for ava. *arXiv preprint arXiv:1807.10066*.
- [41] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S Davis, and Jan Kautz. Step: Spatio-temporal progressive learning for video action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 264–272, 2019.
- [42] Junting Pan, Siyu Chen, Zheng Shou, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. *arXiv preprint arXiv:2006.07976*, 2020.
- [43] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. Actions as moving points. *arXiv preprint arXiv:2001.04608*, 2020.
- [44] Yuxi Li, Weiyao Lin, John See, Ning Xu, Shugong Xu, Ke Yan, and Cong Yang. Cfad: Coarse-to-fine action detector for spatiotemporal action localization. In *European Conference on Computer Vision*, pages 510–527. Springer, 2020.
- [45] Shinchi Satoh. Towards actor/actress identification in drama videos. In *Proceedings of the seventh ACM international conference on Multimedia (Part 2)*, pages 75–78, 1999.
- [46] Mengdi Xu, Xiaotong Yuan, Jialie Shen, and Shuicheng Yan. Cast2face: Character identification in movie with actor-character correspondence. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 831–834, 2010.
- [47] Qingqiu Huang, Yu Xiong, and Dahua Lin. Unifying identification and context learning for person recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2217–2225, June 2018.
- [48] Gioele Ciaparrone, Francisco Luque Sánchez, Siham Tabik, Luigi Troiano, Roberto Tagliaferri, and Francisco Herrera. Deep learning in video multi-object tracking: A survey. *Neurocomputing*, 381:61–88, 2020.

- [49] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2020.
- [50] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942 [cs]*, 2015.
- [51] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, 2016.
- [52] Paul Voigtlaender, Michael Krause, Aljoša Ošep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. MOTS: Multi-object tracking and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [53] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.
- [54] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
- [55] Morton Klein. A primal method for minimal cost flows with applications to the assignment and transportation problems. *Management Science*, 14(3):205–220, 1967.
- [56] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and real-time tracking with a deep association metric. *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649, 2017.
- [57] Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan. Poi: Multiple object tracking with high performance detection and appearance feature. In *ECCV Workshops*, 2016.

- [58] Jakub Segen. A camera-based system for tracking people in real time. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 3, pages 63–67. IEEE, 1996.
- [59] Séverine Dubuisson. Recursive clustering for multiple object tracking. In *2006 International Conference on Image Processing*, pages 2805–2808. IEEE, 2006.
- [60] Margrit Betke, Diane E Hirsh, Angshuman Bagchi, Nickolay I Hristov, Nicholas C Makris, and Thomas H Kunz. Tracking large variable numbers of objects in clutter. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- [61] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3539–3548, 2017.
- [62] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6036–6046, 2018.
- [63] Yu Wu, Yutian Lin, Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5177–5186, 2018.
- [64] Maryam Babaei, Ali Athar, and Gerhard Rigoll. Multiple people tracking using hierarchical deep tracklet re-identification. *arXiv preprint arXiv:1811.04091*, 2018.
- [65] Wei Qu, Dan Schonfeld, and Magdi Mohamed. Real-time distributed multi-object tracking using multiple interactive trackers and a magnetic-inertia potential model. *IEEE Transactions on Multimedia*, 9(3):511–519, 2007.
- [66] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016.

- [67] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information. In *International Workshop on Traffic and Street Surveillance for Safety and Security at IEEE AVSS 2017*, Lecce, Italy, August 2017.
- [68] Peng Dai, Xue Wang, Weihang Zhang, and Junfeng Chen. Instance segmentation enabled hybrid data association and discriminative hashing for online multi-object tracking. *IEEE Transactions on Multimedia*, 21(7):1709–1723, 2018.
- [69] Jing Li, Lisong Wei, Fangbing Zhang, Tao Yang, and Zhaoyang Lu. Joint deep and depth for object-level segmentation and stereo tracking in crowds. *IEEE Transactions on Multimedia*, 21(10):2531–2544, 2019.
- [70] Zeyu Fu, Federico Angelini, Jonathon Chambers, and Syed Mohsen Naqvi. Multi-level cooperative fusion of gm-phd filters for online multiple human tracking. *IEEE Transactions on Multimedia*, 21(9):2277–2291, 2019.
- [71] Xiaolong Jiang, Peizhao Li, Yanjing Li, and Xiantong Zhen. Graph neural based end-to-end data association framework for online multiple-object tracking. *arXiv preprint arXiv:1907.05315*, 2019.
- [72] Cong Ma, Yuan Li, Fan Yang, Ziwei Zhang, Yueqing Zhuang, Huizhu Jia, and Xiaodong Xie. Deep association: End-to-end graph-based learning for multiple object tracking with conv-graph neural network. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pages 253–261, 2019.
- [73] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *arXiv preprint arXiv:1909.12605*, 2019.
- [74] Jiahe Li, Xu Gao, and Tingting Jiang. Graph networks for multiple object tracking. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 719–728, 2020.
- [75] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6247–6257, 2020.

- [76] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris M. Kitani. Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [77] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *ECCV*, 2020.
- [78] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. A simple baseline for multi-object tracking. *arXiv preprint arXiv:2004.01888*, 2020.
- [79] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Tracking by prediction: A deep generative model for multi-person localisation and tracking. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1122–1132. IEEE, 2018.
- [80] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019.
- [81] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–8, 2008.
- [82] Longyin Wen, Wenbo Li, Junjie Yan, Zhen Lei, Dong Yi, and Stan Z Li. Multiple target tracking based on undirected hierarchical relation hypergraph. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1282–1289, 2014.
- [83] Shun Zhang, Jinjun Wang, Zelun Wang, Yihong Gong, and Yuehu Liu. Multi-target tracking by learning local-to-global trajectory models. *Pattern Recognition*, 48(2):580–590, 2015.
- [84] Liqian Ma, Siyu Tang, Michael J. Black, and Luc Van Gool. Customized multi-person tracker. In *ACCV*, 2018.

- [85] Andrea Hornakova, Roberto Henschel, Bodo Rosenhahn, and Paul Swo-boda. Lifted disjoint paths with application in multiple object tracking. In *The 37th International Conference on Machine Learning (ICML)*, pages 1–12, July 2020.
- [86] H. Luo, W. Jiang, Y. Gu, F. Liu, X. Liao, S. Lai, and J. Gu. A strong baseline and batch normalization neck for deep person re-identification. *IEEE Transactions on Multimedia*, pages 1–1, 2019.
- [87] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4733. IEEE, 2017.
- [88] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 695–712, 2018.
- [89] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541, 2017.
- [90] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 786–792, 2018.
- [91] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12026–12035, 2019.
- [92] Fan Yang, Yang Wu, Sakriani Sakti, and Satoshi Nakamura. Make skeleton-based action recognition model smaller, faster and better. In *Proceedings of the ACM Multimedia Asia*, pages 1–6. 2019.

- [93] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7024–7033, 2018.
- [94] An Yan, Yali Wang, Zhifeng Li, and Yu Qiao. Pa3d: Pose-action 3d machine for video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7922–7931, 2019.
- [95] Mohammadreza Zolfaghari, Gabriel L Oliveira, Nima Sedaghat, and Thomas Brox. Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2904–2913, 2017.
- [96] Xueyang Wang, Xiya Zhang, Yinheng Zhu, Yuchen Guo, Xiaoyun Yuan, Liuyu Xiang, Zerun Wang, Guiguang Ding, David Brady, Qionghai Dai, et al. Panda: A gigapixel-level human-centric video dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3268–3278, 2020.
- [97] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. Tao: A large-scale benchmark for tracking any object. *arXiv preprint arXiv:2005.10356*, 2020.
- [98] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.
- [99] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [100] Jan Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele. What makes for effective detection proposals? *IEEE transactions on pattern analysis and machine intelligence*, 38(4):814–830, 2015.

- [101] Eva Gibaja and Sebastián Ventura. A tutorial on multilabel learning. *ACM Computing Surveys*, 47(3):52, 2015.
- [102] David F Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016.
- [103] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5167–5176, 2018.
- [104] Anton Milan, Seyed Hamid Rezaatofghi, Anthony R. Dick, Ian D. Reid, and Konrad Schindler. Online multi-target tracking using recurrent neural networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4225–4232. AAAI Press, 2017.
- [105] Weiqiang Li, Jiatong Mu, and Guizhong Liu. Multiple object tracking with motion and appearance cues. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.
- [106] Xin Li, Kejun Wang, Wei Wang, and Yang Li. A multiple object tracking method using kalman filter. *The 2010 IEEE International Conference on Information and Automation*, pages 1862–1866, 2010.
- [107] Zheng Wang, Ruimin Hu, Chao Liang, Yi Yu, Junjun Jiang, Mang Ye, Jun Chen, and Qingming Leng. Zero-shot person re-identification via cross-view consistency. *IEEE Transactions on Multimedia*, 18(2):260–272, 2016.
- [108] Zheng Wang, Junjun Jiang, Yi Yu, and Shin’ichi Satoh. Incremental re-identification by cross-direction and cross-ranking adaption. *IEEE Transactions on Multimedia*, 21(9):2376–2386, 2019.
- [109] Qiaokang Xie, Wengang Zhou, Guo-Jun Qi, Qi Tian, and Houqiang Li. Progressive unsupervised person re-identification by tracklet association with spatio-temporal regularization. *IEEE Transactions on Multimedia*, 2020.

- [110] Zelong Zeng, Zhixiang Wang, Zheng Wang, Yinqiang Zheng, Yung-Yu Chuang, and Shin'ichi Satoh. Illumination-adaptive person re-identification. *IEEE Transactions on Multimedia*, 2020.
- [111] Shizhou Zhang, Qi Zhang, Yifei Yang, Xing Wei, Peng Wang, Bingliang Jiao, and Yanning Zhang. Person re-identification in aerial imagery. *IEEE Transactions on Multimedia*, 2020.
- [112] Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu. Hierarchical clustering: Objective functions and algorithms. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 378–397. SIAM, 2018.
- [113] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [114] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [115] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(Mar):1109–1135, 2010.
- [116] Hao Luo, Wei Jiang, Youzhi Gu, Fuxu Liu, Xingyu Liao, Shenqi Lai, and Jianyang Gu. A strong baseline and batch normalization neck for deep person re-identification. *IEEE Transactions on Multimedia*, pages 1–13, 2019.
- [117] C Chang and L Youens. Measurement correlation for multiple sensor tracking in a dense target environment. *IEEE Transactions on Automatic Control*, 27(6):1250–1252, 1982.
- [118] Hao Su, Yaran Chen, Shiwen Tong, and Dongbin Zhao. Real-time multiple object tracking based on optical flow. In *2019 9th International Conference on Information Science and Technology (ICIST)*, pages 350–356. IEEE, 2019.

- [119] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [120] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2, 2019.
- [121] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.
- [122] Fan Yang, Xin Chang, Chenyu Dang, Ziqiang Zheng, Sakriani Sakti, Satoshi Nakamura, and Yang Wu. Remots: Self-supervised refining multi-object tracking and segmentation, 2020.
- [123] Bo Pang, Yizhuo Li, Yifan Zhang, Muchen Li, and Cewu Lu. Tubetk: Adopting tubes to track multi-object in a one-step training model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6308–6318, 2020.
- [124] Jinlong Peng, Changan Wang, Fangbin Wan, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *Proceedings of the European Conference on Computer Vision*, 2020.
- [125] Jonathon Luiten, Tobias Fischer, and Bastian Leibe. Track to reconstruct and reconstruct to track. *IEEE Robotics and Automation Letters*, 5(2):1803–1810, 2020.
- [126] Zhenbo Xu, Wei Zhang, Xiao Tan, Wei Yang, Huan Huang, Shilei Wen, Errui Ding, and Liusheng Huang. Segment as points for efficient online multi-object tracking and segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–18, 2020.
- [127] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

- [128] Joseph Roth, Sourish Chaudhuri, Ondrej Klejch, Radhika Marvin, Andrew Gallagher, Liat Kaver, Sharadh Ramaswamy, Arkadiusz Stopczynski, Cordelia Schmid, Zhonghua Xi, et al. Ava active speaker: An audio-visual dataset for active speaker detection. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4492–4496. IEEE, 2020.
- [129] Lili Nurliyana Abdullah and Shahrul Azman Mohd Noah. Integrating audio visual data for human action detection. In *2008 Fifth International Conference on Computer Graphics, Imaging and Visualisation*, pages 242–246. IEEE, 2008.
- [130] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10457–10467, 2020.
- [131] Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th International Symposium on Wearable Computers*, pages 108–109. IEEE, 2012.
- [132] Markus Oberweger and Vincent Lepetit. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *ICCV workshop*, volume 840, page 2, 2017.
- [133] Lihao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 5, 2017.
- [134] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.
- [135] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose

- estimation from a single depth map. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5079–5088, 2018.
- [136] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Dense 3d regression for hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2018.
- [137] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5):169, 2014.
- [138] Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. Fast and robust hand tracking using detection-guided optimization. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3213–3221. IEEE, 2015.
- [139] Jonathan Taylor, Vladimir Tankovich, Danhang Tang, Cem Keskin, David Kim, Philip Davidson, Adarsh Kowdle, and Shahram Izadi. Articulated distance fields for ultra-fast tracking of hands interacting. *ACM Transactions on Graphics (TOG)*, 36(6):244, 2017.
- [140] Sri Raghu Malireddi, Franziska Mueller, Markus Oberweger, Abhishake Kumar Bojja, Vincent Lepetit, Christian Theobalt, and Andrea Tagliasacchi. Handseg: A dataset for hand segmentation from depth images. *arXiv preprint arXiv:1711.05944*, 2017.
- [141] Markus Oberweger, Gernot Riegler, Paul Wohlhart, and Vincent Lepetit. Efficiently creating 3d training data for fine hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4957–4965, 2016.
- [142] Xia Liu and Kikuo Fujimura. Hand gesture recognition using depth data. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 529–534. IEEE, 2004.
- [143] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1106–1113, 2014.

- [144] Ayan Sinha, Chiho Choi, and Karthik Ramani. Deephand: Robust hand pose estimation by completing a matrix imputed with deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4150–4158, 2016.
- [145] Lihao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8417–8426, 2018.
- [146] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [147] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [148] Yibing Song, Chao Ma, Lijun Gong, Jiawei Zhang, Rynson WH Lau, and Ming-Hsuan Yang. Crest: Convolutional residual learning for visual tracking. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2574–2583. IEEE, 2017.
- [149] Lihao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. Real-time 3d hand pose estimation with 3d convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
- [150] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 424–432. Springer, 2016.
- [151] Shanxin Yuan, Qi Ye, Guillermo Garcia-Hernando, and Tae-Kyun Kim. The 2017 hands in the million challenge on 3d hand pose estimation. *arXiv preprint arXiv:1707.02237*, 2017.

- [152] Shanxin Yuan, Qi Ye, Björn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 2605–2613. IEEE, 2017.
- [153] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. *arXiv preprint arXiv:1704.02463*, 2017.
- [154] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *International Conf. on Computer Vision (ICCV)*, pages 3192–3199, December 2013.
- [155] Zhou Ren, Jingjing Meng, Junsong Yuan, and Zhengyou Zhang. Robust hand gesture recognition with kinect sensor. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 759–760. ACM, 2011.
- [156] Shih-En Wei, Nick C Tang, Yen-Yu Lin, Ming-Fang Weng, and Hong-Yuan Mark Liao. Skeleton-augmented human action understanding by learning with progressively refined data. In *Proceedings of the 1st ACM International Workshop on Human Centered Event Understanding from Multimedia*, pages 7–10. ACM, 2014.
- [157] Yao-Jen Chang, Shu-Fang Chen, and Jun-Da Huang. A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in developmental disabilities*, 32(6):2566–2570, 2011.
- [158] Quentin De Smedt, Hazem Wannous, Jean-Philippe Vandeborre, Joris Guerry, Bertrand Le Saux, and David Filliat. Shrec’17 track: 3d hand gesture recognition using a depth and skeletal dataset. In *10th Eurographics Workshop on 3D Object Retrieval*, 2017.
- [159] Guillaume Devineau, Wang Xi, Fabien Moutarde, and Jie Yang. Convolutional neural networks for multivariate time series classification using both inter-and intra-channel parallel convolutions. In *Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP’2018)*, 2018.

- [160] Jingxuan Hou, Guijin Wang, Xinghao Chen, Jing-Hao Xue, Rui Zhu, and Huazhong Yang. Spatial-temporal attention res-tcn for skeleton-based dynamic hand gesture recognition. *gesture*, 30(5):3, 2018.
- [161] Thomas B Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer vision and image understanding*, 81(3):231–268, 2001.
- [162] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.
- [163] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1302–1310. IEEE, 2017.
- [164] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 466–481, 2018.
- [165] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. volume 36, 2017.
- [166] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7356–7365, 2018.
- [167] Fei Wang, Stanislav Panev, Ziyi Dai, Jinsong Han, and Dong Huang. Can wifi estimate person pose? *arXiv preprint arXiv:1904.00277*, 2019.
- [168] Cheng Chen, Yueting Zhuang, Feiping Nie, Yi Yang, Fei Wu, and Jun Xiao. Learning a 3d human pose distance metric from geometric pose descriptor. *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1676–1689, 2011.

- [169] Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeborre. Skeleton-based dynamic hand gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–9, 2016.
- [170] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017.
- [171] Fabio Marco Caputo, Pietro Prebianca, Alessandro Carcangiu, Lucio D Spano, and Andrea Giachetti. A 3 cent recognizer: Simple and effective retrieval and classification of mid-air gestures from single 3d traces. *Smart Tools and Apps for Graphics. Eurographics Association*, 2017.
- [172] Songyang Zhang, Yang Yang, Jun Xiao, Xiaoming Liu, Yi Yang, Di Xie, and Yueting Zhuang. Fusing geometric features for skeleton-based action recognition using multilayer lstm networks. *IEEE Transactions on Multimedia*, 20(9):2330–2343, 2018.
- [173] Xinghao Chen, Guijin Wang, Hengkai Guo, Cairong Zhang, Hang Wang, and Li Zhang. Mfa-net: Motion feature augmented network for dynamic hand gesture recognition from skeletal data. *Sensors*, 19(2):239, 2019.
- [174] Chuankun Li, Yonghong Hou, Pichao Wang, and Wanqing Li. Joint distance maps based action recognition with convolutional neural network. *IEEE Signal Processing Letters*, 24(5):624–628, 2017.
- [175] Inwoong Lee, Doyoung Kim, Seoungyoon Kang, and Sanghoon Lee. Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1012–1020, 2017.
- [176] Yansong Tang, Yi Tian, Jiwen Lu, Peiyang Li, and Jie Zhou. Deep progressive reinforcement learning for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5323–5332, 2018.

- [177] Zhengyuan Yang, Yuncheng Li, Jianchao Yang, and Jiebo Luo. Action recognition with spatio-temporal visual attention on skeleton image sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.
- [178] Dennis Ludl, Thomas Gulde, and Crist’obal Curio. Simple yet efficient real-time pose-based action recognition. In *ITSC*, 2019.
- [179] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015.
- [180] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision*, pages 816–833. Springer, 2016.
- [181] Hongsong Wang and Liang Wang. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In *e Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [182] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI*, volume 1, pages 4263–4270, 2017.
- [183] Chuankun Li, Pichao Wang, Shuang Wang, Yonghong Hou, and Wanqing Li. Skeleton-based action recognition using lstm and cnn. In *Multimedia & Expo Workshops (ICMEW), 2017 IEEE International Conference on*, pages 585–590. IEEE, 2017.
- [184] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1801.07455*, 2018.
- [185] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *arXiv preprint arXiv:1812.03982*, 2018.

- [186] François Chollet et al. Keras, 2015.
- [187] Juan C Nunez, Raul Cabido, Juan J Pantrigo, Antonio S Montemayor, and Jose F Velez. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognition*, 76:80–94, 2018.
- [188] Mengyuan Liu and Junsong Yuan. Recognizing human actions as the evolution of pose estimation maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1159–1168, 2018.
- [189] Aljoša Osep, Wolfgang Mehner, Markus Mathias, and Bastian Leibe. Combined image-and world-space tracking in traffic scenes. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1988–1995. IEEE, 2017.
- [190] Marina Kollmitz, Andreas Eitel, Andres Vasquez, and Wolfram Burgard. Deep 3d perception of people and their mobility aids. *Robotics and Autonomous Systems*, 114:29–40, 2019.
- [191] Hyunggi Cho, Young-Woo Seo, BVK Vijaya Kumar, and Ragnathan Raj Rajkumar. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1836–1843. IEEE, 2014.
- [192] Samuel Scheidegger, Joachim Benjaminsson, Emil Rosenberg, Amrit Krishnan, and Karl Granström. Mono-camera 3d multi-object tracking using deep learning detections and pmbm filtering. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 433–440. IEEE, 2018.
- [193] Sarthak Sharma, Junaid Ahmed Ansari, J Krishna Murthy, and K Madhava Krishna. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3508–3515. IEEE, 2018.
- [194] Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi. Monoloco: Monocular 3d pedestrian localization and uncertainty estimation. In *Proceedings of*

- the IEEE International Conference on Computer Vision*, pages 6861–6871, 2019.
- [195] Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan. Poi: Multiple object tracking with high performance detection and appearance feature. In *European Conference on Computer Vision*, pages 36–42. Springer, 2016.
- [196] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and real-time tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017.
- [197] Ahmad Delforouzi, David Holighaus, and Marcin Grzegorzec. Deep learning for object tracking in 360 degree videos. In *International Conference on Computer Recognition Systems*, pages 6861–6871. Springer, 2019.
- [198] Ahmad Delforouzi, Seyed Amir Hossein Tabatabaei, Kimiaki Shirahama, and Marcin Grzegorzec. A polar model for fast object tracking in 360-degree camera images. *Multimedia Tools and Applications*, 78(7):9275–9297, 2019.
- [199] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.
- [200] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3d pose estimation and tracking by detection. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 623–630. IEEE, 2010.
- [201] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11977–11986, 2019.
- [202] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

- [203] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 0–8, 2019.
- [204] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.
- [205] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2156, 2016.
- [206] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3917–3925, 2018.
- [207] Clément Godard, Oisín Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.
- [208] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the un-trackable: Learning to track multiple cues with long-term dependencies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 300–311, 2017.
- [209] Nicolai Wojke and Dietrich Paulus. Global data association for the probability hypothesis density filter using network flows. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 567–572. IEEE, 2016.
- [210] Tobias Klinger, Frank Rottensteiner, and Christian Heipke. Probabilistic multi-person localisation and tracking in image sequences. *ISPRS Journal of Photogrammetry and Remote Sensing*, 127:73–88, 2017.

- [211] Zheng Tang and Jenq-Neng Hwang. Moana: An online learned adaptive appearance model for robust multiple object tracking in 3d. *IEEE Access*, 7:31934–31945, 2019.
- [212] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.
- [213] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.
- [214] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018.
- [215] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
- [216] Reid Porter, Andrew M Fraser, and Don Hush. Wide-area motion imagery. *IEEE Signal Processing Magazine*, 27(5):56–65, 2010.
- [217] Patrick C Hytla, Kevin S Jackovitz, Eric J Balster, Juan R Vasquez, and Michael L Talbert. Detection and tracking performance with compressed wide area motion imagery. In *Aerospace and Electronics Conference (NAECON), 2012 IEEE National*, pages 163–170. IEEE, 2012.
- [218] Adam Van Etten. You only look twice: Rapid multi-scale object detection in satellite imagery. *arXiv preprint arXiv:1805.09512*, 2018.
- [219] Peiyun Hu and Deva Ramanan. Finding tiny faces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 951–959, 2017.
- [220] Yancheng Bai, Yongqiang Zhang, Mingli Ding, and Bernard Ghanem. Finding tiny faces in the wild with generative adversarial network. 2018.

- [221] Rodney LaLonde, Dong Zhang, and Mubarak Shah. Clusternet: Detecting small objects in large scenes by exploiting spatio-temporal information. In *Computer Vision and Pattern Recognition*, 2018.
- [222] Jiawei He, Zhiwei Deng, Mostafa S Ibrahim, and Greg Mori. Generic tubelet proposals for action localization. In *IEEE Winter Conference on Applications of Computer Vision*, pages 343–351. IEEE, 2018.
- [223] Zhihao Li, Wenmin Wang, Nannan Li, and Jinzhuo Wang. Tube convnets: Better exploiting motion for action recognition. In *IEEE International Conference on Image Processing*, pages 3056–3060. IEEE, 2016.
- [224] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016.
- [225] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *European conference on computer vision*, pages 445–461. Springer, 2016.
- [226] Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, and Qinghua Hu. Vision meets drones: A challenge. *arXiv preprint arXiv:1804.07437*, 2018.
- [227] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision*, pages 734–750, 2018.
- [228] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. Mars: A video benchmark for large-scale person re-identification. In *European Conference on Computer Vision*, pages 868–884. Springer, 2016.
- [229] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *2018 IEEE Winter Conference on Applications of Computer Vision*, pages 748–756. IEEE, 2018.

- [230] E. Khvedchenya V. I. Iglovikov A. Buslaev, A. Parinov and A. A. Kalinin.
Albumentations: fast and flexible image augmentations. *ArXiv e-prints*,
2018.

Publication List

Journal Articles:

- [1] Fan Yang, Wang Zheng, Yang Wu, Sakriani Sakti and Satoshi Nakamura. “Re-identification Dominated Data Association for Visual Multiple Object Tracking.” IEEE Transactions on Multimedia (under review). Related to Chapter 4 & 5.
- [2] Yang Wu, Dingheng Wang, Xiaotong Lu, Fan Yang, Guoqi Li, Weisheng Dong, Jianbo Shi. “Efficient Visual Recognition with Deep Neural Networks: A Survey on Recent Advances and Trends.” IEEE/CAA JOURNAL OF AUTOMATICA SINICA (under review). Related to Chapter 6.
- [3] Ziqiang Zheng, Hongzhi Liu, Fan Yang, and Xingyu Zheng, Zhibin Yu, and Shaoda Zheng. “Representation-guided Generative Adversarial Network for Unpaired Photo-to-caricature Translation.” Computers Electrical Engineering, 2021. Related to Chapter 4.
- [4] Fan Yang, Chang Xin, Yang Wu, Sakriani Sakti and Satoshi Nakamura. “ReMOT: A Model-agnostic Refinement for Multiple Object Tracking.” Image and Vision Computing Journal, 2021. Related to Chapter 4 & 5.
- [5] Fan Yang, Yang Wu, Zheng Wang, Xiang Li, Sakriani Sakti and Satoshi Nakamura. “Instance-level Heterogeneous Domain Adaptation for Limited-labeled Sketch-to-Photo Retrieval.” IEEE Transactions on Multimedia, 2020. Related to Chapter 4.
- [6] Fan Yang, Yang Wu, Sakriani Sakti and Satoshi Nakamura. “A Framework for Knowing Who is Doing What in Aerial Surveillance Videos.” IEEE Access, 2019. Related to Chapter 6 and Appendix.
- [7] Fan Yang, Yang Wu. “A Soft Proposal Segmentation Network (SPS-Net) for Hand Segmentation on Depth Videos.” IEEE Access, 2019. Related to Chapter 6 and Appendix.

Conference Proceedings:

- [1] Fan Yang, Xin Chang, Chenyu Dang, Ziqiang Zheng, Sakriani Sakti, Satoshi Nakamura, Yang Wu. “ReMOTS: Refining Multi-Object Tracking and Segmentation.” BMTT Workshop of CVPR 2020. Related to Chapter 4 & 5.
- [2] Fan Yang, Feiran Li, Yang Wu, Sakriani Sakti and Satoshi Nakamura. “Using Panoramic Videos for Multi-person Localization and Tracking in a 3D Panoramic Coordinate.” ICASSP 2020 (oral). Related to Chapter 6 and Appendix.
- [3] Fan Yang, Yang Wu, Sakriani Sakti and Satoshi Nakamura. “Make Skeleton-based Action Recognition Model Smaller, Faster and Better.” ACM MM ASIA, 2019 (oral). Related to Chapter 6 and Appendix.
- [4] Shanxin Yuan, Guillermo Garcia-Hernando, Björn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Lihao Ge, Junsong Yuan, Xinghao Chen, Guijin Wang, Fan Yang, Kai Akiyama, Yang Wu, Qingfu Wan, Meysam Madadi, Sergio Escalera, Shile Li, Dongheui Lee, Iason Oikonomidis, Antonis Argyros, Tae-Kyun Kim. “Depth-Based 3D Hand Pose Estimation: From Current Achievements to Future Goals.” CVPR 2018. Related to Chapter 6 and Appendix.

Appendix

A.

A.1 Appendix Overview

Since Action Recognition (AR) covers a broad range of sub topics, we introduce our explorations on some of these topics, which could be used to extend Actor-identified Spatiotemporal Action Detection (ASAD) for the future work.

Specifically, we may integrate 3D MOT, fine-grained hand action detection, skeleton-based action classification, and drone-recorded videos in our ASAD framework (Figure A.1).

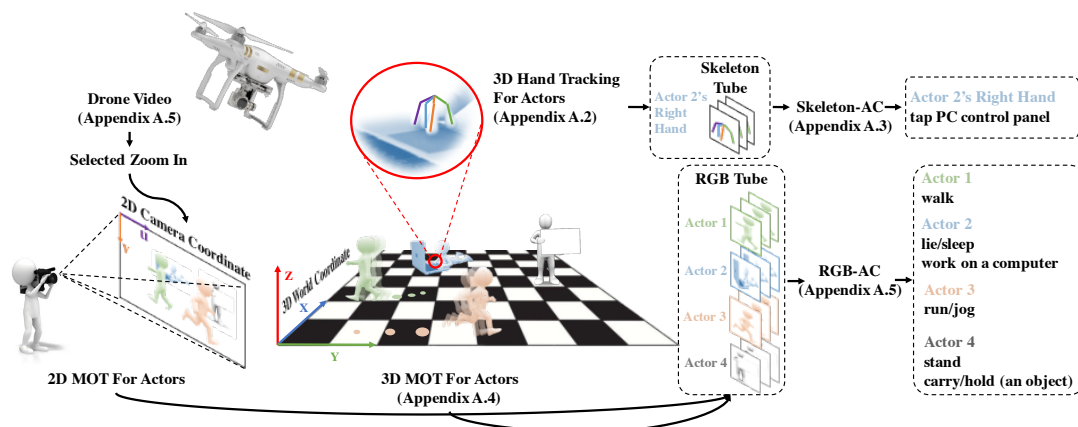


Figure A.1: Integrating 3D MOT, fine-grained hand action detection, skeleton-based action classification, and drone-recorded videos in our ASAD framework.

A.2 Hand Pose/Skeleton Tracking

A.2.1 Overview

Generally, if we only focus on the entire body of an actor for Actor-identified Spatiotemporal Action Detection (ASAD), it could be challenging to recognize fine-grained level actions, since most of the fine-grained level actions may be performed by hands. Hands are essential for human beings to interact with the surrounding environment and play an important role in video action recognition. Hand related applications, such as Virtual/Augmented Reality, are growing rapidly. To approach these applications, 3D hand pose plays an essential role to make the interaction between human hands and devices available. In order to obtain 3D hand poses, plenty of models have been developed. However, most of the existing works only focus on the pose estimation step, and simply suppose the segmented hand is given, or, can be directly acquired by a depth threshold [132, 133, 134, 135, 136]. In this work, we focus on a more realistic situation, where complex background exist (see Fig.A.2) and the aforementioned methods may not be suitable.

To segment hand in a complex background, machine learning approaches are commonly applied on a single depth image [137, 138, 139, 140]. In real applications, we obtain depth videos more than a single depth image, and the temporal information could be employed to improve the hand segmentation performance.

Therefore, we propose a Soft Proposal Segmentation Network (SPS-net), which utilizes the temporal information when performing hand segmentation on depth videos. More technically, SPS-Net generates a soft proposal (detection proposal) in the current frame, meanwhile, another soft proposal (tracking proposal) is generated by a Kalman filter from the previous frame. The final hand segmentation is guided by the merging result of these two soft proposals.

We run segmentation experiments on NYU Hand Dataset [137] and CVAR Dataset [141] to demonstrate the superiority of SPS-Net on segmentation accuracy and generalization ability. Furthermore, by using SPS-Net for segmentation and a simple 3D hand pose estimator, we obtain the new state-of-the-art on the Hand2017 Challenge - 3D Hand Pose Tracking Task¹.

¹<https://competitions.codalab.org/competitions/17356results>

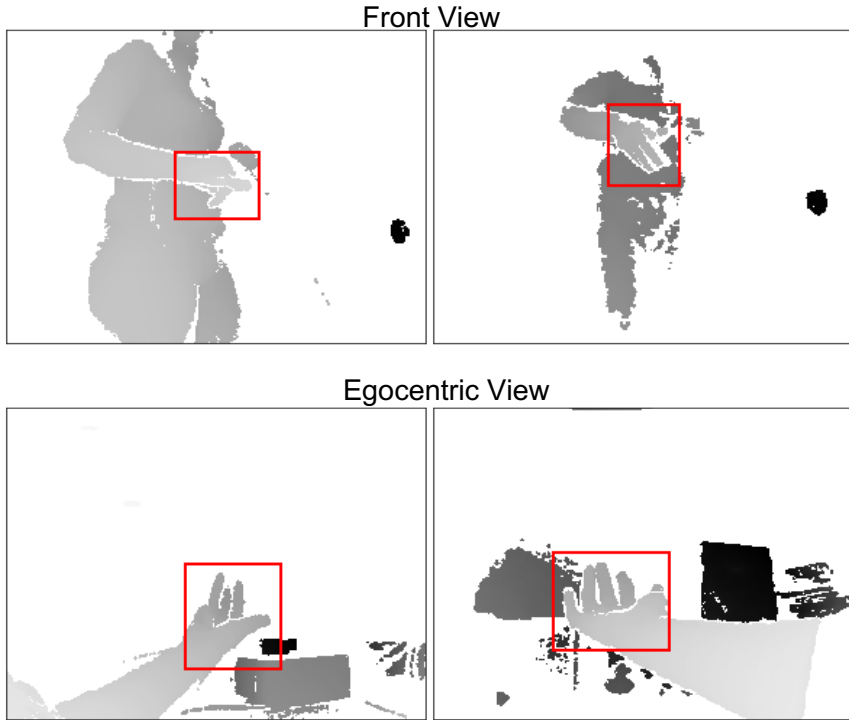


Figure A.2: **Examples of hand location in depth images, where the target hand is inside the red box.** In the egocentric view case, the hand is not the nearest object to depth cameras, so that it cannot be separated from the background through depth threshold.

A.2.2 Background

Most of existing works segment the hand first before performing pose estimation [133, 142, 143, 137, 138, 144, 139, 132, 140, 135, 145, 134, 136]. Such a common choice was mainly driven by three practical considerations. First, it is easier to be extended to multi-hand pose estimation case [139, 140]. Second, when considering the input resolution and model capacity together, only using the hand region as the input is more economic [135, 145]. Third, when depth data is available, once the hand region is correctly obtained, one can normalize the hand size by its depth to eliminate the scale variation problem (see Fig.A.3).

Depth images take a great advantage in the hand segmentation. Compared to the RGB image, the depth image is robust to texture and light intensity

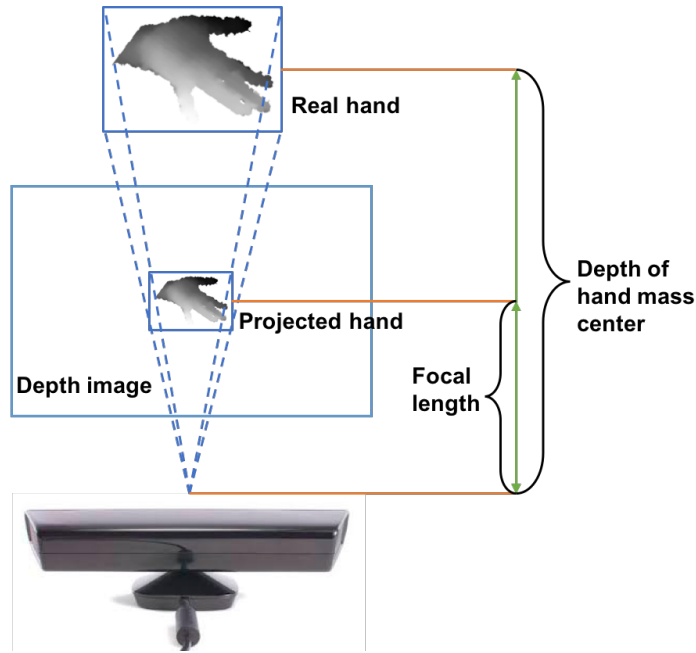


Figure A.3: Normalizing hand scale after segmentation.

variations. Moreover, obtaining the ground truth of the hand region is easier for depth image. For instance, thin gloves with special color could be used to generate the hand masks efficiently without polluting the data itself [139, 140] (see Fig.A.4). Furthermore, since we aim to obtain accurate 3D hand pose by depth images, without further introducing RGB images can reduce the cost of storage and computation.

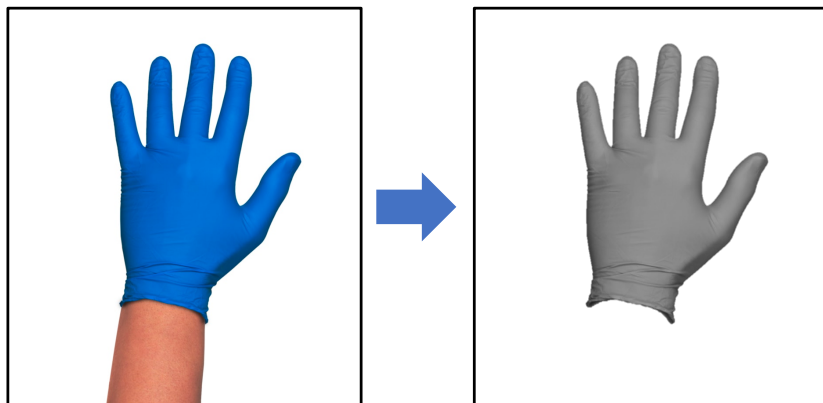


Figure A.4: Obtaining hand segmentation ground truth by using gloves.

However, for depth images, hand segmentation is commonly treated as a trivial problem, a typical assumption is that the hand is always the closest object to the camera, so that the hand can be easily segmented by certain depth thresholds [142, 132]. Apparently, this solution only works for the very restricted scenario: when a single hand in frontal view. A more general and sophisticated method uses random forests together with hand-crafted features [137, 138]. As commonly realized, effective and robust hand-crafted features are hard to get, and thus they usually have relatively limited performance. Hence, recently more efforts have been paid to tailoring powerful deep learning models, such as U-net [146] or Fully Convolution Network (FCN) [147], for hand segmentation [139, 140]. While existing works [139, 140] treat each depth image isolated, our SPS-Net employs the temporal information to further improve the hand segmentation performance on depth videos.

A.2.3 Methodology

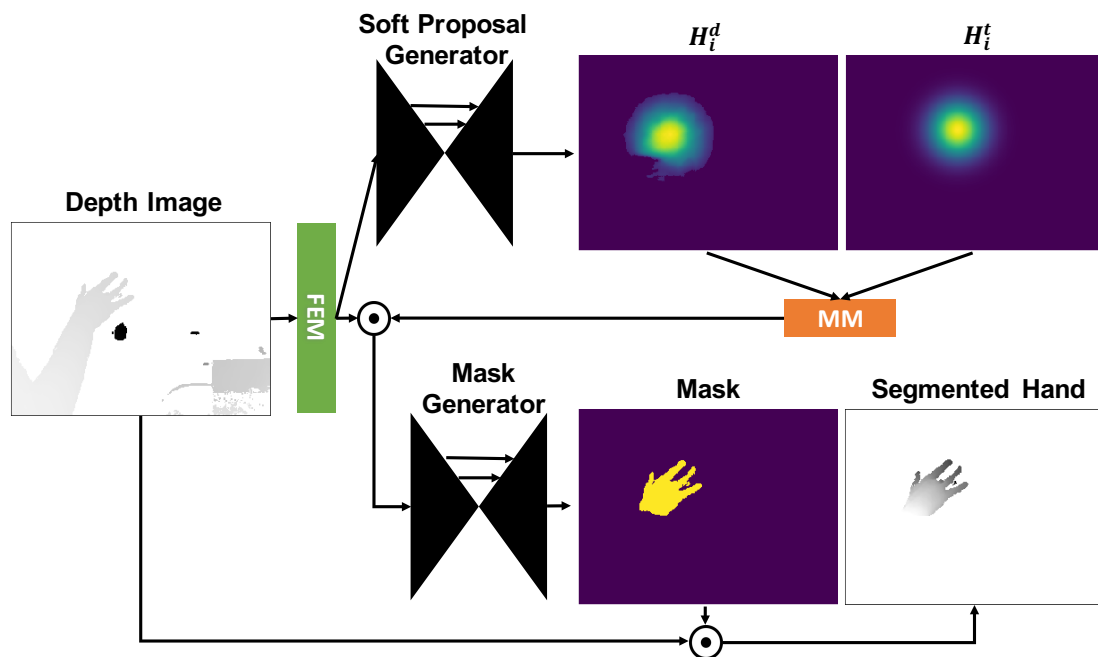
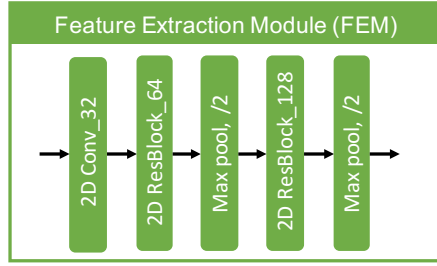
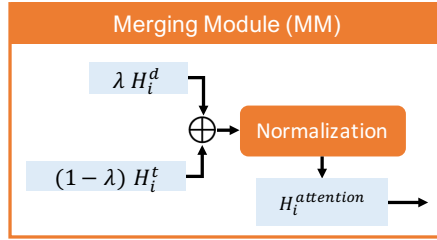


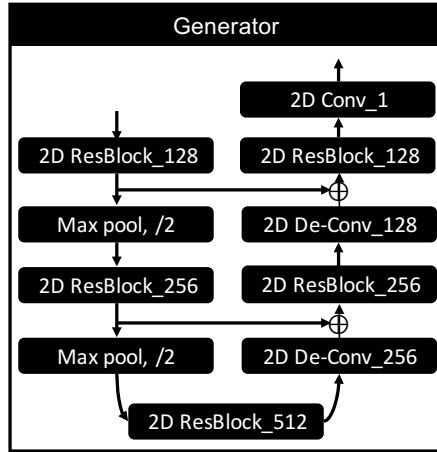
Figure A.5: **The architecture of SPS-Net.** The \odot represents element-wise multiplication. H_i^d and H_i^t are soft proposals generated at frame i .



(1) Feature Extraction Module (FEM)



(2) Merging Module (MM)



(3) Generator

Figure A.6: **The details of each components in SPS-Net.** “2D ResBlock_64” denotes an 2D residual residual convolutional block with kernel size 64. “Max pool, /2” means maxpooling operation with stride = 2. The \oplus denotes element-wise summation.

Soft Proposal Segmentation Network (SPS-net) mainly includes four network components: Feature Extraction Module (FEM), Merging Module (MM), Soft Proposal Generator, and Mask Generator (see Fig.A.5). In addition, an auxiliary

algorithm cooperates with SPS-Net to generate one of the soft proposals (see Algorithm 1).

More specifically, FEM starts with a CNN layer, followed by two CNN Residual Blocks [113] and Max pooling (see Fig.A.6.(1)). MM performs merging and normalization operation to two soft proposals (see Fig.A.6.(2)), which are generated by the Soft Proposal Generator and the Kalman filter (see Algorithm 1), respectively. Soft Proposal Generator and Mask Generator have the same architecture (see Fig.A.6.(3)), which is modified from U-net [146]. Except for the last CNN layer of Soft Proposal Generator and Mask Generator that use Sigmoid Activation, other CNN layers in SPS-Net all use Relu Activation.

Soft Proposal Generator is applied to generate a hand proposal heat map H_i^d at frame i (i.e., the detection proposal). To some extent, H_i^d is a spatial self-attention that can guide the hand segmentation. The ground-truth of H_i^d is H_i^{GT} , which is generated by

$$H_i^{GT}(r, c) = \exp\left(-\frac{(r - p_i^{GT}(r))^2 + (c - p_i^{GT}(c))^2}{2\sigma_i^2}\right), \quad (6.1)$$

where r and c are the row index and column index of the heat map, respectively; p_i^{GT} is the ground-truth hand center with 2D coordinates $[p_i^{GT}(r), p_i^{GT}(c)]$; σ_i is a Gaussian Covariance for frame i , which can be adjusted by the projection scale s_i .

Referenced to Fig.A.3, the projection scale s_i can be calculated as follows:

$$s_i = \frac{f}{d_i^{hand}}, \quad (6.2)$$

where d_i^{hand} is the depth of hand mass center in frame i , and f is the focal length of depth camera. The scale of the hand s_i is inversely proportional to d_i^{hand} , i.e., the distance between the hand center and the camera.

Assuming the proper Gaussian covariance at the focal point is σ_f , which will be determined empirically, the corresponding σ_i at frame i is calculated by

$$\sigma_i = \sigma_f \cdot s_i. \quad (6.3)$$

During the training process, we try to minimize the divergence between the H_i^d and H_i^{GT} by the cross-entropy loss.

Due to the complex background, nonetheless, improper H_i^d could be estimated so that the hand segmentation could be misguided. Inspired by [148], we introduce a soft tracking proposal H_i^t to build a more robust soft proposal. H_i^t is generated by

$$H_i^t(r, c) = \exp\left(-\frac{(r - p_i^t(r))^2 + (c - p_i^t(c))^2}{2\sigma_i^2}\right). \quad (6.4)$$

The equation of generating H_i^t is similar to H_i^{GT} . The difference is that p_i^t is the tracked hand center with 2D coordinates $[p_i^t(r), p_i^t(c)]$, which are obtained in Algorithm 1.

Algorithm 1: Auxiliary algorithm

- 1 **Input:** Depth image I_i , H_i^t , p_{i-1}^t [1] H_i^d , $hand_i^{mask} = \text{SPS-Net}(I_i, H_i^t)$
Obtain the peak point of H_i^d as p_i^d
 - 2 **if** $|p_i^d - p_{i-1}^d| < t_1$ **and** $\max(H_i^d) > t_2$ **then**
 - 3 Update Kalman filter with p_i^d ; $p_i^t = p_i^d$; $j = 0$ **else**
 - 4 $j \geq t_3$ **and** $\max(H_i^d) > t_2$ $p_i^t = p_i^d$; $j = 0$ **else**
 - 5 Predict p_i^t by Kalman filter; $j++$ **endif endif**
 - 6 Segment hand as $hand_i = I_i \odot hand_i^{mask}$ Update d_{i+1}^{hand} by calculating the mean depth of $hand_i$ Update σ_{i+1} by d_{i+1}^{hand} using Eq.(6.10) and Eq.(6.11)
Obtain H_{i+1}^t by p_i^t and σ_{i+1} , using Eq.(6.23) **Output:** $hand_i$, $hand_i^{mask}$, H_{i+1}^t , p_i^t and d_{i+1}^{hand} .
-

In Algorithm 1, it is natural to suppose the hand center should shift less than a threshold t_1 between two adjacent frames. In addition, the intensity of H_i^t presents the confident level of where the hand center is located. When the maximum intensity of H_i^t is smaller than a threshold t_2 , most likely, we obtain an improper detection proposal. Hence, we will predict the hand center by the Kalman filter instead. However, the Kalman filter cannot keep a long time tracking without correct updating. For this reason, we re-initialize the tracking proposal after t_3 frames. Here, t_1 , t_2 , t_3 are determined empirically.

After obtaining H_i^d and H_i^t , MM is designed to merge them. The merging

function is

$$H_i^{merge} = \lambda H_i^d \oplus (1 - \lambda) H_i^t, \quad (6.5)$$

where \oplus means element-wise sum; λ is a coefficient to weight the importance of H_i^d and H_i^t . We set $\lambda = 0.5$ as default.

We further normalize H_i^{merge} by

$$H_i^{merge} = \frac{H_i^{merge} - \min(H_i^{merge})}{\max(H_i^{merge}) - \min(H_i^{merge})}. \quad (6.6)$$

The input of Mask Generator is the result of element-wise multiplication between H_i^{merge} and output of FEM, while the output is a binary hand mask, where the hand part is represented by value 1 and the background is 0.

For the segmentation process, all depth images are normalized to be 240×320 as inputs. After obtaining the hand mask, we rescale it back to the original size of the depth image. By applying another element-wise multiplication between the hand mask and raw depth image, the segmented hand is obtained (see Fig.A.5).

A.2.4 Experiments

A.2.4.1 Other models used in experiment

Although we focus on hand segmentation in this thesis, it should be clarified that the hand segmentation serves for the 3D hand pose estimation. Thus, after confirming our SPS-Net can achieve high performance in the segmentation task, we further check the weather a hand the estimator can use the segmentation result to generate accurate 3D hand poses. We, therefore, use the volumetric representations from [149], and apply a shallow 3D U-Net [150] to build a hand pose estimator, which is modified from [135].

To explore the impacts of soft proposals in SPS-Net, we compare the segmentation performance by removing soft proposals in the ablation study. Referred to Fig.A.5, we construct three ablation networks as SPS-Net (without H^d), SPS-Net (without H^t), and SPS-Net (without $H^d \& H^t$). The Soft Proposal Generator and MM module may or may not be used based on the needs.

To compare the hand segmentation performance, Randomized Decision Forest (RDF) [137], U-Net [146] and Mask-RCNN [119] are used in our experiment. Besides, we suppose the normal hand length is 250 mm and use it as a depth threshold to perform hand segmentation.

RDF and U-Net directly segment the hand, while Mask-RCNN predicts a hand region box first, and then segment the hand within the box. To some extent, SPS-Net uses soft proposals while Mask-RCNN applies hard proposals (i.e., the box). The advantage of using soft proposal is that it can be seamlessly fused across frames, which helps to utilize the temporal information. In addition, the hand segmentation is more changeable than the body segmentation: firstly, there is no clear boundary to distinguish the hand so that it is hard to generate hard proposals; secondly, using hard proposals may exclude fingers before segmentation, this will significantly affect the hand pose estimation performance. Soft proposals, nonetheless, only give a likelihood of hand location, could alleviate the aforementioned issues.

A.2.4.2 Implementation Details

On experimental datasets, we create the hand mask referring to the given 3D hand poses. In the training, we simulate the hand movement by randomly shifting p_i^t away from the ground-truth hand center. Whereas, in the testing, we follow the procedure in Algorithm 1 to generate p_i^t .

Since our network is simple, it can be trained from scratch. During the training, all of the training samples are included in one epoch. The Adam [121] optimizer with a learning rate 1^{-3} is applied for the first 5 epochs and then the learning rate is changed to be 1^{-4} for another 5 epochs. The batch size is set up to be 16. We jointly perform data augmentation on depth images and their correlated ground truth, which includes shifting, rotation, and scaling. We perform the training and testing on a single NVIDIA Titan X GPU.

A.2.4.3 Datasets

	NYU [137]	CVAR [141]	Hand2017 Challenge [151]
Observation Views	Left, right, front	Egocentric	Front, egocentric
Number of samples	81,009 \times 3(3 <i>views</i>)	4,332	1,251,000
Evaluation task	Hand segmentation	Hand segmentation	3D pose tracking
Sequential training data	Yes	Yes	No
Sequential testing data	Yes	Yes	Yes

Table A.1: Properties of experimental datasets.

We perform our experiment on three datasets: NYU Hand Dataset [137], CVAR Dataset [141], and Hand2017 Challenge Dataset [151]. Their properties

are illustrated in Table.A.1.

The NYU Hand Dataset maintains the sequential ordering, realistic background, as well as the given 3D hand pose for generating the ground-truth hand area. It covers the left view, the right view, and the front view. The CVAR Dataset [141] offers sequential depth videos and corresponding 3D hand poses from an egocentric view. The CVAR Dataset includes 6 videos, which entirely contains 4,332 frames of depth videos and corresponding 3D hand pose. It is relatively small and only contains egocentric view cases, which is missing NYU Hand Dataset. We perform leave-one-out cross-validation in it. Samples from BigHand 2.2M Dataset [152] and First-Person Hand Action Dataset [153] are combined to make the Hand2017 Challenge Dataset [151], it, therefore, covers the front view and egocentric view scenarios. In the testing set of tracking tasks, there are 99 videos. In each video, depth images are organized by sequence. However, in the training set, samples are disordered, which increases the difficulty of employing temporal information during training. Nonetheless, our SPS-Net still can be trained on such non-sequential data but utilize temporal information during the testing.

A.2.4.4 Evaluation Metrics

The mean of Intersection over Union (mIoU) is commonly used in image segmentation evaluation. At frame i , supposing X_i is the ground-truth hand mask and Y_i is the predicted hand mask by SPS-Net, mIoU can be calculated by

$$mIoU = \frac{1}{N} \sum_{i=1}^N \frac{|X_i \cap Y_i|}{|X_i| + |Y_i| - |X_i \cap Y_i|}; \quad (6.7)$$

where N is the number of frames.

In 3D hand pose estimation, the evaluation metric is the average Euclidean-distance error (ADE) between estimated 3D hand poses and ground truth, and the unit is a millimeter in general.

$$ADE = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M \|x_i^j - y_i^j\|_2; \quad (6.8)$$

where N and M are the number of frames and joints, respectively. For joint j at frame i , x_i^j and y_i^j are the ground-truth and predicted 3D coordinates, respectively.

A.2.4.5 Evaluation Results

Models	mIoU
By Depth Threshold = 250 mm	0.36
RDF	0.82
U-net	0.87
Mask-RCNN	0.82
SPS-Net (without H^d & H^t)	0.86
SPS-Net (without H^d)	0.84
SPS-Net (without H^t)	0.88
SPS-Net	0.94

Table A.2: Hand segmentation results on NYU Hand Dataset.

Models	mIoU
By Depth Threshold = 250 mm	0.43
RDF	0.82
U-net	0.84
Mask-RCNN	0.83
SPS-Net (without H^d & H^t)	0.83
SPS-Net (without H^d)	0.82
SPS-Net (without H^t)	0.85
SPS-Net	0.89

Table A.3: Hand segmentation results on CVAR Dataset.

RANK	TEAMS	ALL (mm)	SEEN (mm)	UNSEEN (mm)
1	Ours (SPS-Net + Hand Pose Estimator)	10.48	8.28	12.26
2	NVIDIA Research and UMontreal	10.51	8.21	12.37
3	THU VCLab	13.65	11.02	15.70
4	Baseline from Organizer [151]	20.63	16.04	24.36

Table A.4: The top-4 results of the Hand2017 Challenge - 3D Hand Pose Tracking Task. ALL denotes ADE of all joints; SEEN denotes ADE over visible joints; UNSEEN denotes ADE over occluded joints.

For NYU Hand Dataset, the quantitative results and qualitative hand segmentation performance are shown in Table.A.2 and Fig.A.1, respectively. The

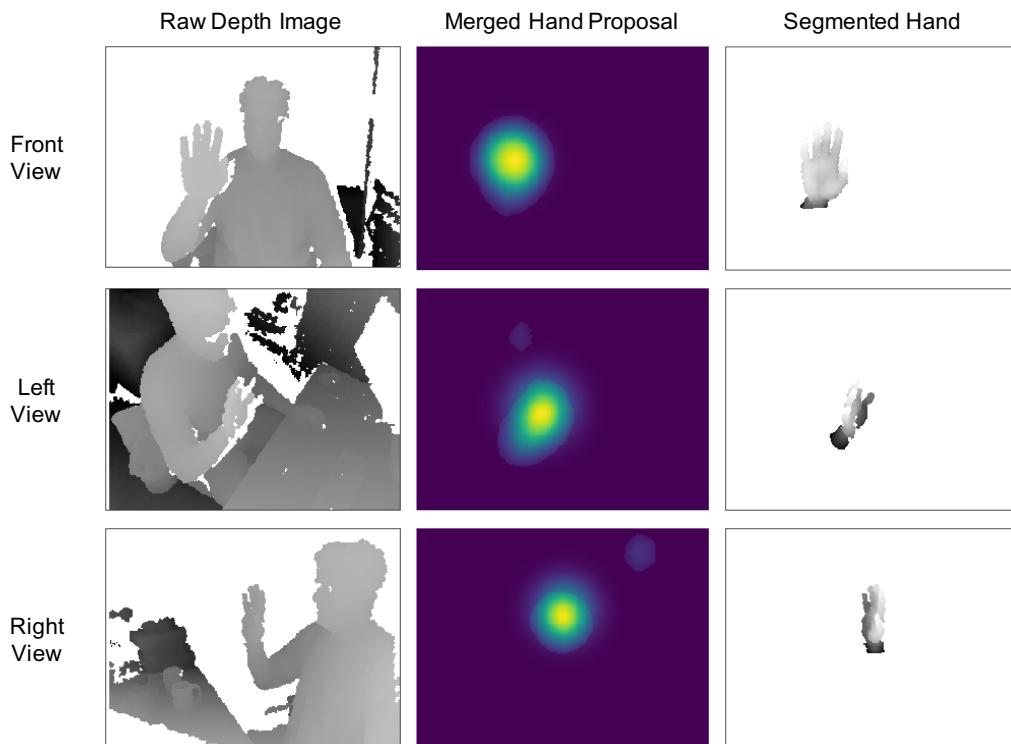


Figure A.1: Qualitative results of hand segmentation on the NYU Hand Dataset.

performance of each model on the CVAR Dataset is given in Table.A.3. We list the current leader board of 3D hand poses tracking task as Table.A.4 shows. Additionally, its qualitative results are shown in Fig.A.2.

A.2.4.6 Result Analysis

Overall, our SPS-Net can achieve superior results to other methods on three experiential datasets. In contrast, simply using depth threshold for hand segmentation, which is commonly applied in existing works, could be failed in NYU Hand Dataset and CVAR Dataset, when side-view and egocentric-view cases existing. In Hand2017 Challenge - 3D Hand Pose Tracking Task, The ADE of all joints that our 3D hand tracking system achieves is as low as 10.48 mm . This indicates that SPS-Net can generate high-quality segmented hands, which intermediately helps the 3D hand pose estimator to generate accurate 3D hand poses.

In ablation studies, we can further inspect that using two soft proposals generates better segmentation performance than only using one of them or none of

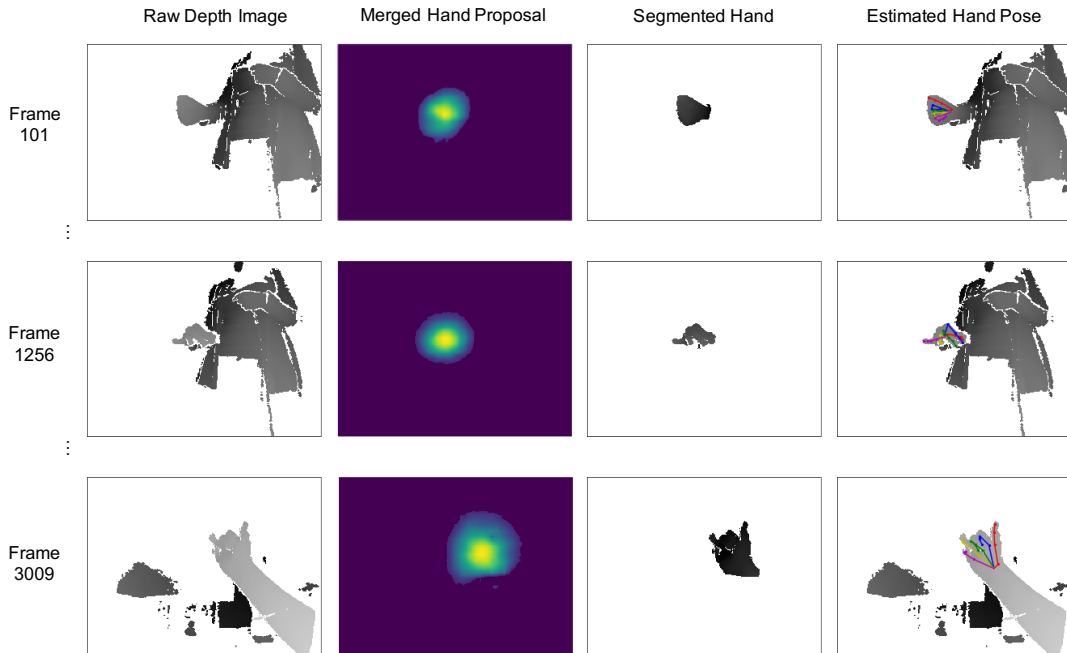


Figure A.2: Qualitative results of 3D hand pose tracking on the Hand2017 Challenge.

them. This suggests that both of the soft proposals contribute to improving the segmentation performance. The Soft Proposal Generator module generates the soft proposal H^d by using the entire spatial context. Compared to Mask-RCNN, which utilizes hard proposals (i.e., boxes), SPS-Net (without H^t) can generate better hand segmentation result by using soft proposals.

However, when ambiguous contexts existing, H^d may not be correctly generated. The segmentation performance could be compromised by solely using H^d . Through introducing H^t , the improper H^d could be corrected by using temporal information. Despite that, H^t are purely based on temporal information, without H^d , its confidence will decrease as the frame increasing. To obtain a robust soft proposal, both of H^t and H^d are needed to work collaboratively. From Fig.A.1 and Fig.A.2, it can be noticed that the center of the merged hand proposal is close to the hand center, while the soft proposal region matches the hand region. Therefore, the hand can be properly segmented under the guidance of the merged

soft proposal.

A.2.5 Discussion

In this thesis, we argue that hand segmentation from depth data is an essential process for accurate 3D hand pose estimation, while it is a non-trivial problem. Although most related works suppose that the hand can be easily segmented by a depth threshold, we take experiments to demonstrate it does not have the generalization in real scenarios, where the complex background exists. Our Soft Proposal Segmentation Network (SPS-Net) is proposed to serves real scenarios. On NYU Hand Dataset and CVAR Dataset, which cover samples from the front view, side view, and egocentric view, our SPS-Net outperforms other related models. In the ablation study, we further confirm that SPS-Net could improve the hand segmentation performance by utilizing the temporal information in depth videos. With the high-quality segmentation results from SPS-Net, we are able to estimate accurate 3D hand poses and achieve the leading result on the Hand2017 Challenge - 3D Hand Pose Tracking Task. Moreover, hand segmentation is a pre-process which can serve for a wide range of post-hoc applications, and integrating SPS-Net into other application framework is available.

A.3 Skeleton-based Action Classification

A.3.1 Overview

After applying MOT to obtain the spatiotemporal boundary for a specific actor, we can utilize the visual information within an actor’s spatiotemporal boundary to estimate the action categories. We will discuss the skeleton-based AC in this part since skeleton-based AC is a more complicated but useful case for action recognition studies.

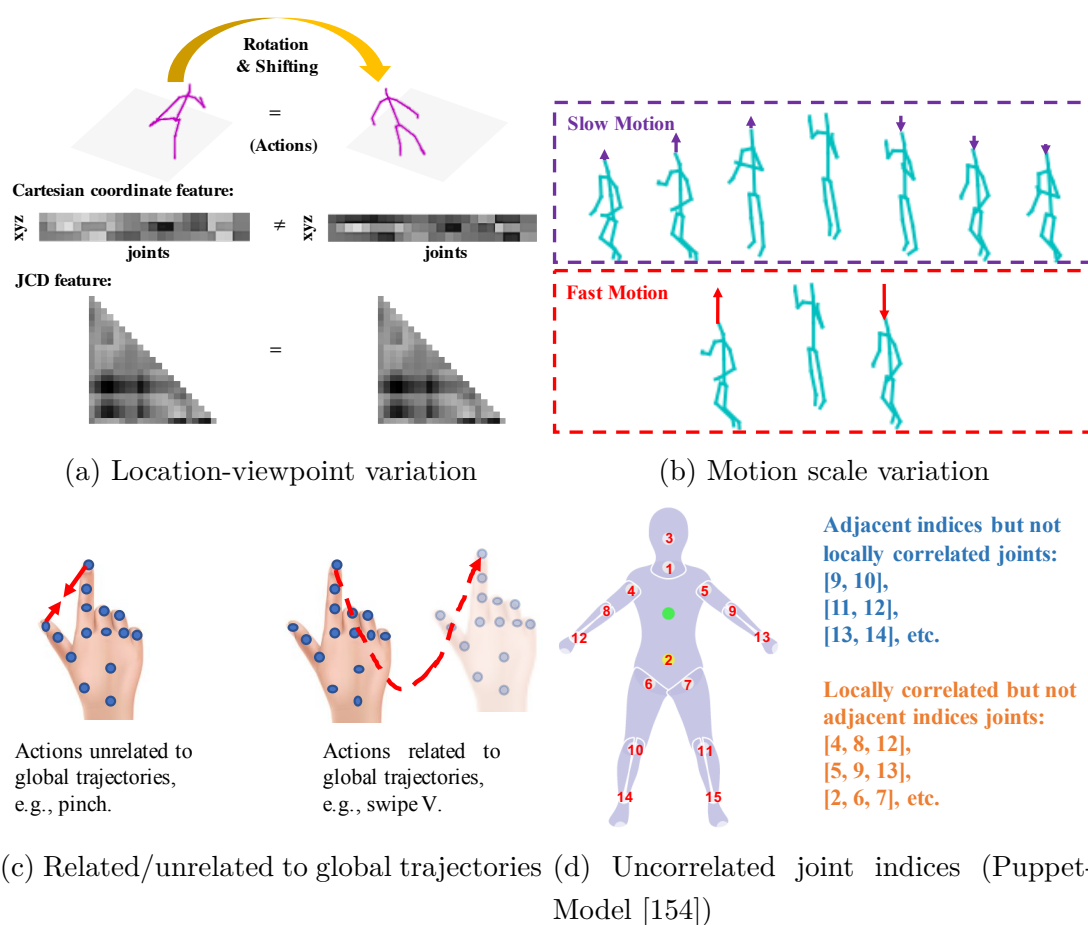


Figure A.3: Concerned skeleton sequence properties.

Skeleton-based action classification has been widely used in real applications, such as human-computer interaction [155], human behavior understanding [156] and medical assistive applications [157]. However, most of the existing methods

may suffer from a large model size and slow execution speed [158, 159, 160, 95, 93].

In real applications, a desirable skeleton-based action classification model should run efficiently by using a few parameters, and, also be adaptable to various application scenarios (*e.g.*, hand/body, 2D/3D skeleton, and actions related/unrelated to global trajectories). To achieve this goal, we investigate skeleton sequence properties to propose a lightweight Double-feature Double-motion Network (DD-Net), which is equipped with a Joint Collection Distances (JCD) feature and a two-scale global motion feature.

More specifically, we conduct research on four types of skeleton sequence properties (see Fig. A.3): (a) location-viewpoint variation, (b) motion scale variation, (c) related/unrelated to global trajectories, (4) uncorrelated joint indices. To address challenges caused by these properties, previous works may prone to propose complicated neural network models, which end up with a large model size.

In contrast, we address these challenges by simplifying both the input feature and the network structure. Our JCD feature contains the location-viewpoint invariant information of skeleton sequences. Compared with other similar features, it can be easily computed and includes fewer elements. Since global motions cannot be incorporated into a location-viewpoint invariant feature, we introduce a two-scale global motion feature to improve the generalization of DD-Net. Besides, its two-scale structure makes it robust to the motion scale variance. Through an embedding process, DD-Net can automatically learn the proper correlation of joints, which is hard to be predefined by joint indices.

Compared to methods relying on complicated model structures, DD-Net provides higher action classification accuracy and demonstrates its generalization on our experiential datasets. With its efficiency both in terms of computational complexity and the number of parameters, DD-Net is sufficient to be applied in real applications.

A.3.2 Background

Nowadays, with the fast advancement of deep learning, skeleton acquisition is not limited to use motion capture systems [161] and depth cameras [162]. The RGB data, for instance, can be used to infer 2D skeletons [163, 164] or 3D skeletons [165, 149] in real-time. Moreover, even WiFi signals can be used to estimate

skeleton data [166, 167]. Those achievements have made skeleton-based action classification available on a huge amount of multimedia resources and therefore have stimulated the model’s development.

In general, in order to achieve a better performance for skeleton-based action classification, previous studies attempt to work on two aspects: introduce new features for skeleton sequences [168, 169, 170, 171, 172, 93, 173], and, propose novel neural network architectures [174, 175, 176, 177, 159, 160, 178].

A good skeleton-sequence representation should contain global motion information and be location-viewpoint invariant. However, it is challenging to satisfy both requirements in one feature. The studies [169, 171, 93, 173] focused on global motions without considering the location-viewpoint variation in their features. Other studies [168, 170, 172], on the contrary, introduced location-viewpoint invariant features without considering global motions. Our work bridges their gaps by seamlessly integrating a location-viewpoint invariant feature and a two-scale global motion feature together.

Although Recurrent Neural Networks (RNNs) are commonly used in skeleton-based action classification [179, 180, 181, 182, 183, 172], we argue that it is relatively slow and difficult for parallel computing, compared with methods [174, 159, 173] that use Convolutional Neural Networks (CNNs). Since we take the model speed as one of our priorities, we utilize 1D CNNs to construct the backbone network of DD-Net.

A.3.3 Methodology

The network architecture of Double-feature Double-motion Network (DD-Net) is shown in Fig. A.4. In the following, we explain our motivation for designing input features and network structures of DD-Net.

A.3.3.1 Modeling Location-viewpoint Invariant Feature by Joint Collection Distances (JCD)

For skeleton-based action classification, two types of input features are commonly used: the geometric feature [168, 172] and the Cartesian coordinate feature [181, 182, 184, 160, 95]. The Cartesian coordinate feature is variant to locations and viewpoints. As Fig. A.3 (a) shows, when skeletons are rotated or shifted, the Cartesian coordinate feature can be significantly changed. The geo-

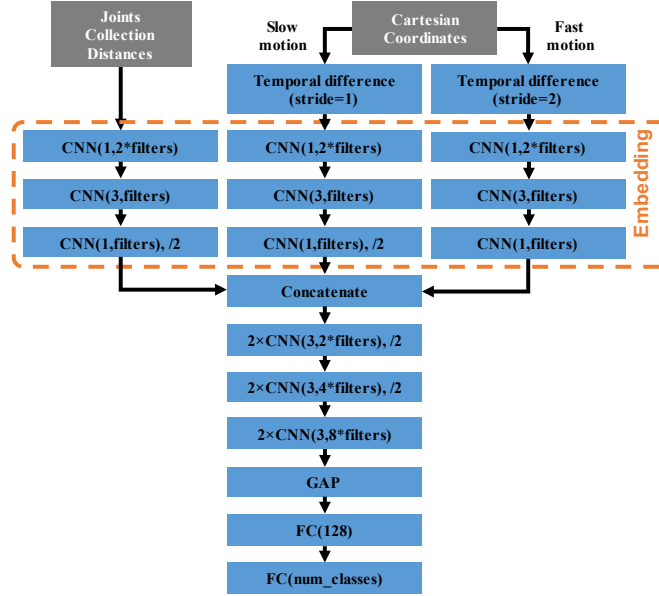


Figure A.4: The network architecture of DD-Net. “ $2 \times \text{CNN}(3, 2 * \text{filters}), /2$ ” denotes two 1D ConvNet layers (kernel size = 3, channels = $2 * \text{filters}$) and a Maxpooling (strides = 2). Other ConvNet layers are defined in the same format. GAP denotes Global Average Pooling. FC denotes Fully Connected Layers (or Dense Layers). We can change the model size by modifying *filters*.

metric feature (*e.g.*, angles/distances), on the other hand, is location-viewpoint invariant, and thereby it has been utilized for skeleton-based action classification for a while. However, existing geometric features may need to be heavily redesigned from one dataset to another [168, 172], or, contain redundant elements [183]. To alleviate these issues, we introduce a Joint Collection Distances (JCD) feature.

We calculate the Euclidean distances between a pair of collective joints to obtain a symmetric matrix. To reduce the redundancy, only the lower triangular matrix without the diagonal part is used as the JCD feature (see Fig. A.5). Hence, the JCD feature is less than half the size of [183].

In more detail, we assume the total frame number is K ($K = 32$ as the default setting) and there are totally N joints for one subject. At frame k , the 3D Cartesian coordinates of joint n is represented as $J_i^k = (x, y, z)$, while the 2D Cartesian coordinates is represented as $J_i^k = (x, y)$. Put all of joints together,

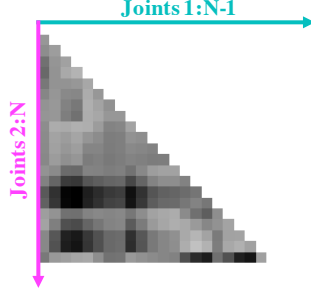


Figure A.5: An example of Joint Collection Distances (JCD) feature at frame k , where the number of joints is N .

we have a joint collection $S^k = \{J_1^k, J_2^k, \dots, J_N^k\}$. The formula for calculating the JCD feature of S^k is:

$$JCD^k = \begin{bmatrix} \left\| \overrightarrow{J_2^k J_1^k} \right\|_2 & & & \\ \vdots & \ddots & & \\ \vdots & \cdots & \ddots & \\ \left\| \overrightarrow{J_N^k J_1^k} \right\|_2 & \cdots & \cdots & \left\| \overrightarrow{J_N^k J_{N-1}^k} \right\|_2 \end{bmatrix}; \quad (6.9)$$

where $\left\| \overrightarrow{J_i^k J_j^k} \right\|_2$ ($i \neq j$) denotes the Euclidean distance between J_i^k and J_j^k .

In our processing, the JCD feature is flattened to be a one-dimensional vector as our model's input. The dimension of flattened JCD is $\binom{N}{2}$.

A.3.3.2 Modeling Global Scale-invariant Motions by a Two-scale Motion Feature

Although the JCD feature is location-viewpoint invariant, the same as other geometric features, it does not contain global motion information. When actions are associated with global trajectories (see Fig. A.3 (c)), solely using the JCD feature is insufficient. Unlike previous works that only utilize either the geometric feature [168, 172] or the Cartesian coordinate feature [174, 175, 176, 177], our DD-Net seamlessly integrates both of them.

We calculate the temporal differences (*i.e.*, the speed) of the Cartesian coordinate feature to obtain global motions, which is location-invariant. For the same action, however, the scale of global motions may not be exactly identical. Some might be faster, and others might be slower (see Fig. A.3 (b)). To learn a robust global motion feature, both fast and slow motions should be considered.

Conferring this intuition to DD-Net, we employ a fast global motion and a slow global motion to form a two-scale global motion feature. This idea is inspired by the two-scale optical flows proposed for RGB-based action classification [185].

Technically, the two-scale motions can be generated by the following equation:

$$\begin{aligned} M_{slow}^k &= S^{k+1} - S^k \text{ for } k \in \{1, 2, 3, \dots, K-1\}; \\ M_{fast}^k &= S^{k+2} - S^k \text{ for } k \in \{1, 3, \dots, K-2\}; \end{aligned} \quad (6.10)$$

where M_{slow}^k and M_{fast}^k denote the slow motion and the fast motion at frame k , respectively. S^{k+1} and S^{k+2} are behind the S^k of one frame and two frames, respectively. Corresponding to $S^{[1, \dots, K]}$, we have $M_{slow}^{[1, \dots, K-1]}$ and $M_{fast}^{[1, \dots, K/2-1]}$ when K is an even number.

To generate an one-dimensional input at each frame, we reshape M_{slow}^k and M_{fast}^k as $M_{slow}^k \in \mathbb{R}^{D_{motion}}$ and $M_{fast}^k \in \mathbb{R}^{D_{motion}}$, respectively, where D_{motion} is the dimension of flattened vector. To match the frame number of the JCD feature, we perform linear interpolation to resize $M_{slow}^{[1, \dots, K-1]}$ and $M_{fast}^{[1, \dots, K/2-1]}$ as $M_{slow}^{[1, \dots, K]}$ and $M_{fast}^{[1, \dots, K/2]}$, respectively. Consequently, two-scale global motion feature is composed of $M_{slow}^{[1, \dots, K]} \in \mathbb{R}^{K \times D_{motion}}$ and $M_{fast}^{[1, \dots, K/2]} \in \mathbb{R}^{(K/2) \times D_{motion}}$. Such a process can be done in our DD-Net, and only the Cartesian coordinate feature is needed as the input.

A.3.3.3 Modeling Joint Correlations by an Embedding

Fig. A.3 (d) shows that the joint indices (*i.e.*, the IDs of the head, left and right hands, *etc.*) are not locally correlated. Moreover, in different actions, the correlation of joints could be dynamically changed. Hence, the difficulty arises when we try to pre-define the correlation of joints by manually ordering their indices.

Since most neural networks inherently assume that inputs are locally correlated, directly processing the locally uncorrelated joint feature is inappropriate. To tackle this problem, our DD-Net embeds the JCD feature and the two-scale motion feature into latent vectors at each frame. The correlation of joints is automatically learned through the embedding. As another benefit, the embedding process also reduces the effect of skeleton noise.

More formally, let embedding representations of JCD^k , M_{slow}^k and M_{fast}^k to

be ε_{JCD}^k , $\varepsilon_{M_{slow}}^k$ and $\varepsilon_{M_{fast}}^k$, respectively, the embedding operation is as follows,

$$\begin{aligned}\varepsilon_{JCD}^k &= Embed_1(JCD^k); \\ \varepsilon_{M_{slow}}^k &= Embed_1(M_{slow}^k); \\ \varepsilon_{M_{fast}}^k &= Embed_2(M_{fast}^k).\end{aligned}\tag{6.11}$$

where the $Embed_1$ is defined as $Conv1D(1, 2 * filters) \rightarrow Conv1D(3, filters) \rightarrow Conv1D(1, filters)$, and the $Embed_2$ is defined as $Conv1D(1, 2 * filters) \rightarrow Conv1D(3, filters) \rightarrow Conv1D(1, filters) \rightarrow Maxpooling(2)$, because JCD^k and M_{slow}^k have twice the temporal length of M_{fast}^k .

DD-Net further concatenates embedding features to a representation ε^k by

$$\begin{aligned}\varepsilon^k &= \varepsilon_{JCD}^k \oplus \varepsilon_{M_{slow}}^k \oplus \varepsilon_{M_{fast}}^k, \\ w.r.t. \ \varepsilon^k &\in \mathbb{R}^{(K/2) \times filters};\end{aligned}\tag{6.12}$$

where \oplus is the concatenation operation.

After the embedding process, subsequent processes are not affected by the joint indices, and therefore DD-Net can use the 1D ConvNet to learn the temporal information as Fig. A.4 shows.

A.3.4 Experiments

A.3.4.1 Experimental Datasets

We select two skeleton-based action classification datasets, as SHREC dataset [158] and JHMDB dataset [154], to evaluate our DD-Net from different perspectives (see Table A.5).

Although other information (*e.g.*, RGB data) is available, only the skeleton information is used in our experiments. 3D skeletons are given by the SHREC dataset, which is derived from RGB-D data and contain more spatial information. In the JHMDB dataset, 2D skeletons are interpreted from RGB videos, which can be applied in more general cases where inferring the depth information may be hard or impossible. Besides, actions in SHREC dataset are strongly correlated to the subject’s global trajectories (*e.g.*, a hand swipes a ‘V’ shape), while the JHMDB dataset may have a weak connection with global trajectories. We show how these properties affect the performance and demonstrate the generalization of DD-Net in our ablation studies.

	SHREC Dataset	JHMDB Dataset
Number of samples	2,800	928
Training/ Testing Setup	1 Training Set 1 Testing Set	3 Split Training/ Testing Sets
Dimension of skeletons	3	2
subject	hand	body
Number of actions	14 and 28	21
Actions are strongly correlated to global trajectories	✓	✗

Table A.5: Properties of experimental datasets

A.3.4.2 Evaluation Setup

The SHREC dataset is evaluated in two cases: 14 gestures and 28 gestures. The JHMDB dataset is evaluated by using the manually annotated skeletons, and we average the results from three split training/testing sets.

In ablation studies, we explore how each DD-Net component contributes to the action classification performance by removing one component while remaining the others unchanged. Furthermore, we also explore how the performance varies with different model sizes by adjusting the value of *filters* in Fig. A.4.

A.3.4.3 Implementation Details

Since the DD-Net is small, it is feasible to put all of the training sets into one batch on a single GTX 1080Ti GPU. We choose Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$) [121] as the optimizer, with an annealing learning rate that drops from 1^{-3} to 1^{-5} . During the training, DD-Net only takes a temporal augmentation, which randomly selects 90% of all frames.

To demonstrate the superiority of DD-Net, we do not apply any ensemble strategy or pre-trained weights to boost the performance. To make DD-Net easily deployable to real applications, we implement it by Keras [186] with Tensorflow backend, which is “notorious” for its slow execution speed. Using other neural

network frameworks may make DD-Net even faster.

A.3.4.4 Result Analysis

The action classification results of SHREC dataset are presented in Table A.6 and more details are listed in their confusion matrix. The confusion matrix of 14 actions and 28 actions are Fig. A.6 and Fig. A.7, respectively. The action classification results of JHMDB dataset are presented in Table A.7.

Methods	Parameters	14 Gestures	28 Gestures	Speed on GPU
Dynamic hand [169] (CVPRW16)	-	88.2%	81.9%	-
Key-frame CNN [158] (3DOR17)	7.92 M	82.9%	71.9%	-
3 Cent [171] (STAG17)	-	77.9%	-	-
CNN+LSTM[187] (PR18)	8-9 M	89.8%	86.3%	238 FPS
Parallel CNN [159] (RFIAP18)	13.83 M	91.3%	84.4%	-
STA-Res-TCN [160] (Gesture18)	5-6 M	93.6%	90.7%	303 FPS
MFA-Net [173] (Sensor19)	-	91.3%	86.6%	361 FPS
DD-Net (filters=64, w/o global fast&slow motion)	1.70 M	55.2%	41.6%	-
DD-Net (filters=64, w/o global slow motion)	1.76 M	92.7%	90.2%	-
DD-Net (filters=64, w/o global fast motion)	1.76 M	93.3%	90.5%	-
DD-Net (filters=64)	1.82 M	94.6%	91.9%	2,200 FPS
DD-Net (filters=32)	0.50 M	93.5%	90.4%	3,100 FPS
DD-Net (filters=16)	0.15 M	91.8%	90.0%	3,500 FPS

Table A.6: Results on SHREC (Using 3D skeletons only)

Overall, although DD-Net takes fewer parameters, it can achieve superior results on SHREC dataset and JHMDB dataset. The confusion matrix also shows that DD-Net is robust to each action class. Despite the data property divergence existing, DD-Net demonstrates its generalization ability, which suggests it can accommodate a wide range of skeleton-based action classification scenarios.

From ablation studies, we can inspect that when actions are strongly correlated to global trajectories (*e.g.*, SHREC dataset), just using the JCD feature cannot produce a satisfactory performance. When actions are not strongly correlated to global trajectories (*e.g.*, JHMDB dataset), the global motion feature still helps to improve the performance, but not as significant as the previous case.

Methods	Parameters	Manually annotated skeletons	Speed on GPU
Chained Net [95] (ICCV17)	17.50 M	56.8%	33 FPS
EHPI [178] (ITSC19)	1.22 M	65.5%	29 FPS
PoTion [93] (CVPR18)	4.87 M	67.9%	100 FPS
DD-Net (filters=32, w/o global fast&slow motion)	0.46 M	71.4%	-
DD-Net (filters=32, w/o global slow motion)	0.48 M	74.9%	-
DD-Net (filters=32, w/o global fast motion)	0.48 M	75.8%	-
DD-Net (filters=32)	0.50 M	78.0%	3,100 FPS
DD-Net (filters=64)	1.82 M	77.8%	2,200 FPS
DD-Net (filters=16)	0.15 M	74.7%	3,500 FPS

Table A.7: Results on JHMDB (Using 2D skeletons only)

Such results agree with our assumptions: although the JCD feature is location-viewpoint invariant, it is isolated from global motions. The results also show that using the two-scale motion feature generates higher classification accuracy than only using a one-scale motion feature, which suggests that our proposed two-scale global motion feature is more robust to scale variation of motions. With the same components, DD-Net can adjust its model size by modifying the value of *filters* in CNN layers. We select 64, 32 and 16 as the values of *filters* to perform experiments. When DD-Net reaches the best performance on SHREC and JHMDB datasets, the values of *filters* are 64 and 32, respectively. It is worth noting that DD-Net can generate comparable results by only using 0.15 *million* parameters.

In addition, since DD-net employs one-dimensional CNNs to extract the feature, it is much faster than other models that use RNNs [181, 172, 182, 175] or 2D/3D CNNs [159, 188, 95, 93, 178]. During its inferences, DD-Net’s speed can reach around 3,500 FPS on one GPU (*i.e.*, GTX 1080Ti), or, 2,000 FPS on one CPU (*i.e.*, Intel E5-2620). While RNN-based models face great challenges for parallel processing (due to sequential dependency), our DD-Net does not have this issue because CNNs are used. Therefore, whether low-computational (*e.g.*, on small devices) or high-computational applications (*e.g.*, on parallel computing stations) are concerned, our DD-Net enjoys significant superiority.

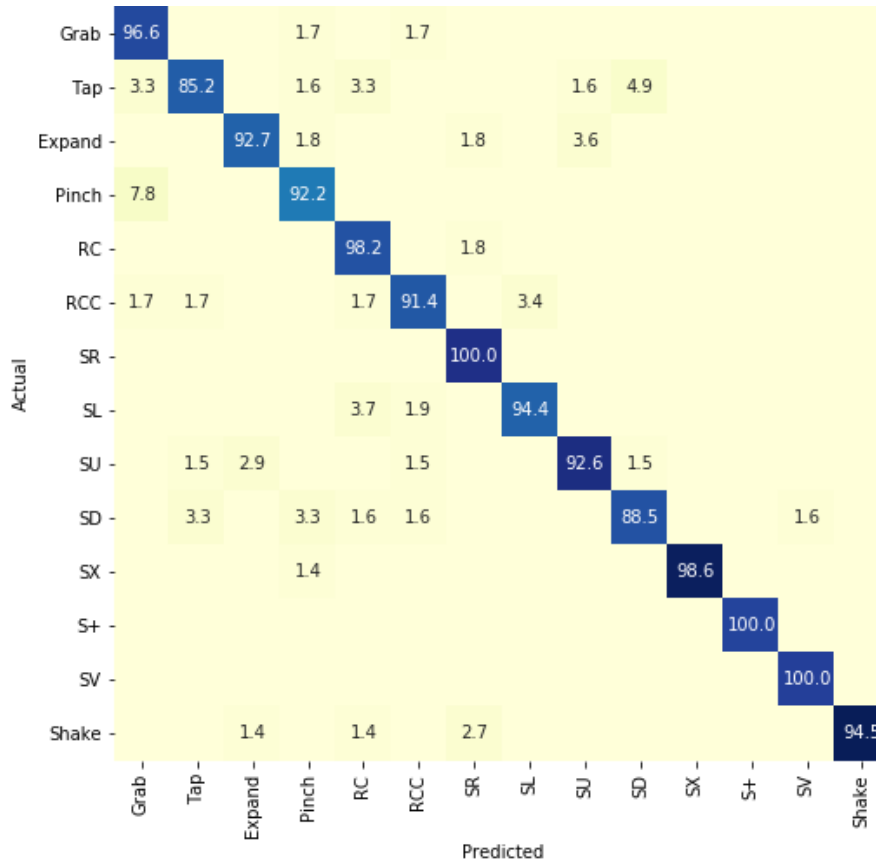


Figure A.6: Confusion matrix of SHREC dataset (14 hand actions) obtained by DD-Net.

A.3.5 Discussion

By analyzing the basic properties of skeleton sequences, we propose two features and a DD-Net for efficient skeleton-based action classification. Although DD-Net only contains a few parameters, it can achieve state-of-the-art performance on our experimental datasets. Due to the simplicity of DD-Net, many possibilities exist to enhance/extend it for broader studies. For instance, online action classification can be approached by modifying the frame sampling strategies; RGB data or depth data could be used to further improve the action classification performance; it is also possible to extend it for temporal action detection by adding temporal segmentation related modules.

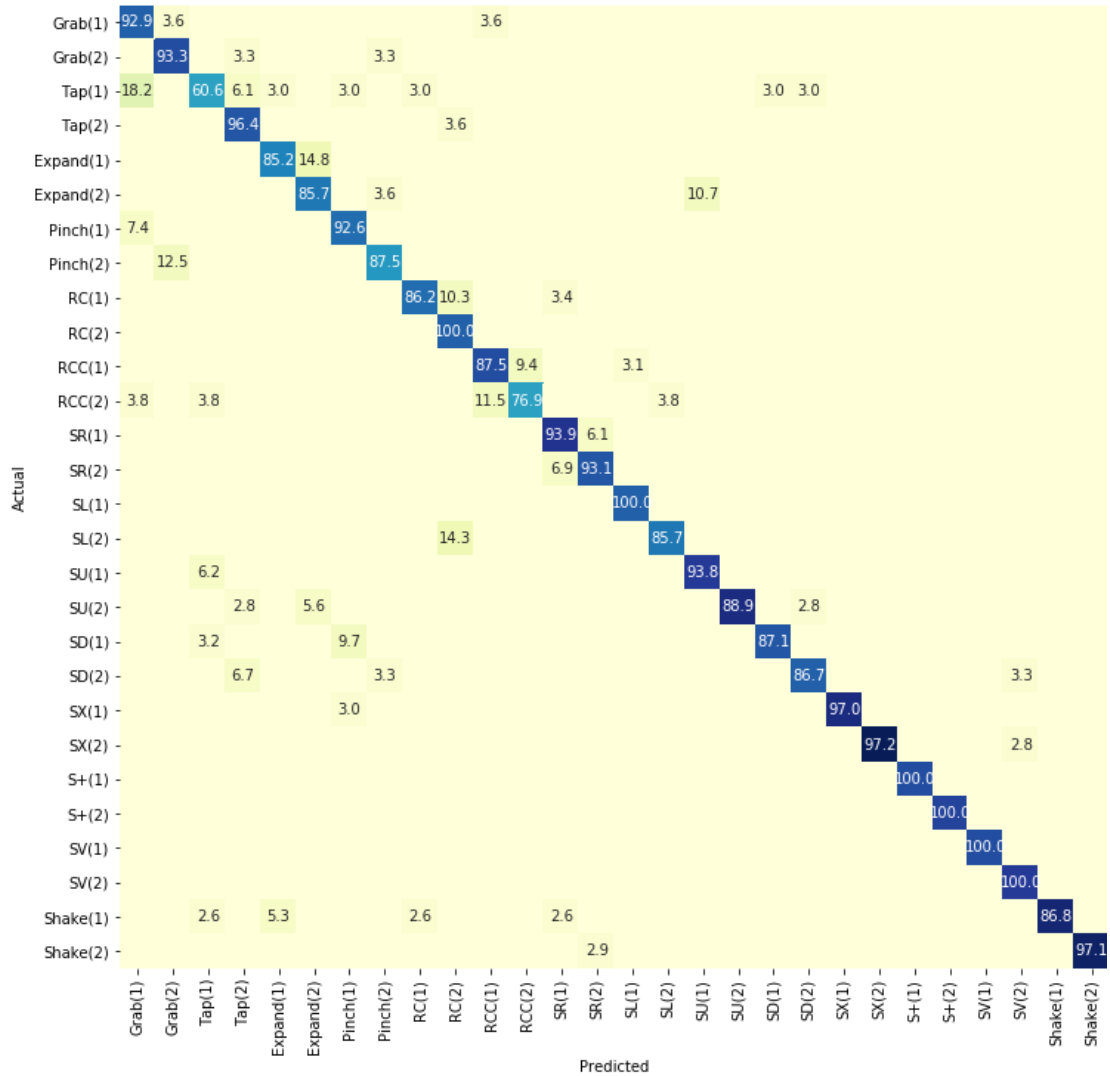


Figure A.7: Confusion matrix of SHREC dataset (28 hand actions) obtained by DD-Net.

A.4 3D Multiple Object Tracking

A.4.1 Overview

In the previous section, we have introduced our work in 2D MOT, however, in reality, we understand surrounding visual scenes in a 3D environment. For instance, we usually decide how to interact with surrounding people by first localizing and tracking them in a 3D egocentric coordinate: when we are walking down the street, we plan our path to avoid collisions by analyzing the trajectories of the surrounding people; when we see friends walking towards us, we might also walk to them and have a greeting. For applications (*e.g.*, social robotics) that also require visual scene understanding, performing multi-person localization and tracking in a 3D coordinate is strongly desired. With a study on the 3D Panoramic MOT, we can extend our Actor-identified Spatiotemporal Action Detection (ASAD) framework to the realistic 3D world.

A.4.2 Background

Typical single-view 3D coordinate localization methods fall into two categories: using depth sensors, or, using object size and camera geometry. Previous studies [189, 190] relied on depth sensors (*e.g.*, LiDAR) and instance segmentation to obtain the target location in a 3D camera coordinate. In practice, however, the instance person segmentation algorithm is imperfect in crowd scenes, resulting in the assigning of incorrect locations to a target person. To some extent, these methods are more suitable for multiple vehicle tracking [191], since they are rigid objects with known shapes and the distance between them is generally larger. In contrast, other methods [192, 193] infer 3D camera-coordinate locations by object bounding box size and camera geometry. However, there is a scale variance between standing persons and sitting persons in terms of bounding box height. Moreover, when a person is near the camera, only the upper body can be observed. Consequently, simply taking the bounding box height as a reference is inappropriate. Recently, a study [194] demonstrated that using the skeleton length can obtain more accurate locations than using bounding boxes. We embrace this idea into our framework to obtain target locations in a single-view 3D camera coordinate.

Conventional multi-person tracking takes two stages. The first detects each person by an object detector, and the second associates the cross-frame identities by considering their appearance similarity and trajectory trend [195, 196]. Most existing studies work on 2D/3D single-view [190, 193], or, 2D panoramic multi-target tracking [197, 198]. However, previous works have limitations: the 2D tracking results could not be directly used in some applications (*i.e.*, robotics) since the real-world coordinate in 3D; it is easy to lose the tracking target in a 3D narrow-angle-view coordinate since it only covers a part of the surrounding environment. Therefore, we propose 3D panoramic multi-target tracking to address the aforementioned issues.

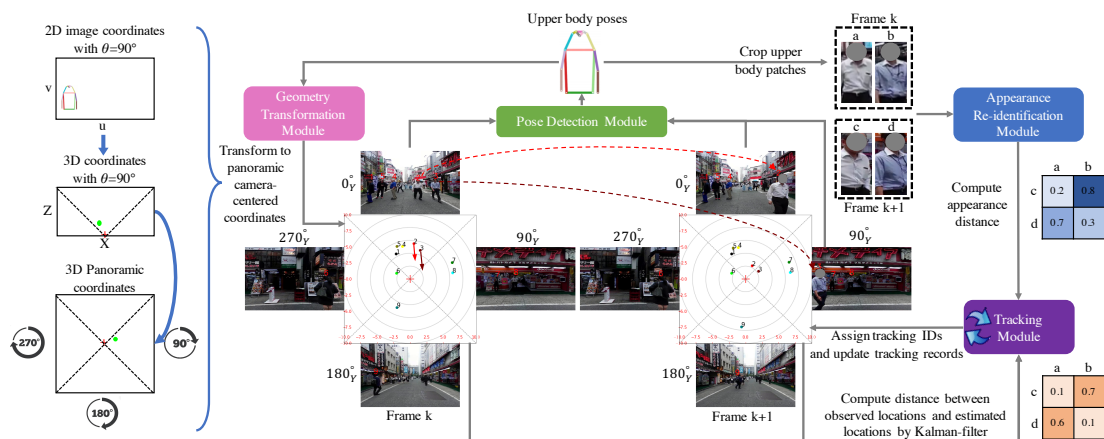


Figure A.8: Our framework for 3D panoramic multi-person localization and tracking.

We propose a novel framework for multi-person localization and tracking in a 3D panoramic coordinate with panoramic RGB videos. In our framework, 2D human poses are estimated for each person to obtain the 2D location and body height. Utilizing single-view intrinsic camera parameters, a person’s 3D location can be approximated by assuming the body height is a constant. We further transform locations from a 3D single-view camera coordinate to a 3D panoramic coordinate using extrinsic camera parameters. Unlike in a 2D image coordinate, the real-scale location and motion are preserved in a 3D coordinate. As a benefit, it is easier to harness the power of the Kalman filter to model human trajectories. To further address issues like occlusion and

miss detection, we associate the appearance similarity and the trajectory trend together to approach multi-person tracking.

We annotated a Multi-person Panoramic Localization and Tracking (MPLT) dataset to evaluate our framework. We also compared our framework with others on the KITTI dataset [199] and the 3D MOT dataset [200], where only single-view 3D localization and tracking results are provided.

A.4.3 Methodology

Our framework includes four modules: Pose Detection Module, Geometry Transformation Module, Appearance Re-identification Module, and Tracking Module (see Fig. A.12). They work seamlessly together to achieve the target goal.

A.4.3.1 Obtain 2D Person Poses

Similar to previous work [194], we use off-the-shelf PifPaf [201] as our Pose Detection Module to estimate 2D human poses. Depends on the need, 2D person poses can be obtained either by a top-down approach or a bottom-up approach. In the former, an object detector (*e.g.*, YOLO [202]) is used to acquire the 2D bounding box for each person, and then PifPaf estimates 2D poses within each single bounding box. Alternatively, PifPaf can simultaneously estimate 2D poses for all persons and assign them to each person, which is a bottom-up approach. Compared with the top-down approach, the bottom-up approach is faster but less accurate.

A.4.3.2 Coordinate Transformation

We build a Geometry Transformation Module to map person locations from 2D image coordinates to a 3D panoramic coordinate. In our setting, four single-view cameras are used to capture panoramic videos. By removing the overlapping areas, we obtain four single-view images with a 90° Horizontal Field of View at each frame. Following a clockwise path, we can assign each single-view image with view angle θ , where $\theta \in \{0_Y^\circ, 90_Y^\circ, 180_Y^\circ, 270_Y^\circ\}$.

Let $[u_\theta, v_\theta]^T$ be a point in the 2D image coordinate and let $[X_\theta, Y_\theta, Z_\theta]^T$ be the corresponding point in the 3D camera coordinates of each single view. Then,

we have

$$\begin{bmatrix} u_\theta \\ v_\theta \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} X_\theta \\ Y_\theta \\ Z_\theta \end{bmatrix}, \quad (6.13)$$

where \mathbf{K} is the intrinsic matrix.

To transform locations from 3D camera coordinates to a 3D panoramic coordinate, we construct an extrinsic matrix:

$$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \equiv [\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{4 \times 4} | \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3, \quad (6.14)$$

where $SO(n)$ denotes a Special Orthogonal Group with dimension n ; $\mathbf{0}$ indicates a zero vector; \mathbf{R} and \mathbf{t} are the 3D rotation matrix and the translation matrix. In our settings, all single-view coordinate centers are close to each other, so that \mathbf{t} can be approximated by a zero vector and \mathbf{R} only contain Y-axis rotation.

Accordingly, for location $[X, Y, Z]^T$ in a 3D panoramic coordinate, the complete projection matrix can be defined:

$$\mathbf{P}_\theta = \mathbf{K}[\mathbf{R}(\theta)|\mathbf{t}], \quad (6.15)$$

and we have

$$\begin{bmatrix} u_\theta \\ v_\theta \\ 1 \end{bmatrix} = \mathbf{P}_\theta \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (6.16)$$

At first glance, $[X, Y, Z]^T$ cannot be determined by $[u_\theta, v_\theta]^T$ in the above equation. However, we assume that real-world body height H_{body} is a constant value. Since the corresponding body height in a 2D image coordinate (*i.e.*, h_{body}) can be obtained by a pose estimator, for each person, the corresponding X and Z can be calculated by solving

$$\begin{bmatrix} u_\theta \\ h_{body} \\ 1 \end{bmatrix} = \mathbf{P}_\theta \begin{bmatrix} X \\ H_{body} \\ Z \\ 1 \end{bmatrix}, \exists H_{body} \approx \text{constant}. \quad (6.17)$$

Hence, we can transform a target from a 2D single-view image coordinate to a 3D panoramic coordinate. Since most real applications focus on the ground plane scenario, we treat $Y = 0$ for all the persons in the 3D panoramic coordinate.

A.4.3.3 Matching Cost

The appearance of people can be utilized as an important tracking cue to alleviate the occlusion issue in tracking. Although existing works exploit the entire body appearance [195, 196], we suppose that only using the upper body appearance can alleviate occlusion problems in crowd scenes. We further demonstrate this point in our experimental results of Table A.11. Since 2D body poses are estimated in this work, the upper body image patches can be cropped accordingly. We use an off-the-shelf model [203] as our Appearance Re-identification Module. Given an upper-body image patch, it extracts the correspondent appearance embedding vector.

In the tracking processes, appearance similarly is used to re-identify each person in the spatio-temporal domain. More specifically, the appearance cost between two consecutive frames is formulated as

$$\mathbf{C}_{i,j}^{app} = 1 - \frac{\mathbf{a}_i \mathbf{a}_j}{\|\mathbf{a}_i\| \|\mathbf{a}_j\|}, i \in \{1, \dots, N_i\}, j \in \{1, \dots, N_j\} \quad (6.18)$$

where $\mathbf{C}_{i,j}^{app}$ is the appearance cost of instance i of the previous frame to instance j of the current frame; N_i and N_j are the corresponding number of instances; \mathbf{a}_i and \mathbf{a}_j are the appearance embedding vectors with dimension 2048.

Apart from the appearance cue, the trajectory trend is also a critical cue to track targets. With regard to previous works [195, 196], Kalman filter [204] is commonly used to model the trajectory trend. In contrast with modeling the trajectory trend in a 2D image coordinate, modeling it in a 3D coordinate can alleviate the position and motion distortions, which simplifies the procedure of applying Kalman filter to model trajectories. To be consistent with \mathbf{C}^{app} at value range 0 – 1, we apply an exponential kernel to calculate the distance between detected locations and Kalman filter estimated locations that are normalized by H_{body} . The trajectory cost between two consecutive frames is defined by

$$\mathbf{C}_{i,j}^{traj} = 1 - \exp\left(-\frac{(\hat{X}_i - X_j)^2 + (\hat{Z}_i - Z_j)^2}{H_{body}^2}\right) \quad (6.19)$$

where $\mathbf{C}_{i,j}^{traj}$ is the trajectory cost of instance i of the previous frame to instance j of current frame. Additionally, $[\hat{X}_i, \hat{Z}_i]$ denotes the estimated location of instance i at **current** frame by Kalman filter, while $\mathbf{L}_{j,:} = [X_j, Z_j]$ presents the detected

location of instance j at current frame, where \mathbf{L} denotes the location values of all the detected instances.

We can simply associate $\mathbf{C}_{i,j}^{app}$ and $\mathbf{C}_{i,j}^{tra}$ by letting

$$\mathbf{C}_{i,j} = \mathbf{C}_{i,j}^{tra} + \mathbf{C}_{i,j}^{app}, \quad (6.20)$$

where $\mathbf{C}_{i,j}$ is the associate cost of matching instance i of the previous frame to instance j of the current frame. Then optimal assignment \mathbf{M}^* is obtained by minimizing the total cost

$$\mathbf{M}^* = \arg \min_{\mathbf{M}} \sum_i \sum_j \mathbf{C}_{i,j} \mathbf{M}_{i,j}, \quad (6.21)$$

where \mathbf{M} is a Boolean matrix. When row i is assigned to column j , we have $\mathbf{M}_{i,j} = 1$. Note that, each row can be assigned to at most one column and each column to at most one row. The optimization can be done by the Hungarian method.

A.4.3.4 Multi-person Tracking

In the Tracking Module, we create a tracking set \mathbb{T} to store and update tracked instances. At the k -th tracked frame, we obtain a set of 3D panoramic locations \mathbf{L} by Eq. (6.17) and the cost matrix \mathbf{C} by Eq. (6.20). In the first frame, all the observed locations are assigned to a tracking set. After that, the across-frame connections are determined by $\mathbf{M}_{i,j}$ and $\mathbf{C}_{i,j}$. When $\mathbf{M}_{i,j} = 1$ and $\mathbf{C}_{i,j}$ is smaller than a threshold ε , the instance i of frame $k - 1$ is likely to be the instance j of frame k . However, across-frame instances may not always be perfectly matched. For unmatched instance j , we assign it to \mathbb{T} as a new instance. For unmatched instance i , which is already recorded in \mathbb{T} , we reduce its lifespan by 1. While new instances come into the tracking area, old instances may also leave. Therefore, we delete unseen instances in the tracking set after 10 frames. We summarize this process in Algorithm 3.

A.4.4 Experiments

Experimental Datasets. We annotate a Multi-person Panoramic Localization and Tracking (MPLT) Dataset to enable model evaluation on 3D panoramic

multi-person localization and tracking. It represents real-world scenarios and contains a crowd of people in each frame. And, over 1.8K frames and densely annotated 3D trajectories are included. For comparison with related works, we also evaluate our framework on the KITTI [199] and 3D MOT [200] datasets. The properties of three experimental datasets are listed as follows:

Algorithm 2: Tracking algorithm

Input : k (current tracked frame number), \mathbf{C} (association cost matrix), \mathbf{L} (instance location matrix), \mathbb{T} (active instance set), ε (matching cost threshold)

- 1 **if** $k = 1$ **then**
- 2 Initialize active instance set $\mathbb{T} \leftarrow \emptyset$.
- 3 **for** $j \leftarrow 1$ **to** N_j **do**
- 4 $\mathbb{T}_j[\text{location}] \leftarrow \mathbb{T}_j[\text{location}] \cup \{\mathbf{L}_{j,:}\}$.
- 5 $\mathbb{T}_j[\text{lifespan}] = 10$.
- 6 **else**
- 7 Obtain \mathbf{M}^* by optimizing Eq. (6.21) with the Hungarian method.
- 8 Initialize the $N_j \times 1$ dimensional matching indicator vector $\mathbf{m} = \mathbf{0}$ for current frame k .
- 9 **for** $i \leftarrow 1$ **to** N_i **do**
- 10 $\mathbb{T}_i[\text{lifespan}] = \mathbb{T}_i[\text{lifespan}] - 1$.
- 11 **for** $j \leftarrow 1$ **to** N_j **do**
- 12 **if** $\mathbf{M}_{i,j}^* = 1$ **and** $\mathbf{C}_{i,j} < \varepsilon$ **then**
- 13 $\mathbb{T}_i[\text{location}] \leftarrow \mathbb{T}_i[\text{location}] \cup \{\mathbf{L}_{j,:}\}$.
- 14 $\mathbb{T}_i[\text{lifespan}] = 10$.
- 15 $\mathbf{m}_j = 1$.
- 16 **for** $j \leftarrow 1$ **to** N_j **do**
- 17 **if** $\mathbf{m}_j = 0$ (*i.e., instance j is unmatched*) **then**
- 18 Add one more active instance to \mathbb{T} :
- 19 $\mathbb{T}_{|\mathbb{T}|+1}[\text{location}] \leftarrow \mathbb{T}_{|\mathbb{T}|+1}[\text{location}] \cup \{\mathbf{L}_{j,:}\}$.
- 20 $\mathbb{T}_{|\mathbb{T}|+1}[\text{lifespan}] = 10$.
- 21 **if** $\mathbb{T}_i[\text{lifespan}] = 0$ **then**
- 22 Remove \mathbb{T}_i from \mathbb{T} .
- 23 **for** $l \leftarrow 1$ **to** $|\mathbb{T}|$ **do**
- 24 Update Kalman filter with $\mathbb{T}_l[\text{location}]$.
- 25 $\mathbb{T}_l[\text{location_estimated}] = [\hat{X}_l, \hat{Z}_l]$, estimated using the updated Kalman filter.

Output: \mathbb{T}

Table A.8: Properties of experimental datasets.

Dataset	3D single-view localization	3D single-view localization& tracking	3D panoramic localization& tracking
KITTI [199]	✓		
3D MOT [200]		✓	
MPLT			✓

Experimental Setup. Since off-the-shelf pose detector and appearance extractor are applied, we do not train any models in this work. For the KITTI and MPLT datasets, we apply the bottom-up pose estimation approach. For 3D MOT dataset, we apply the top-down pose estimation with the given public bounding boxes. Based on the properties of each dataset, we evaluate the model performance from different perspectives.

Experimental Results. In Table A.9, we report the localization precision under three thresholds for the KITTI Dataset. It was analyzed in work [2]: “MonoDepth neural network primarily uses the vertical position of objects in the image to estimate their depth, rather than their apparent size”. Training on the KITTI dataset, therefore, can give a strong prior to estimating the 3D location of a person rather than using the camera geometry and person size. We show the generalization of our method without using any KITTI data for training. In Table A.10, we compare our framework with others on the 3D MOT Benchmark², which targets at 3D single-view localization and tracking. We achieve the state-of-the-art performance (*i.e.*, 1st place of the public leaderboard) on the dominant criterion (*i.e.*, MOTA [10]), which outperforms the second place method by 1.5. For our proposal dataset MPLT, we list the performance of our framework and make it as a baseline (see Table A.11). Furthermore, we also show, due to the occlusion, selecting the whole body appearance may impair the model performance. The qualitative evaluation results are available on our project page³.

²https://motchallenge.net/results/3D_MOT_2015/

³<https://github.com/fandulu/MPLT>

Table A.9: Monocular-camera-based localization precision on KITTI Dataset. If distance from predicted locations to ground-truth location is within a threshold, it is correctly predicted.

Methods	Localization precision by threshold		
	< 0.5m	< 1.0m	< 2.0m
Mono3D [205] (trained on KITTI)	13.2%	23.3%	39.0%
SAMono[206] (trained on KITTI)	19.8%	33.9%	48.5%
MonoDepth [207] (trained on KITTI)	20.6%	35.4%	50.7%
MonoLoco [194] (trained on KITTI&COCO)	29.0%	49.6%	71.2%
MPLT (trained on COCO)	<u>22.0%</u>	<u>39.4%</u>	<u>63.5%</u>

Table A.10: 3D MOT Benchmark. $\uparrow(\downarrow)$ indicates that the larger(smaller) the value is, the better the performance. Multiple Object Tracking Accuracy (MOTA) is the dominant criterion. The details of the evaluation metrics were previously explained in [10].

Methods	MOTA \uparrow	MT \uparrow	ML \downarrow	FP \downarrow	FN \downarrow
AMIR3D [208]	25.0	3.0%	27.6%	2,038	9,084
MCFPHD [209]	39.9	25.7%	16.8%	3,029	6,700
GPDBN [210]	49.8	25.7%	17.2%	1,813	6,300
MOANA [211]	52.7	28.4%	22.0%	2,226	5,551
MPLT w/ DeepSORT	54.2	30.6%	20.9%	2,385	4,930

Table A.11: Comparing using features of the whole body and the upper body on the MPLT dataset. We evaluate localization and tracking performance within 10 m of the coordinate center.

Appearance Selection	Threshold	MOTA \uparrow
Whole body	< 0.5m	62.4
Whole body	< 1.0m	70.2
Upper body	< 0.5m	65.2
Upper body	< 1.0m	74.9

A.4.5 Discussion

We proposed a simple yet effective solution for 3D panoramic multi-person localization and tracking with panoramic videos. On two existing datasets, the effectiveness of our method is demonstrated by the promising performance. Meanwhile, a strong baseline is offered for our new benchmark dataset. Since our method can faithfully keep the realistic locations and motions for tracking targets in a 3D panoramic coordinate, it can help human-related video understanding applications. In future work, we plan to integrate our framework with a previous work [39] for automatically detecting human activities in a 3D panoramic coordinate.

A.5 Actor-specified Spatiotemporal Action Detection (ASAD) for 4K-resolution Aerial Videos

ASAD might be applied to 4K-resolution videos (*e.g.*, surveillance system). It is difficult to directly process large size videos and downscale the video may lose the detailed information for action recognition. We consider this issue to propose an efficient ASAD framework for 4K-resolution videos.

A.5.1 Overview

Surveillance cameras are commonly installed in city regions to increase public safety. However, it is inapplicable to densely set up surveillance cameras in sparsely populated regions (*e.g.*, suburb), while the safety concern is needed therein. Because some of the sparsely populated regions are not covered by tall trees or buildings, it is possible to periodically take surveillance videos by drones. Due to drones' mobility, a wide range of sparsely populated regions can be monitored at a low cost. Aerial surveillance videos have some special properties, which include: (1) to capture visual details from the sky, each frame of aerial surveillance videos is preferred to be an **4K-resolution** image (*e.g.*, 2160×3840); (2) relative to the entire aerial image, each actor appears to be a **tiny** object but could still contain a **large** amount of pixels, which are sufficient for obtaining his/her actions; (3) actors are **sparsely located**; (4) the drone could move **fast**, resulting in **significant relative position shift** of the targets even in adjacent frames; (t) the actor should be identified to know "who is doing what" in videos.

To address above issues, we specifically designed a ASAD framework for 4K-resolution drone videos (see Figure A.9). Although we treat MOT and AC as two fundamental element of ASAD, MOT could be further divided to actor detection phase and data association phase (Chapter 4). actor detection aims to locate each actor in a spatial domain by bounding boxes. An 4K-resolution aerial image, however, is too large to be the input of normal object detectors [212, 213, 214, 215], while down-scaling it could impair detection performance. As an alternative approach, an 4K-resolution aerial image could be cropped into smaller patches before performing actor detection. Some existing methods divide the entire aerial image into patches by a sliding window [216, 217, 218]. Although such methods

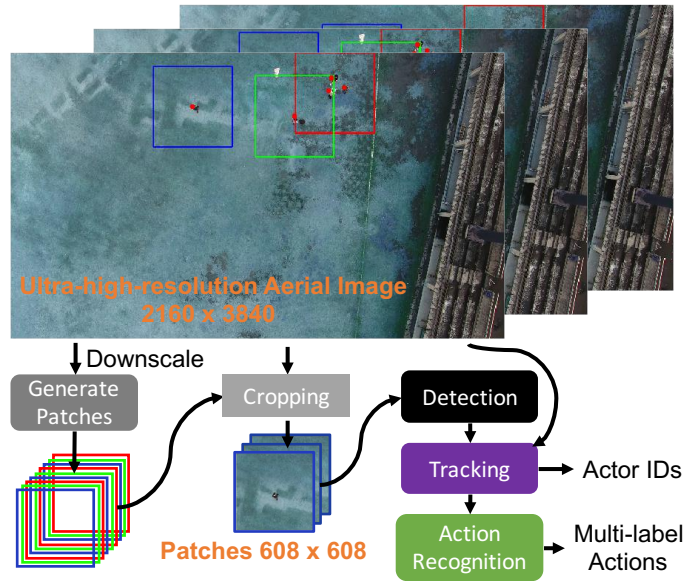


Figure A.9: Overview of our ASAD framework.

have considerably improved actor detection performance, they are inefficient when target objects are sparsely located. We propose a Clustering Region Proposal Network (C-RPN) to alleviate this issue. C-RPN works by only selecting patches that may include target objects. Subsequently, the number of selected patches could be fewer than using a sliding window when actors are sparsely located. Despite that ASAD estimates actions at each frame (*i.e.*, “is doing what”), spatio-temporal context is needed to obtain the actor motion information. Previous works [38, 40] obtain spatio-temporal tubes by extending bounding boxes from the central frame to nearby ones. In drone-recorded aerial videos, even if the absolute location of an actor is static, its relative location may shift remarkably due to the drone movement. To eliminate the effect of drones’ movement, we construct spatio-temporal tubes by a multi-object tracking method [51], and then align a spatio-temporal tube referred to its first frame. Since non-target objects might be included in the spatio-temporal tubes, action recognition performance could be affected. To tackle this issue, we assume the target actor can be consistently observed in his/her spatio-temporal tube while others may not. Based on this assumption, we propose a novel Spatio-temporal Attention Module (STAM) to obtain attention for the target actor in the spatio-temporal tube.

In summary, our **contributions** include: (1) we proposed a novel task, Actor-identified Spatiotemporal Action Detection (ASAD), to bridge the gap between existing SAD studies and the new demand of identifying actors. (2) we proposed a novel framework for multi-label ASAD on aerial surveillance videos, which outperforms other methods in our experiments.

A.5.2 Background

In this part, we briefly discuss related works of actor detection on aerial videos, model structures that could be potentially used for ASAD, and related datasets.

A.5.2.1 Actor Detection on Aerial Videos

Detecting tiny objects is a nontrivial problem and many studies are trying to tackle it. There could be two cases in tiny actor detection. One is the entire image has a low resolution and thereby the tiny objects only contain a few numbers of pixels. To improve the detection performance, amplification [219] and resolution enhancement [220] are applied. In another case, the object itself has plenty of pixels, but the object only constitutes a very small portion of the entire image so that it is relatively tiny. An 4K-resolution aerial image belongs to the second case and performing actor detection on the original image size is desired (Figure A.10).

Although the idea of transforming each frame of aerial videos into smaller patches for actor detection has been around for some time [216, 217, 218], it is only recently that region proposals and clustering have been jointly applied to reduce the number of patches when objects are sparsely located [221]. Using the downsized aerial image, promising regions that may contain objects can be learned by density map regression. Based on the predefined patch size, these regions can be further clustered by their relative distances.

We assume that a good clustering strategy should satisfy two conditions: **first, reducing the number of patches; second, keeping the object appearance complete in patches**. However, to some extent, these two conditions work against each other. Solely satisfying the first condition may lead to an object being partially cropped, while assigning each object to a patch can effectively satisfy the second condition but may introduce redundant patches. In the previous study [221], grid-based clustering is used. Nevertheless, it is limited by predefined grid size and location, and thus objects may be incompletely cropped

and further affect the bounding box detection. To resolve this issue, peak point Non-Max Suppression (NMS) and hierarchical clustering are used in our C-RPN, attempting to make every object complete in at least one patch.

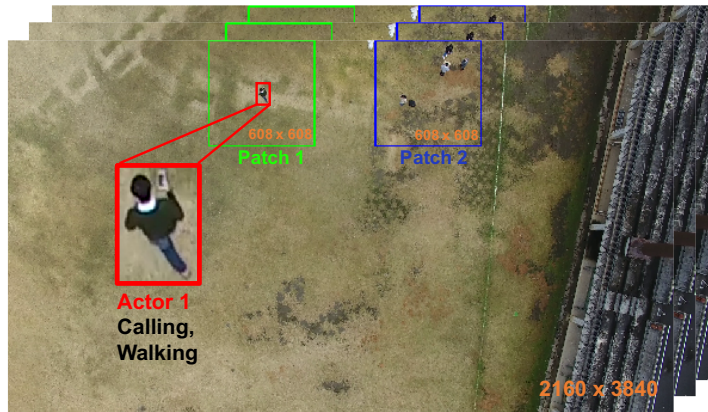


Figure A.10: Illustration of our room-in detection.

A.5.2.2 Applicable Model Structures for Multi-stage ASAD

Several models that can be used for ASAD are illustrated in Figure A.11. Figure A.11a, Figure A.11c and Figure A.11d learn actions and boxes by networks with end-to-end training, while Figure A.11b and Figure A.11e use independent detectors to generate boxes, and then connect boxes by MOT. This means generating spatio-temporal tubes and predicting actions are separate steps. Actions and boxes are jointly generated in Figure A.11a, and boxes are linked to form tubes by an offline tracking. To better model the spatio-temporal information, Figure A.11c and Figure A.11d learn features by a 3D ConvNet. The main difference is that Figure A.11c generates boxes and performs 2D Region of Interest (RoI) pooling for each frame, while Figure A.11d extends the central frame boxes to adjacent frames. Additionally, Figure A.11c and Figure A.11d.i) apply temporal pooling to fuse features, while Figure A.11d.ii) uses a 3D ConvNet to process features and obtain a better action recognition performance.

Owing to the divergence of the patch’s local coordinate and the entire image’s global coordinate, our inputs can only be aligned at the box level. Therefore, it is challenging to jointly detect bounding boxes and actions in our framework. Similar to Figure A.11b, our framework (*i.e.*, Figure A.11e) generates boxes by an independent detector and then connects boxes in the temporal domain by a

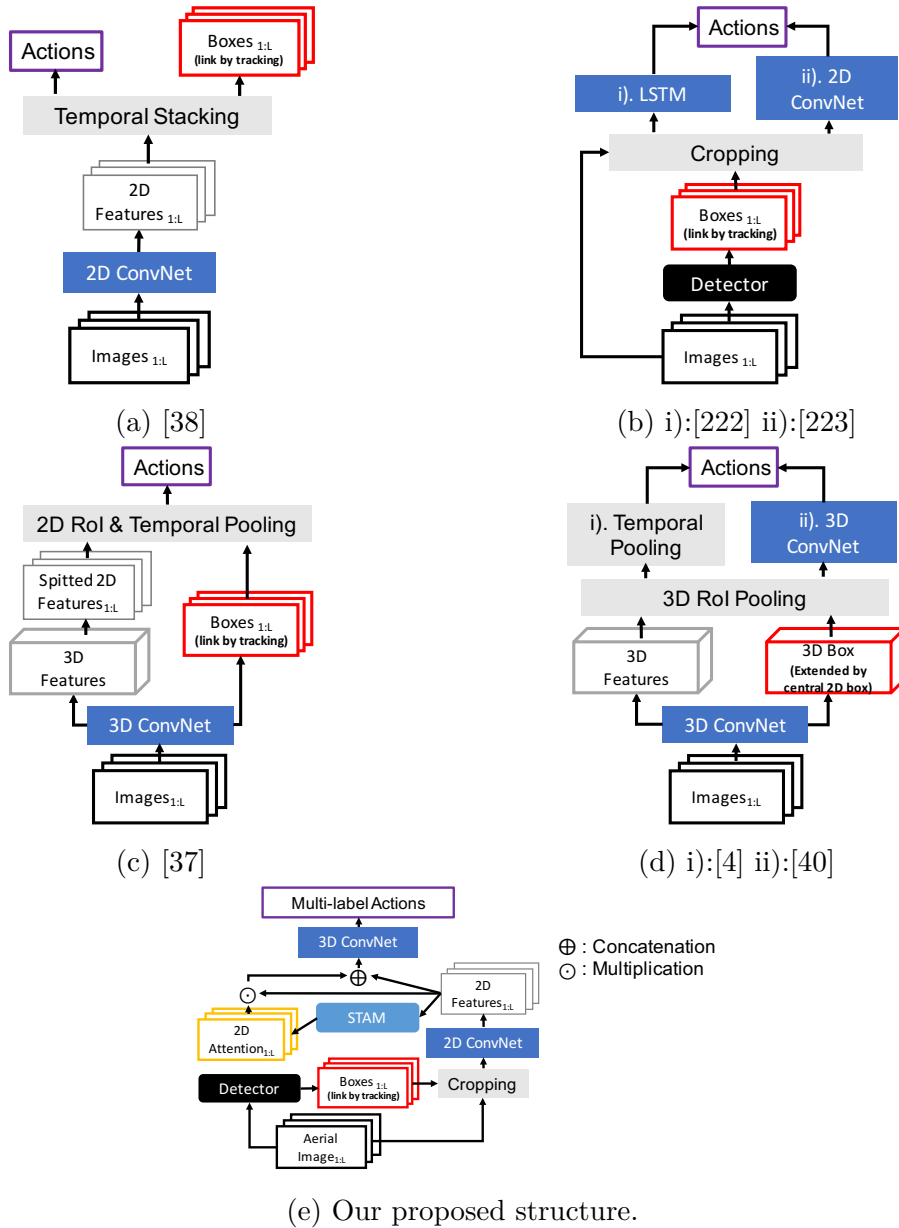


Figure A.11: **Applicable model structures for ASAD when only RGB data is used.** L denotes the number of frames used in the model. i) and ii) represent different models that share the same structure at the beginning.

independent MOT algorithm. Moreover, we propose a STAM to focus on the target object at each frame and use a 3D ConvNet for action recognition.

A.5.2.3 Related Datasets

The primary focus of conventional aerial video study are actor detection and tracking [224, 225, 226]. In this thesis, however, we concentrate on **multi-label ASAD in aerial videos**. We utilize Okutama-action dataset [6] for our experiments. The dataset comprises 43 minute-long drone-recorded aerial videos, with fully annotated bounding boxes in each frame and corresponding multi-label action classes. In all, there are 12 categories of human actions: Handshaking, Hugging, Reading, Drinking, Pushing/Pulling, Carrying, Calling, Running, Walking, Lying, Sitting and Standing. In the multi-label action annotation, one action class could associate with another one. For instance, “Reading” and “Sitting” could be assigned to the same actor at the same time. Besides, the actor ID (i.e., track ID) is given in this dataset.

A.5.3 Methodology

Our proposed framework coherently generates patches, bounding boxes, spatio-temporal tubes, 2D CNN features, attention maps, and multi-label action classes (see Figure A.12). Using a video frame of size 2160×3840 , our C-RPN first generates patches of size 608×608 . Based on selected patches, normal detectors (*e.g.*, YOLOv3-tiny [215]) can generate fine-grained bounding boxes for each actor. After that, fine-grained bounding boxes are connected to form spatio-temporal tubes by a MOT algorithm (*e.g.*, Deep SORT [196]). Next, we sample L frames from spatio-temporal tubes and obtain their corresponding 2D CNN features. STAM then takes 2D CNN features to generate attention maps that focus on target actors. In the end, the concatenation of 2D CNN features and their multiplication with attention maps, are used to estimate multi-label action classes by a 3D ConvNet. For the overall processing, it is a special multi-label SAD that serves for aerial surveillance videos.

A.5.3.1 Clustering Region Proposal Network (C-RPN)

The Clustering Region Proposal Network (C-RPN) takes downsized aerial images (544×960) as its input. Since each actor is relatively tiny compared with the aerial image, the coarse position of actor could be modeled by a 2D Gaussian density map. The mean of 2D Gaussian is the centroid of a actor and the covariance represents the uncertainty of this position, which is set to

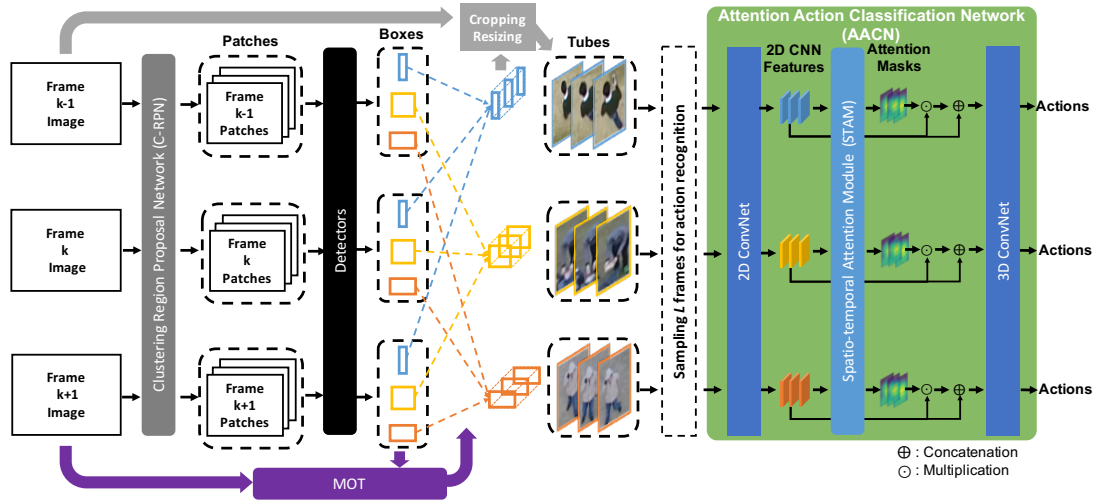


Figure A.12: **Architecture of the proposed framework.** Given each 4K-resolution aerial image of size 2160×3840 , C-RPN is utilized to select patches (608×608) that might contain actors. Based on selected patches, normal detectors are used to generate fine-grained bounding boxes for each actor. After that, fine-grained bounding boxes are further connected to be spatio-temporal tubes by a MOT algorithm. Next, we sample L frames from spatio-temporal tubes and obtain their corresponding 2D CNN features. STAM then takes 2D CNN features to generate attention maps that focus on target actors. In the end, the concatenation of 2D CNN features and their multiplication with attention maps, are used to estimate multi-label action classes by a 3D ConvNet.

be roughly half of the bounding box size. Thus, coarse actor locations can be learned by density map regression. Based on the predefined patch size, coarse actor locations can be further clustered by their relative distances and patches that may contain actors are generated (see Figure A.13).

At frame k , let the network output of C-RPN be H_k^{pred} and its ground truth be H_k^{true} . When both H_k^{pred} and H_k^{true} have a row number of R and a column number of C , we can represent them by

$$H_k^{pred} = \bigcup_{r=1}^R \bigcup_{c=1}^C h_{krc}^{pred}, H_k^{true} = \bigcup_{r=1}^R \bigcup_{c=1}^C h_{krc}^{true}, \quad (6.22)$$

where r and c are the row index and column index of the heat map, respectively; h_{krc}^{pred} and h_{krc}^{true} denote the pixel at position $[r, c]$ of H_k^{pred} and H_k^{true} , respectively.

The h_{krc}^{true} is generated by

$$h_{krc}^{true} = \sum_{i=1}^N \exp\left(-\frac{(r - p_{ki}(x) * s_1 * s_2)^2 + (c - p_{ki}(y) * s_1 * s_2)^2}{2\sigma_{ki}^2}\right); \quad (6.23)$$

$$h_{krc}^{true} = \begin{cases} 1, & \text{if } h_{krc}^{true} > 1; \\ h_{krc}^{true}, & \text{else.} \end{cases}$$

where $[p_{ki}(x), p_{ki}(y)]$ are the center coordinates of the i^{th} ground-truth bounding box. Since the overlapping boxes may generate values larger than 1, we clip the maximum value of h_{krc}^{true} at 1. The downscale factor from original image to C-RPN input is denoted as s_1 , and the down-sampling factor from C-RPN input to C-RPN output is denoted as s_2 . In this work, we set $s_1 \approx 1/4$ (to be divisible by s_2) and $s_2 = 1/8$.

More specifically, σ_{ki} , $p_{ki}(x)$ and $p_{ki}(y)$ are generated by

$$\sigma_{ki} = \frac{s_1 * s_2}{4} \left((x_{ki}^{max} - x_{ki}^{min}) + (y_{ki}^{max} - y_{ki}^{min}) \right);$$

$$p_{ki}(x) = \frac{s_1 * s_2}{2} (x_{ki}^{max} + x_{ki}^{min}); \quad (6.24)$$

$$p_{ki}(y) = \frac{s_1 * s_2}{2} (y_{ki}^{max} + y_{ki}^{min});$$

where $[x_{ki}^{min}, y_{ki}^{min}, x_{ki}^{max}, y_{ki}^{max}]$ are corner positions of the i^{th} ground-truth bounding box at frame k . Here, σ_{ki} is roughly half size of the bounding box i at frame k .

We modify a penalty-reduced pixel-wise logistic regression with focal loss [227] and let it be our loss function $\mathcal{L}_{raw_pos,k}$ as follows:

$$\mathcal{L}_{raw_pos,k} = - \sum_{r=1}^R \sum_{c=1}^C \begin{cases} (1 - h_{krc}^{pred})^\alpha \log(h_{krc}^{pred}), \\ \text{if } h_{krc}^{true} = 1; \\ (1 - h_{krc}^{true})^\beta (h_{krc}^{pred})^\alpha \log(1 - h_{krc}^{pred}), \\ \text{otherwise;} \end{cases} \quad (6.25)$$

where α and β are hyper parameters for focal loss and we follow work [227] to set α and β to be 2 and 4, respectively.

Ideally, each object center is a peak point on this density map, thus, we can apply peak point Non-Max Suppression (NMS) to obtain corresponding peak

points. Nonetheless, there is no magic in the network of C-RPN, and it is still suffering the dilemma of detectors in setting a confidence threshold: better precision, or better recall. In C-RPN, although false-positive (FP) peak points may generate redundant patches, such a redundancy has little effect on the final fine-grained actor detection. Therefore, we set a low confidence threshold for peak point NMS to obtain peak points, regardless of it may end up with low precision and high recall.

Because peak points could be sparsely distributed, grouping neighboring peak points to guide patch generalization can reduce the number of patches. As we have discussed in related works, grid-based clustering may not fit our requirements as it may incompletely crop actor appearance in all patches. To make a trade-off between reducing the number of patches and preserving the objects appearance, we choose hierarchical clustering. In hierarchical clustering, by adjusting the threshold distance to generate suitable overlapping regions dynamically, we could make actor appearance complete in at least one patch.

We do not need to specify how many actors are included in each patch, because another object detector (*e.g.*, YOLOv3) will take patches as inputs to generate bounding box for each actor. Since overlapping patches could be generated, we not only have duplicated boxes in the same patch, but also have duplicated boxes on the overlapping regions between patches. In our approach, therefore, we only perform bounding box NMS once after transferring bounding boxes from the patch coordinate to the original aerial image coordinate.

A.5.3.2 The Multiple Object Tracking Module

In our approach, we first employed Deep SORT [196], a traditional MOT method, to link bounding boxes into spatio-temporal tubes. Deep SORT takes an IoU (Intersection over Union) descriptor, an appearance descriptor, and a Kalman filter to perform bipartite bounding box assignments across frames. The appearance descriptor, which is used to overcome occlusion and long-time tracking issues, is a CNN network trained on a actor re-identification dataset [228] by a Cosine Softmax Classifier [229]. We then utilized our proposed Offline ReID-dominated MOT (Chapter 4) to show an improved results.

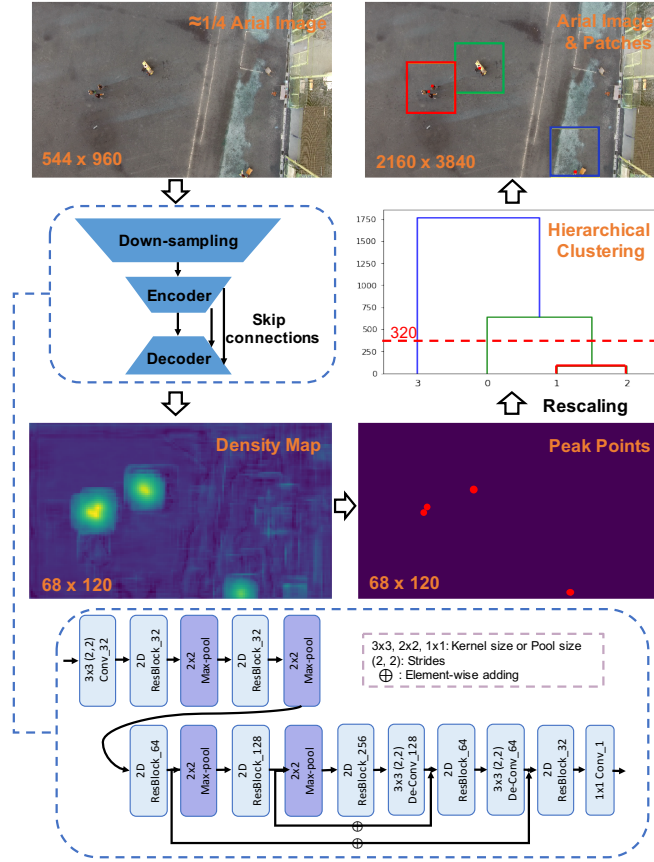


Figure A.13: The demonstration of generating patches by C-RPN. The downscale factor $s_1 \approx 1/4$, and the down-sampling factor $s_2 = 1/8$.

A.5.3.3 The RGB-based Action Classification Module

After obtaining the spatio-temporal tube for each actor, we obtain their actions at each frame by a novel Attention Action Classification Network (AACN). Since ASAD focuses on instantaneous actions other than long-term actions, we only take a short-term temporal context and sample L frames from each spatio-temporal tube for action recognition. Frames within 2 seconds (*i.e.*, 60 frames in 30 FPS videos) ahead of the target frame are excluded. For an actor whose track ID is n , we denote the earliest and latest frames in the corresponding spatio-temporal tube as k_{min} and k_{max} , respectively. Setting k_{max} as the target frame, then L frames are sampled to form a set $\{x_0^n, x_1^n, \dots, x_L^n\} \in X_{k_{max}}^n$ for action recognition. The details of our online sampling strategy are described in Algo-



Figure A.14: Visualizations of optical flow maps generated by PWC-Net [5], using Okutama-action Dataset. Due to the tiny size of actor and the drone camera movement, it is challenging to obtain actor motion information from the optical flow.

rithm 3.

Algorithm 3: On-line sampling from a spatio-temporal tube

Input : Spatio-temporal tube $T_{[k_{min}:k_{max}]}^n$

- 1 **if** $\text{len}(T_{[k_{min}:k_{max}]}^n) < L$ **then**
- 2 $X_{k_{max}}^n \leftarrow \{T_{[k_{min}:k_{max}]}^n + \text{Repeat Padding with } T_{k_{max}}^n\};$
- 3 **else**
- 4 $\delta = \text{len}(T_{[\max(k_{min}, k_{max}-60):k_{max}]}^n) // L;$
- 5 $X_{k_{max}}^n \leftarrow \{\text{Randomly choose } L \text{ frames from } T_{[\max(k_{min}, k_{max}-60):k_{max}]}^n \text{ with the interval } \delta\}.$

Output: $X_{k_{max}}^n$

Instead of directly processing RGB data $X_{k_{max}}^n$ by 3D ConvNet, we extract their corresponding 2D CNN features $\{f_1^n, f_2^n, \dots, f_L^n\} \in F_{k_{max}}^n$ at the first step. Then, we proposed a Spatio-temporal Attention Module (STAM), which is a 3D encoder-decoder with skip connections, to generate attentions maps $\{a_1^n, a_2^n, \dots, a_L^n\} \in A_{k_{max}}^n$ by encoding and decoding the global spatio-temporal representation of $F_{k_{max}}^n$. After that, we perform element-wise multiplication between $F_{k_{max}}^n$ and $A_{k_{max}}^n$, and concatenate with $F_{k_{max}}^n$ to obtain a representation that can selectively

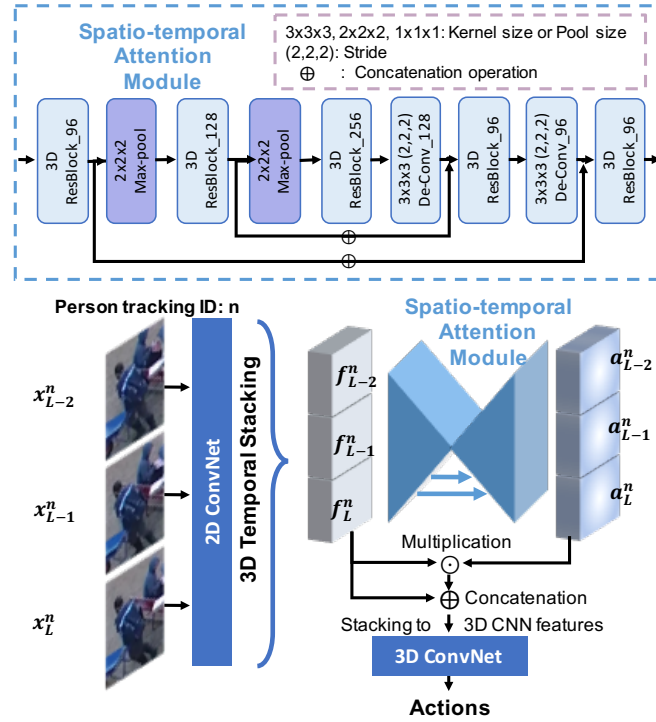


Figure A.15: An illustration of proposed Attention Action Classification Network (AACN), with its Spatio-temporal Attention Module (STAM). Three frames are used in this illustration, where $\{x_{L-2}^n, x_{L-1}^n, x_L^n\}$ are RGB features sampled by Algorithm 3, and they are fed to 2D ConvNet to generate 2D CNN features $\{f_{L-2}^n, f_{L-1}^n, f_L^n\}$. STAM takes stacking 2D CNN features to obtain corresponding attention maps $\{a_{L-2}^n, a_{L-1}^n, a_L^n\}$. The multiplication results of 2D CNN features and attention maps are concatenated with 2D CNN features again, and then be used to estimate multi-label actions by a 3D ConvNet.

focus on the target actor across all frames. Finally, aforementioned 2D CNN features are stacked to be 3D CNN features, which are then fed to a 3D ConvNet to estimate multi-label action classes (see Figure A.15).

Although it is common to utilize optical flow for action recognition, we do not use it in our framework. In drone-recorded aerial videos, even if the absolute location of an instance is static, its relative location may have a huge change across nearby frames, which is caused by the drone camera movement and tiny object size. In Okutama-action data, we use a state-of-the-art optical flow generator [5]

to produce optical flows between nearby frames, and show them in Figure A.14. We can see, it is hard to identify the movement of each actor in the optical flow map.

A.5.4 Experiments

A.5.4.1 Training and Testing Setup

By following the previous work [6], Okutama-action dataset is split into a training set with 33 aerial videos and a testing set with 10 aerial videos.

For C-RPN, the Adam [121] optimizer with a learning rate 0.001 is applied for the first 50 epochs and then the learning rate is changed to be 0.0001 for another 150 epochs. The batch size is set up to be 8. Images and their corresponding density maps are jointly augmented by Albumentations [230].

We perform a peak point detection on a validation set (*i.e.*, 20% of the training set) and find that a density map can reach the confidence of $0.5 \sim 1.0$ and $0.0 \sim 0.1$ at the target and the non-target positions, respectively. To reach a high recall on the testing set, we set the peak point NMS confidence threshold as 0.3. We search the maximum actor bounding boxes size in Okutama-action dataset to decide the distance threshold in peak point NMS. More specifically, the maximum actor bounding box size is about 200 on the original size image. Considering the total downscale from the original size image (2160×3840) to the output density map (68×120) is about 32, the maximum actor size on output density map is about 6. Since distance threshold should be an odd number, we take value 5 here. Using Python code, peak point NMS can easily be implemented by

```
1 from scipy.ndimage import maximum_filter
2 Peaks_map = (H_pred>0.3)*
3             (H_pred==maximum_filter(H_pred,
4             footprint = np.ones((5,5))))
```

Listing 6.1: Peak piont NMS

For other detectors used for comparison, as R-FCN-ResNet50 [213], Retinanet-ResNet50 [214], SSD-ResNet50 [212] and YOLOv3-tiny [215], we take their pre-trained weights on COCO dataset [99] and fine-tune them on our experimental datasets by their default training strategy.

Method	AP@0.5 \uparrow	Speed for entire image (FPS) \uparrow	Average patches \downarrow
Using entire downscale image of size 608×608			
R-FCN-ResNet50 [213]	53.5	6	-
Retinanet-ResNet50 [214]	56.3	10	-
SSD-ResNet50 [212]	52.3	18	-
YOLOv3-tiny [215]	52.4	120	-
Using Patches of size 608×608 (without downsizing)			
Sliding (stride=388,404) [218]+YOLOv3-tiny	82.0	3	45.0
Sliding (stride=580,580) [218]+YOLOv3-tiny	79.4	5	28.0
C-RPN (Grid: $grid_{size}=216 \times 384$) [221]+YOLOv3-tiny	77.5	25	3.9
C-RPN (Grid: $grid_{size}=270 \times 480$) [221]+YOLOv3-tiny	78.3	28	3.7
C-RPN (Hierarchical: $d_{threshold} = 128$)+YOLOv3-tiny	85.0	26	3.8
C-RPN (Hierarchical: $d_{threshold} = 320$)+YOLOv3-tiny	85.2	30	3.1
C-RPN (Hierarchical: $d_{threshold} = 512$)+YOLOv3-tiny	82.9	38	2.2

Table A.12: **Actor spatial detection performance on Okutama-action dataset.** The symbol $\uparrow(\downarrow)$ indicates that the larger(smaller) the value, the better the performance.

To train AACN, we equally sample 64 ground-truth spatio-temporal tubes from each action class, and then sample $X_{k_{max}}^n$ from each spatio-temporal tube (see Algorithm 3). As only part of the training samples are included in one epoch training, it takes more iterations to get converged. We also apply the Adam optimizer for it, with learning rate 0.001, 0.0001, and 0.00001 for each 500 epochs. The batch size is set up to be 16. We perform the same data augmentation, *i.e.*, flipping, rotation, resizing, and cropping to all samples in $X_{k_{max}}^n$. During the inference process, Algorithm 3 is applied again to obtain inputs for the inference process.

Even though we are working on ASAD with large-size aerial videos, our framework decomposes the whole problem into multiple simple tasks. Thus, all our experiments can be implemented on a single NVIDIA TITAN X GPU.

A.5.4.2 Performance Evaluation

Our proposed metrics evaluate the multi-label ASAD performance by two steps. Firstly, we evaluate actor detection performance, by using the AP@0.5 metrics [99]. Secondly, we evaluate multi-label action recognition performance for positively detected samples (*i.e.*, a sample with $mAP \geq 0.5$). We jointly

inspect the performance of two steps to obtain the overall multi-label ASAD performance.

A.5.4.3 Actor Spatial Detection Evaluation

For the actor spatial detection evaluation, our main purpose is to verify three assumptions: (1) compared with detectors that work on the downsized aerial image (608×608 with padding), although using our proposed C-RPN may take more running time, it should improve the actor detection performance; (2) compared with partitioning the entire aerial image (2160×3840) into patches with a sliding window [218], our C-RPN should be faster when actors are sparsely located; (3) in contrast to grid-based clustering [221], using hierarchical clustering with a proper distance threshold can keep the complete appearance in at least one patch so that our method can achieve better actor detection performance.

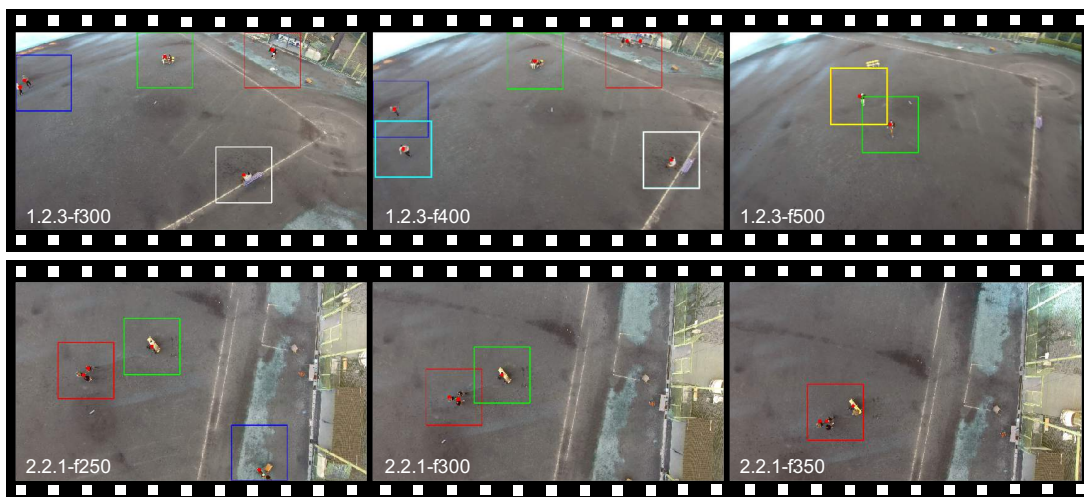


Figure A.16: **Patch proposals in Okutama-action testing sets, which are generated by C-RPN.** Generated peak points are marked by red, and patches are enclosed by colorful rectangles. The first row shows three sequential frames (*i.e.*, 300, 400 and 500) in video 1.2.3. The second row shows three sequential frames (*i.e.*, 250, 300 and 350) in video 2.2.1. To efficiently cover target actors, clusters automatically merge and split, based on the relative distance within peak points.

For detectors that take the entire aerial image as input, we standardize their input size to be 608×608 by padding, since it is difficult to train and test a

detector with larger input size. When the sliding window passes the aerial image margin, we pad zeros to the inputs. To reach a fast speed, we choose YOLOv3-tiny [215] as the base detector in our framework. Although our default setting is hierarchical-clustering C-RPN, for a fair comparison with previous work [221], we form a grid-clustering C-RPN by solely replacing the clustering method.

To further check the generalization of our C-RPN, we perform patch generation on VisDrone dataset [226], which has a complicated background. Since our target object is “actor”, we select objects with label “pedestrian” and “people” from VisDrone dataset, and then define them as label “actor” in our experiment. The network of C-RPN is trained by the training set, and the patch generation is performed on the testing set, where the ground-truth has not been unreleased yet.

The qualitative results of dense map estimation and patch generation are shown in Figure A.17, Figure A.18 and Figure A.19. By using a low confident threshold value, the rough locations of “actor” objects could be successfully detected from complex backgrounds. Meanwhile, a few of redundant patches are generated. However, the overall results coincide with our aims: reducing the number of patches but keeping the object appearance complete in a least one patch.

The qualitative results of our patch generation and bounding box estimation are shown in Figure A.16 and Figure A.20, respectively. The quantitative results of the Okutama-action testing set are shown in TABLE A.12. Taking the original-size aerial images (2160×3840), our C-RPN + YOLOv3-tiny achieves 85.2 AP@0.5 in terms of “actor” actor detection, which remarkably outperforms detectors that utilize downsized aerial images. Besides, by using C-RPN, the final actor detection performance is even better than using a sliding window, since some ambiguous background might be excluded by C-RPN in advance. Last but not least, because we try to make the actor appearance complete in at least one patch, the performance of hierarchical-clustering C-RPN outperforms grid-clustering C-RPN [221]. Moreover, we quantitatively calculate the average number of patches generated by each method in Okutama-action testing set. When hierarchical-clustering C-RPN reach the best detection performance, it only generates 3.1 patches averagely on Okutama-action testing set, which is more efficient than



Figure A.17: Visualizations of dense map estimation and patch generation on VisDrone testing dataset (1/3). The size of above aerial images is 1080×1920 and the patch size is 608×608 . Proposal patches are bounded by colorful rectangles.



Figure A.18: **Visualizations of dense map estimation and patch generation on VisDrone testing dataset (2/3).** The size of above aerial images is 1080×1920 and the patch size is 608×608 . Proposal patches are bounded by colorful rectangles.



Figure A.19: **Visualizations of dense map estimation and patch generation on VisDrone testing dataset (3/3).** The size of above aerial images is 1080×1920 and the patch size is 608×608 . Proposal patches are bounded by colorful rectangles.

sliding window approach and similar to the grid-clustering C-RPN. Therefore, our approach can achieve a comparable speed of 30 FPS on the full resolution data.

A.5.4.4 Multi-label Action Classification Evaluation

We modify Better-AVA model [40] to jointly estimate multi-label actions and bounding boxes. Its inputs are L frames of downscale aerial images (608×608 with padding), which are sampled near the target frame. Due to the limitation of our computational resource, we choose $L = 5$ for it.



Figure A.20: **Examples of multi-label action classification results in our framework.**

To inspect whether our AACN can improve the action recognition performance by introducing spatio-temporal attention, we construct an ablation study by replacing AACN with I3D [87] and Lite ECO [88] in our framework. The results of applying our proposed metrics are shown in Table A.13.

Compared with Better-AVA, our framework achieves better performance in both actor detection and multi-label action recognition. Besides, our framework is faster than Better-AVA on our target task. Considering our framework decomposes the whole pipeline into several independent steps, less memory cost is needed in our framework.

Method	HL@0.5↓	AP@0.5↑	Speed (FPS)↑
Off-line multi-label ASAD			
Input: $L \times 608 \times 608$ (L frames of downsized aerial images with padding)			
Better-AVA [40] ($L=5$)	0.20	0.54	8
On-line multi-label ASAD			
Input: $L \times 96 \times 96$ (L frames of cropped images from spatio-temporal tube)			
Replacing AACN by I3D [87] in our framework ($L=8$)	0.14	0.85	14
Replacing AACN by Lite ECO [88] in our framework ($L=8$)	0.15	0.85	15
Our framework ($L=8$)	0.13	0.85	14

Table A.13: **Multi-label ASAD results. Action Identification performance will be separately evaluated.** The symbol $\uparrow(\downarrow)$ indicates that the larger(smaller) the value, the better the performance. Only RGB data is used in this test. Note, we choose $L = 5$ for Better-AVA due to computation memory limitation and it has to be an odd number. While other models utilize $L = 8$ since instantaneous actions are defined in ASAD. Except for Better-AVA, other action detection models use bounding boxes that are generated by C-RPN + YOLOv3-tiny, which achieves $AP@0.5=85.2$.

Through introducing spatio-temporal attentions, our AACN performs better than I3D and Lite ECO, in terms of action recognition in our target task. Examples of attention maps generated by STAM can be visualized in Figure A.21, which shows that STAM can learn to focus on the target actor in an unsupervised manner.

A.5.5 Actor Identification Evaluation

To compensate for the actor identification evaluation, we utilized part of MOT evaluation metrics for actor identification evaluation, as IDF1 (ratio of correctly identified detections), MT (mostly tracked targets), ML (mostly lost targets), and ID Switches [50, 51]. The actor identification performance are shown in Table A.14. By sacrificing a little speed, our framework can generate better detection and leads to better actor identification. By using our ReID-dominated MOT (Chapter 4), the actor identification is improved compared with using the



Figure A.21: Visualization of attentions for the target actor. We assume that the target actor consistently appears in his/her spatio-temporal tube while others may not. The attention mask is learned in an unsupervised manner.

DeepSORT [56].

Approach	IDF1 (%) \uparrow	MT (%) \uparrow	ML (%) \downarrow	# ID Sw. \downarrow
entire downscale image of size 608×608				
Retinanet-ResNet50 [214] & DeepSORT [56]	53.8	32.0	32.7	234
YOLOv3-tiny [215] & DeepSORT [56]	47.4	27.4	34.6	256
Using Patches of size 608×608 (without downsizing)				
Our framework w/ YOLOv3-tiny [215] & DeepSORT [56]	62.8	45.0	15.8	198
Our framework w/ YOLOv3-tiny [215] & Our Offline ReID-dominated MOT (Chapter 4)	63.9	46.2	14.5	186

Table A.14: Evaluation performance on actor identification by referring selected MOT metrics.

Note that, in our designing, the overall Actor-identified Spatiotemporal Action Detection (ASAD) performance should consider multiple metrics, including Table A.12, Table A.13, Table A.14.

A.5.6 Discussion

To automatically perform Actor-identified Spatiotemporal Action Detection (ASAD) in 4K-resolution drone videos, we specifically propose a novel multi-label ASAD framework and corresponding evaluation metrics. Our framework gives the flexibility to replace its detector and tracker based on the need, which makes it

possible to train and infer all modules on a single GPU. Thus, our framework can be more suitable than existing solutions for multi-label ASAD in aerial videos.