One-Shot Deep Model for End-to-End Multi-Person Activity Recognition

Shuhei Tarashima
tarashima@acm.org

Innovation Center
NTT Communications Corp.
Tokyo, Japan

Abstract

In this work we tackle the multi-person activity recognition problem, where actor detection, tracking, individual action recognition and group activity recognition tasks are jointly solved given an input sequence. Since related works in the literature only deal with parts of the whole problem despite sharing similar architectures, trivial combinations of them result in slow and redundant pipelines and miss the opportunity to leverage inter-task mutual dependency. This motivates us to introduce a novel deep learning model, named *TrAct-Net*, that can jointly solve all the above tasks in a unified architecture. A new multi-branch CNN in TrAct-Net makes inference efficient and simple, and a novel relation encoder successfully takes both positional and identical relation of detections into consideration to boost both individual action and group activity recognition performances. The whole network is trained end-to-end using a multi-task learning framework. To the best of our knowledge, TrAct-Net is the first end-to-end trainable model to solve the whole problem in a one-shot manner. Experiments on public datasets demonstrate that TrAct-Net achieves superior performance to combinations of state-of-the-arts with much fewer model parameters and faster inference speed.

1 Introduction

In videos captured for various computer vision applications such as surveillance, sports analytics and autonomous driving, a vast range of individuals move and perform their own actions while interacting with each other to form group activities among them. Understanding such dynamic scenes is quite challenging, since it consists of multiple sub-tasks including detection [49], tracking [48], individual action recognition and group activity recognition [18], and the performance of each sub-task heavily depends on the others. Here we address this multi-person activity recognition problem, where all the above tasks should be solved jointly from a given sequence (*cf.* Figure 1).

Despite the mutual dependency between sub-tasks, in the literature existing research efforts end up with parts of the whole problem using task-specific models. For example, in a line of group activity recognition research [2, 17, 18, 21, 30, 44], detection or tracking results are assumed to be from separate models or systems. Notice that these task-specific models are trained individually though mildly sharing common architectures (*e.g.* CNN backbones [10, 20, 36, 37]). One straightforward way to address the whole problem is to aggregate the results of these methods. However, this often results in slow and excessively complicated systems with huge computational overhead, which is not desirable for real-world AI.

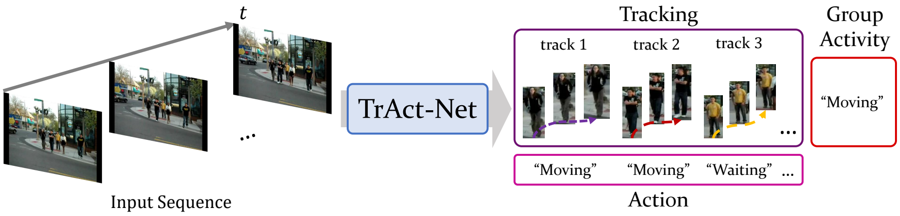


Figure 1: Given an input sequence, our TrAct-Net simultaneously detects actors and associates them in a sequence, while classifying individual actions and their group activity.

Also, this naive aggregation misses important opportunities for sharing computation, which potentially leads to leveraging a synergy between sub-tasks.

Based on this observation, in this work we study a unified architecture for multi-person activity recognition. To improve efficiency and simplicity, we introduce a novel convolutional neural network (CNN) module which has multiple branches to simultaneously produce a set of embeddings for detection, re-identification (re-ID) for tracking, and action/activity recognition from frames in a given sequence. Additionally, to exploit mutual dependency between sub-tasks, we propose a new relation encoder module that transforms embeddings from the CNN branches for individual action and group activity classification. The above proposals are integrated into a one-shot deep learning model, which is trainable end-to-end using a multi-task learning framework. In this paper we name this model *TrAct-Net*, and evaluate it through extensive experiments on major group activity recognition benchmarks [13, 22]. To summarize, our contributions in this work are:

- We propose TrAct-Net, a one-shot deep model that can jointly perform actor detection, tracking, individual action recognition and group activity recognition. To the best of our knowledge, TrAct-Net is the first unified solution for the whole multi-person activity recognition problem.
- We introduce a multi-branch CNN module that yields all the embeddings for every multi-person activity recognition sub-task in a one-shot manner. Experimental results demonstrate that our CNN module can accommodate all the sub-tasks without sacrificing performances, resulting in more efficient inference with smaller model size than the combinations of task-specific models.
- We also introduce a novel relation encoding module that take both position and identity information of detections into consideration for individual action and group activity classification. Empirical evaluations show that our relation encoder successfully integrates these auxiliary supervisions and produces superior classification performances over existing alternatives [38, 42].
- For extensive evaluation, we create a full annotation of the Collective dataset [13], which includes bounding boxes and track IDs of actors in all the frames of each sequence, in addition to their individual action label and their group activity class. We make this annotation publicly available[1].

[1] https://github.com/starashima/cd_full

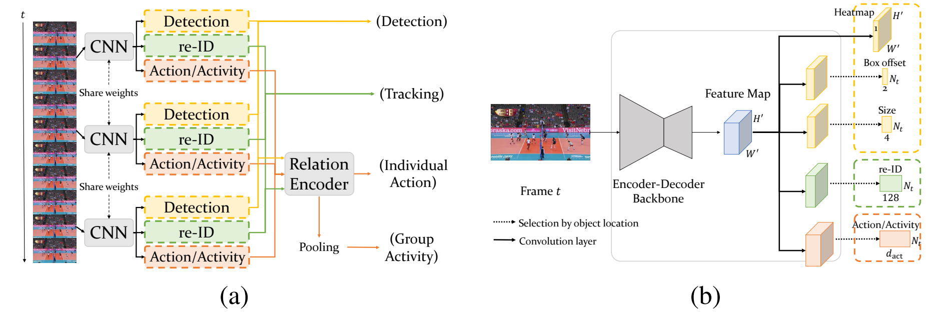


Figure 2: TrAct-Net architecture (a) and its multi-branch CNN (b). (H', W') is feature map size, N_t is the number of detections in frame t and d_{act} is action/activity embedding size.

2 Related Works

2.1 Detection and Tracking

Recent object detection and multi-object tracking (MOT) approaches have both been dominated by deep neural networks (DNNs). In the tracking-by-detection paradigm for MOT, usually each frame in an input sequence is first fed into a detector to produce target hypotheses, then they are linked between frames based on their identity [9, 28, 46] and/or positional features through data association.

While in classical methods [5, 8, 43] detection and tracking are performed separately by task-specific networks, some recent approaches [39, 41, 48] can jointly yield detections and identity embeddings for tracking within a single model. In this work we address the problem of sharing computation among sub-tasks in multi-person activity recognition via extending one of the joint detection and identity embedding approach. Specifically, we refer to FairMOT [48] since it is based on an anchor-free object detector [49], which is shown to learn detection and re-ID tasks more fairly than alternative architectures [39, 41]. We empirically validate that it is also the case in our problem setting, *i.e.*, the anchor-free approach achieves superior performances to anchor-based ones on multi-person activity recognition sub-tasks. Due to space limitation, we present this result in our supplemental material.

2.2 Individual Action and Group Activity Recognition

Classical group activity recognition studies relied on hand-crafted features extracted from each actor, which were usually fed into probabilistic graphical models to take their relation/interaction into consideration [1, 11, 12, 14, 19, 25, 26]. With the rise of deep learning, actor feature extraction has been jointly optimized with more modern relational modeling techniques such as recurrent neural networks [4, 15, 21, 22, 33, 35, 40], CNNs [2, 47], graph neural networks [7, 16, 44] and Transformers [18, 27, 30, 45]. Since most of the above works assume detection or tracking results are given by external models or systems, they neither care about the efficiency/simplicity of the whole architecture including detection and tracking nor the interdependency of sub-tasks in multi-person activity recognition.

Note that there are a few exceptions in the literature, which try to solve group activity recognition in conjunction with detection or tracking. For example, about a decade ago Choi *et al.* [11] proposed a unified framework of tracklet association and group activity recognition. However, they assume tracklets are given by external trackers and feature extraction is

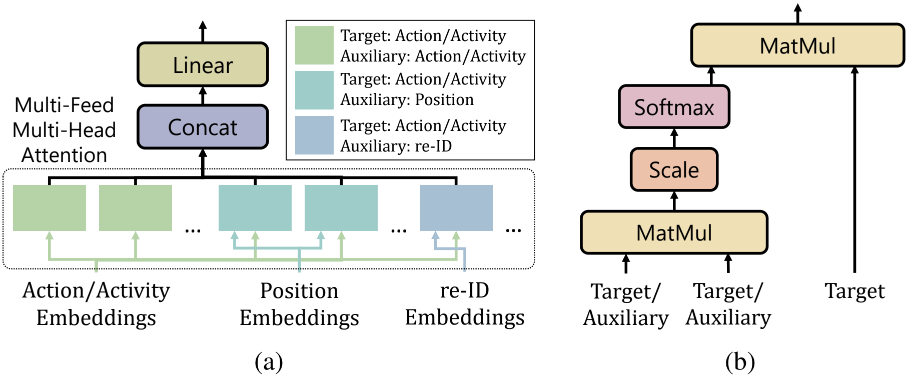

Linear
Concat
Multi-Feed Multi-Head Attention
Target: Action/Activity Auxiliary: Action/Activity
Target: Action/Activity Auxiliary: Position
Target: Action/Activity Auxiliary: re-ID
...
Action/Activity Embeddings
Position Embeddings
re-ID Embeddings
(a)
MatMul
Softmax
Scale
MatMul
Target/ Auxiliary
Target/ Auxiliary
Target
(b)

Figure 3: (a) The relation encoder has multi-feed multi-head attention mechanism, where each head transforms action/activity embeddings using any one of modalities including action/activity, position and re-ID. Each colored rectangle box in (a) represents an attention head, which is detailed in (b).

not jointly optimized with remaining modules. Besides, Bagautdinov *et al.* [4] proposed a DNN-based unified model, which jointly detect actors in a sequence while classifying their individual actions and their collective activity. However, since it does not include any re-ID feature extractor, neither can it track actor's appearance without any external model nor exploit identity supervision in action/activity classifiers.

3 TrAct-Net

Figure 2 (a) illustrates the whole architecture of our TrAct-Net, which consists of a multi-branch CNN and a relation encoder. The multi-branch CNN takes each frame in a given sequence to produce a set of embeddings for detection, re-ID for tracking and action/activity recognition. While detection and re-ID embeddings are directly used for corresponding subtasks, all the embeddings from each branch are gathered in the sequence then fed into the relation encoder, which yields features for individual action and group activity classification.

3.1 Multi-Branch CNN

Our CNN module is shown in Figure 2 (b), which includes an encoder-decoder backbone to produce a feature map of a given frame, and three branches that are responsible for detection, re-ID for tracking and action/activity classification, respectively.

3.1.1 Detection Branch

We build this branch by adopting an anchor-free object detector proposed in [48, 49]. In this approach, object centers are detected as peaks in a heatmap produced by applying several convolutions to the backbone output. The other two heads, which are responsible for box offset and size, produce equal-sized maps, and their embeddings from peak positions in the heatmap are aggregated to recover actor bounding boxes. We follow [48] to train all the modules in this branch, calling the training loss as L_{det} hereafter. Also, 4-dimensional embeddings that represent bounding boxes are fed into the relation encoder after normalization.

3.1.2 Re-identification (re-ID) Branch

This branch aims to generate embeddings for distinguishing different actors. Again, we follow [48] to build this branch, where the 128-dimensional embedding of an actor centered at (x,y) is extracted from the same location in a produced re-ID embedding map, and they are directly used for loss computation and relation modeling.

We train this branch using the triplet margin loss with semi-hard negative mining [29], which we call L_{reid} in the following sections. Notice that in the Volleyball dataset used in our experiments [22, 32] actor IDs are annotated to be unique *only* within a sequence, *i.e.*, different training sequences may include the same actors, but the consistency of their identities are not taken into consideration (*cf.* §4.1). To address this issue, we compute triplet losses *sequence-wise*, *i.e.*, triplets are sampled from frames within the same sequence, and computed sequence-wise losses are averaged through the batch. We experimentally analyzed this trick using the Collective dataset [13] with a full annotation created by us, and found the influence for performance is minimum. We show this result in our supplementary material.

3.1.3 Action/Activity Branch

This branch yields embeddings for individual action and group activity classification. We apply a 3 × 3 convolutional layer to the backbone output, followed by a ReLU and a 1 × 1 convolution to produce the embedding map, then extract d_{act}-dimensional embeddings for detected actors as is the same with §3.1.2. d_{act} is a parameter, which will be tuned in §4.4. All the extracted embeddings in a sequence are fed into our relation encoder, which is detailed in the next subsection.

3.2 Relation Encoder (RE)

Modeling relation between actors is crucial for both individual action and group activity recognition. Existing approaches use position and appearance information for the modeling, but do not fully take advantage of identical consistency between detections (*cf.* §2.2). To establish better relation modeling for multi-person activity recognition, here we propose a novel relation encoder (RE) module illustrated in Figure 3, which jointly take appearance, position and identity information into consideration. This RE can be seen as a natural extension of multi-head attention [38] to a multi-feed setting: It feeds action/activity embeddings, position embeddings and re-ID embeddings produced by the multi-branch CNN, then transforms the action/activity embeddings through a supervision of every embedding type.

3.2.1 Attention Head

An attention head, illustrated in Figure 3 (b), is very similar to the scaled-dot self-attention [38]. Suppose we have two feature sets $\mathbf{X}_{\text{tgt}} \in \mathbb{R}^{N \times d_{\text{tgt}}}$ and $\mathbf{X}_{\text{aux}} \in \mathbb{R}^{N \times d_{\text{aux}}}$, where $\mathbf{X}_{\text{tgt}}$ is a set of d_{tgt}-dimensional target embeddings to be transformed and $\mathbf{X}_{\text{aux}}$ is a set of d_{aux}-dimensional auxiliary embeddings to transform $\mathbf{X}_{\text{tgt}}$. N is the number of embeddings. The transformed d_{hid}-dimensional embeddings $h_{\text{tgt,aux}} \in \mathbb{R}^{N \times d_{\text{hid}}}$ are computed as follows:

$$h_{\text{tgt,aux}} = softmax(\frac{(\mathbf{X}_{\text{aux}}\mathbf{W}^{Q}_{\text{aux}})(\mathbf{X}_{\text{aux}}\mathbf{W}^{K}_{\text{aux}})^{T}}{\sqrt{d_{\text{hid}}}})\,(\mathbf{X}_{\text{tgt}}\mathbf{W}^{V}_{\text{tgt}}), \tag{1}$$

where $\mathbf{W}^Q_{\text{aux}} \in \mathbb{R}^{d_{\text{aux}} \times d_{\text{hid}}}$, $\mathbf{W}^K_{\text{aux}} \in \mathbb{R}^{d_{\text{aux}} \times d_{\text{hid}}}$ and $\mathbf{W}^V_{\text{tgt}} \in \mathbb{R}^{d_{\text{tgt}} \times d_{\text{hid}}}$ are linear projections with learnable parameters. When $\mathbf{X}_{\text{tgt}}$ is equal to $\mathbf{X}_{\text{aux}}$, Equation 1 is identical to the scaled-dot self-attention itself. In our RE, $\mathbf{X}_{\text{tgt}}$ is always a set of action/activity embeddings while every embedding type can be $\mathbf{X}_{\text{aux}}$.

3.2.2 Multi-Feed Multi-Head Attention

[38] shows that using multiple attention heads in parallel is beneficial for relation modeling. Here we propose a multi-feed multi-head attention, which extends the idea of multi-head attention to our problem setting. As illustrated in Figure 3 (a), the multi-feed multi-head attention consists of multiple attention heads defined in §3.2.1. Each attention head feeds action/activity embeddings $\mathbf{X}_{\text{act}} \in \mathbb{R}^{N \times d_{\text{act}}}$ as targets to be transformed, and any of action/activity embeddings, positional embeddings $\mathbf{X}_{\text{pos}} \in \mathbb{R}^{N \times 4}$ and re-ID embeddings $\mathbf{X}_{\text{reid}} \in \mathbb{R}^{N \times 128}$ as auxiliaries to transform the targets. Notice that N is the number of detections from an input sequence, and in each attention head d_{hid} is set to d_{act}/n, where n is the total number of heads in the multi-feed multi-head attention. All the heads perform attention operation in parallel, and the outputs are concatenated followed by an additional linear projection. Formally, these operations can be described as:

$$\phi(\mathbf{X}_{\text{act}}, \mathbf{X}_{\text{pos}}, \mathbf{X}_{\text{reid}}) = concat(h^1_{\text{act,act}}, h^2_{\text{act,act}}, \ldots, h^1_{\text{act,pos}}, h^2_{\text{act,pos}}, \ldots, h^1_{\text{act,reid}}, \ldots)\mathbf{W}^O, \tag{2}$$

where $\mathbf{W}^O \in \mathbb{R}^{d_{\text{act}} \times d_{\text{act}}}$ is a linear projection with learnable parameters.

Following [18, 38], in RE we apply dropout [34] with a rate of 0.1 to the result of Equation 2 before it is added to the target, then perform normalization [1] to produce the output action/activity features $\tilde{\mathbf{X}}_{\text{act}} \in \mathbb{R}^{N \times d_{\text{act}}}$. Formally, $\tilde{\mathbf{X}}_{\text{act}}$ is computed as follows:

$$\tilde{\mathbf{X}}_{\text{act}} = LayerNorm(\mathbf{X}_{\text{act}} + Dropout(\phi(\mathbf{X}_{\text{act}}, \mathbf{X}_{\text{pos}}, \mathbf{X}_{\text{reid}}))). \tag{3}$$

Notice that this RE module can easily be stacked to perform deeper relation modeling. The number of RE layers along with the number of heads in a multi-feed multi-head attention are hyperparameters, which will be tuned in our experiments (*cf.* §4.4).

Finally, resulting features are fed into a linear classifier for individual action recognition, while the max-pooled feature is passed through another classifier for group activity recognition. Cross-entropy losses are used to train them, denoted as L_{actn} and L_{acty}, respectively.

3.2.3 Discussion

Here we compare our RE with potential alternatives in the literature. Briefly, RE transforms action/activity embeddings through supervisions of appearance, position and identity information in a unified manner. From the viewpoint of embedding position information, positional encoding (PE) [38] seems to be an alternative of RE. However, we empirically found that PE doesn't work well in the multi-person activity recognition problem (*cf.* §4.4). Cross-Attention (CA) [42] is also a relevant approach to RE, in which self-attention is applied to stacked features of different modalities. We note that one important difference between RE to CA is its asymmetric nature: While CA unavoidably transforms all the modalities, RE can selectively perform transformation only to target modalities (*i.e.* action/activity embedding in our problem), which can save some computational burden and can empirically achieve better action/activity recognition performances (*cf.* §4.4).

Table 1: Results for our relation encoder (RE) evaluation. n_{act}, n_{pos}, n_{reid} are the number of attention heads in which action/activity, positional and re-ID embeddings are used as auxiliary supervisions, respectively. Notice that detection/tracking results are almost the same in all the settings.

(a)

Method			Action	Activity
	n_{act}	n_{pos}	mAP	Accuracy
PE [38]	-	-	44.8	93.2
RE (Ours) $d_{\mathrm{act}} = 256$	1	0	45.0	94.4
	0	1	45.3	94.5
	2	0	45.7	94.5
	0	2	45.4	**94.6**
	1	1	45.3	94.2
	4	0	45.7	94.5
	0	4	45.3	94.5
	3	1	45.7	94.2
	1	3	45.3	94.5
	2	2	**46.0**	**94.6**

(b)

				Action	Activity
n_{act}	n_{pos}	n_{reid}	d_{act}	mAP	Accuracy
2	2	0	256	46.0	94.6
2	2	1	320	**46.5**	**95.0**
2	2	2	384	**46.5**	94.8

(c)

Method			Action	Activity
	d_{act}	# stacks	mAP	Accuracy
CA [42]	320	1	45.8	94.6
RE (Ours)	160	1	45.6	94.2
	320	1	**46.5**	**95.0**
	320	2	46.2	94.8
	640	1	**46.5**	94.8

3.3 Training

To train the whole model, we adopt the uncertainty loss [23] to optimize the balance of individual losses in addition to the model parameters. Specifically, we define the total loss $L_{\mathrm{total}} = e^{-w_1} L_{\mathrm{det}} + e^{-w_2} L_{\mathrm{reid}} + e^{-w_3} L_{\mathrm{actn}} + e^{-w_4} L_{\mathrm{acty}} + \sum_{i=1}^{4} w_i$, where ws are all learnable.

3.4 Inference

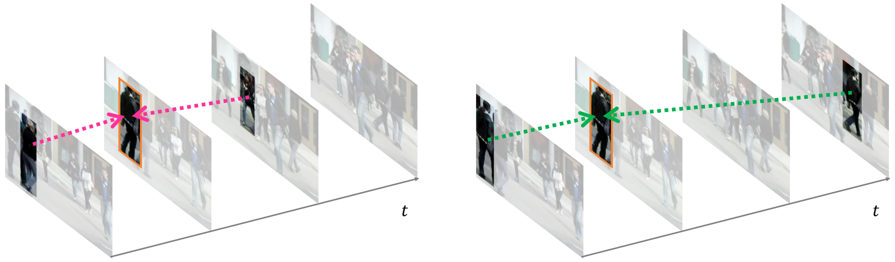
Given a sequence, all the included frames are first fed into the multi-branch CNN to produce embedding maps from every branch. Peaks are found from each detection embedding using a pre-defined threshold ρ, then corresponding re-ID and action/activity embeddings are extracted from the remaining embedding maps. Embeddings from every branch are gathered in a sequence then fed into the relation encoding module, followed by individual action and group activity classifications. In parallel, Re-ID embeddings are used to associate the same identity between frames. We adopt a standard data association algorithm [8, 48] with slight modifications, so as not to interpolate missing detections. Notice that individual action labels from the relation encoder may not be consistent through the same identities. We ensure this consistency by averaging action classifier outputs through each tracking result.

4 Evaluation

4.1 Dataset

The Volleyball Dataset[2] [22], composed of 3493 training and 1337 testing sequences, is gathered from 55 volleyball games. Following [45], we merged "*-pass" and "*-set" labels of the original dataset into "*-pass-set", resulting in 6 group activity labels and 9 individual action labels, respectively. We use annotations provided by [32] to get bounding boxes, track IDs and their action labels for every frame.

[2] https://github.com/mostafa-saad/deep-activity-rec



(a) Auxiliary: Position

(b) Auxiliary: re-ID

Figure 4: Visualization of attention results in different heads, showing top 2 attentive detections for an exemplar (orange). (a) If attention is computed by positional embeddings, spatially coherent detections tend to have higher attention weights. (b) Meanwhile, if re-ID embeddings are used, detections capturing the same instance tend to have higher weights even if they are spatially far apart.

The Collective Dataset[3] [13] consists of 44 short video clips, which is split into 1746 training and 765 testing sequences. Following [17, 40], we merge *Walking* and *Crossing* classes as a *Moving* class since their difference is ambiguous. A group activity label is defined as an individual action in which most people participate in the keyframe. Notice that original annotations are provided only to keyframes in each sequence. To fully evaluate our approach, we annotated actors' bounding boxes and their track ID in every frame of all the sequences.

4.2 Implementation Details

We adopted the enhanced DLA [49] as a CNN backbone. We initialized the parameters of TrAct-Net using a pretrained model [48], then finetuned it with the Adam optimizer [24] for 30 epochs. We started with a learning rate to $1e^{-4}$ for the Volleyball dataset and $1e^{-5}$ for the Collective dataset, decaying it by 0.1 at 20th epoch. The training batch size is set to 12. The input frame size is 1280×736 for the Volleyball dataset, and 800×400 for the Collective dataset, respectively. The detection threshold ρ for inference is set to 0.2. In all the experiments we used four A100 GPUs for training and a single A100 GPU for inference.

4.3 Evaluation Protocols

We use average precision (AP) for object detection, IDF1 and MOTA for for tracking, mean average precision (mAP) for action classification and accuracy for group activity classification. Notice that for action classification we evaluate action labels of each detection individually (*i.e.* we do not consider identity in evaluation).

4.4 Evaluation of Relation Encoder (RE)

Here we evaluate the relation encoder (RE) introduced in §3.2. In RE the numbers of attention heads n, embedding dimension d_{act} and the number of RE stacks should be tuned, and n is further broken down into $n_{\mathrm{act}} + n_{\mathrm{pos}} + n_{\mathrm{reid}}$, where n_{act}, n_{pos} and n_{reid} are the number of attention heads in which auxiliary embeddings correspond to action/activity embeddings, positional embeddings and re-ID embeddings, respectively. In Table 1 (a), we first tuned n_{act}

[3] http://vhosts.eecs.umich.edu/vision//activity-dataset.html

Table 2: Comparison on the Volleyball dataset [22]. Notice that in multi-shot results without re-ID we perform tracking without appearance [6].

					Det	Tracking		Action	Activity	Inference
Shot	Det	re-ID	Actn/Acty	Size	AP	IDF1	MOTA	mAP	Accuracy	Sec.
Multi	FRCNN [31]	-	ARG [44]	66.3M	93.7	94.8	91.1	43.4	91.8	1.7
	FRCNN	ABD [9]	ARG	73.4M	93.7	95.8	91.6	43.4	91.8	1.8
	CenterNet [49]	-	ARG	46.8M	93.4	94.7	90.4	43.8	92.0	1.3
	CenterNet	ABD	ARG	53.9M	93.4	95.8	91.5	43.8	92.0	1.6
	FairMOT [48]	FairMOT	ARG	46.3M	93.5	95.7	91.2	43.9	92.1	1.4
Single	SSU [4]	-	SSU	27.4M	66.8	-	-	28.5	83.6	0.5
	TrAct-Net	TrAct-Net	TrAct-Net	20.7M	93.4	95.8	91.4	46.5	95.0	0.8

Table 3: Comparison on the Collective dataset [13]. Notice that in multi-shot results without re-ID we perform tracking without appearance [6]. SSU [4] cannot be applicable to this dataset since the number of actors varies between sequences.

					Det	Tracking		Action	Activity	Inference
Shot	Det	re-ID	Actn/Acty	Size	AP	IDF1	MOTA	mAP	Accuracy	Sec.
Multi	FRCNN [31]	-	ARG [44]	66.3M	98.8	90.8	80.4	45.7	90.3	0.7
	FRCNN	ABD [9]	ARG	73.4M	98.8	91.5	80.9	45.8	90.3	0.8
	CenterNet [49]	-	ARG	46.8M	98.7	90.7	80.4	45.6	90.5	0.5
	CenterNet	ABD	ARG	53.9M	98.7	91.5	81.0	45.8	90.5	0.7
	FairMOT [48]	FairMOT	ARG	46.3M	98.8	91.4	81.1	45.9	90.5	0.6
Single	TrAct-Net	TrAct-Net	TrAct-Net	20.7M	98.6	91.3	81.1	47.0	91.9	0.3

and n_{pos} then performed comparison to positional encoding (PE) [38] with respect to individual action and group activity classification. We can see the best performance is achieved when $n_{\text{act}} = n_{\text{pos}} = 2$, and it outperforms PE by a meaningful margin. These results indicate that RE can successfully incorporate positional information for action/activity classification. Next, we tuned n_{reid} by inheriting the best setting in Table 1 (a). The result shown in Table 1 (b) demonstrates that performance is further boosted by introducing identity supervision. The best performance comes from $n_{\text{reid}} = 1$, which seems to be a meaningful number since the role of identity supervision is to transform each embedding using others of the same identity. In fact, as illustrated in Figure 4, the attention head supervised by re-ID embedding tries to reconstruct a target with the same actor in different frames, even though their spatial positions are far apart. Finally, we tuned d_{act} and the number of RE stacks, then compared it to Cross-Attention (CA) [42]. The results are shown in Table 1 (c). We can see both action and group activity recognition achieve best results when $d_{\text{act}} = 320$, and they are both better than CA. From these results, we can say RE successfully exploit appearance, position and identity information to perform better relation modeling than existing methods.

4.5 Performance and Efficiency of TrAct-Net

As discussed in §2.1, there is no existing method that can directly be compared to TrAct-Net in the literature. To perform holistic evaluation, we employed state-of-the-art methods for each sub-task and combined them to solve the whole multi-person activity recognition problem. Specifically we chose FRCNN [31] and CenterNet [49] for detection, ABD [9] for re-ID, ARG [44] for individual action and group activity recognition and FairMOT [48] for joint detection and re-ID. We also employed SSU [4], which integrates detection and action/activity recognition into a single model. We deployed all the methods using publicly available codes[4][5][6][7], finetuning all the models using target datasets.

[4] https://github.com/ifzhang/FairMOT

[5] https://github.com/wjchaoGit/Group-Activity-Recognition

[6] https://github.com/cvlab-epfl/social-scene-understanding

[7] https://github.com/VITA-Group/ABD-Net

Table 4: Single shot vs. multi shot on the Volleyball [22] and the Collective [13] datasets.

			Detection	Tracking		Action	Activity	Inference
Dataset	Shot	Size	AP	IDF1	MOTA	mAP	Accuracy	Sec.
Collective [13]	Multi	41.4M	98.8	91.4	81.1	46.2	91.0	0.6
	Single	20.6M	98.6	91.3	81.1	47.0	91.9	0.3
Volleyball [22]	Multi	41.5M	93.5	95.8	91.2	46.2	94.6	1.5
	Single	20.7M	93.4	95.8	91.4	46.5	95.0	0.8

Table 2 shows the results on the Volleyball dataset. First, compared to SSU [4], TrAct-Net achieves much better performances for every sub-task, while having smaller model size. Also TrAct-Net achieves higher individual action and group activity classification performances than ARG [44], which is one of the state-of-the-art methods. Further, we can see re-ID embedding from TrAct-Net help improve tracking performance, and it is competitive to a task-specific existing method [9]. Needless to say, TrAct-Net inference is much faster than multi-shot methods. We also performed comparison on the Collective dataset, and obtain consistent results with the above (*cf.* Table 3). These results indicate the superiority of TrAct-Net to combinations of existing methods.

Next, to evaluate the influence of parameter sharing among sub-tasks, we made a multi-shot variant of TrAct-Net and compare the results with the single-shot one. To make multi-shot TrAct-Net, we defined two similar models which share the same backbone with TrAct-Net but one of which is only responsible for detection and tracking while the other for individual action and group activity recognition. In this case the relation encoder (RE) is attached only to the action/activity classifier, and re-ID embeddings are not fed into the RE. The results are shown in Table 4. For detection and tracking, we can see their performances are almost the same between methods in both the Collective and the Volleyball datasets. Interestingly, for individual action and group activity recognition, the single shot TrAct-Net is superior to its multi-shot variant even though the parameter is much fewer. One possible reason is the existence of identity information: Identity embedding, which is only exploitable for the single-shot model, helps action/activity classification even on the smaller parameter setting. Also, we can see the above effect is slightly higher on the Collective dataset [13] (*i.e* group activity recognition improves from 91.0 to 91.9 in the Collective while from 94.6 to 95.0 in the Volleyball). This may be due to the different frame rates between datasets: Since in the Collective frame rates are smaller thus actors tend to move farther between frames, identity supervision helps more to improve action/activity recognition.

Based on these observations, we say our TrAct-Net achieves superior multi-person activity recognition performance to combinations state-of-the-art methods, with much faster inference and smaller model size.

5 Conclusion

In this paper we proposed TrAct-Net, a one-shot deep model to jointly solve detection, tracking, individual action recognition and group activity recognition in a unified architecture. Experimental comparison on public benchmarks demonstrated TrAct-Net's higher performance for this multi-person activity recognition problem, while inference is much faster and model size get halved.

For future research, we will explore to apply TrAct-Net to more challenging scenarios, including action/activity prediction and weakly-supervised learning.

References

[1] M. R. Amer, P. Lei, and S. Todorovic. Hirf: Hierarchical random field for collective activity recognition in videos. In *ECCV*, 2014.

[2] S. Mokhtarzadeh Azar, M. G. Atigh, A. Nickabadi, and A. Alahi. Convolutional relational machine for group activity recognition. In *CVPR*, 2019.

[3] J. L. Ba, J. R. Kiros, and G. E Hinton. Layer normalization. *arXiv preprint arxiv:1607.06450*, 2016.

[4] T. Bagautdinov, A. Alahi, F. Fleuret, P. Fua, and S. Savarese. Social scene understanding: End-to-end multi-person action localization and collective activity recognition. In *CVPR*, 2017.

[5] P. Bergmann, T. Meinhardt, and L. Leal-Taixé. Tracking without bells and whistles. In *ICCV*, 2019.

[6] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *ICIP*, 2016.

[7] J. Chen, W. Bao, and Y. Kong. Group activity prediction with sequential relational anticipation model. In *ECCV*, 2020.

[8] L. Chen, H. Ai, Z. Zhuang, and C. Shang. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *ICME*, 2018.

[9] T. Chen, S. Ding, J. Xie, Y. Yuan, W. Chen, Y. Yang, Z. Ren, and Z. Wang. Abd-net: Attentive but diverse person re-identification. In *ICCV*, 2019.

[10] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *CVPR*, 2020.

[11] W. Choi and S. Savarese. A unied framework for multi-target tracking and collective activity recognition. In *ECCV*, 2012.

[12] W. Choi and S. Savarese. Understanding collective activities of people from videos. *TPAMI*, 2014.

[13] W. Choi, K. Shahid, and S. Savarese. What are they doing? : Collective activity classification using spatio-temporal relationship among people. In *ICCV Workshops*, 2009.

[14] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *CVPR*, 2011.

[15] Z. Deng, A. Vahdat, H. Hu, and G. Mori. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In *CVPR*, 2016.

[16] M. Ehsanpour, A. Abedin, F. Saleh, J. Shi, I. Reid, and H. Rezatofighi. Joint learning of social groups, individuals action and sub-group activities in videos. In *ECCV*, 2020.

[17] Progressive Relation Learning for Group Activity Recognition. G. hu and b. cui and y. he and s. yu. In *CVPR*, 2020.

[18] K. Gavrilyuk, R. Sanford, M. Javan, and C. G. M. Snoek. Actor-transformers for group activity recognition. In *CVPR*, 2020.

[19] H. Hajimirsadeghi, W. Yan, A. Vahdat, and G. Mori. Visual recognition by counting instances: A multi-instance cardinality potential kernel. In *CVPR*, 2015.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[21] M. S. Ibrahim and G. Mori. Hierarchical relational networks for group activity recognition and retrieval. In *ECCV*, 2018.

[22] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori. A hierarchical deep temporal model for group activity recognition. In *CVPR*, 2016.

[23] A. Kendall, Yarin Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018.

[24] D. P. Kingma and J. L. Ba. Adam: a method for stochastic optimization. In *ICLR*, 2015.

[25] T. Lan, Y. Wang, W. Yang, S. N. Robinovitch, and G. Mori. Discriminative latent models for recognizing contextual group activities. *TPAMI*, 2011.

[26] T. Lan, L. Sigal, and G. Mori. Social roles in hierarchical models for human activity recognition. In *CVPR*, 2012.

[27] S. Li, Q. Cao, L. Liu, K. Yang, S. Liu, J. Hou, and S. Yi. Groupformer: Group activity recognition with clustered spatial-temporal transformer. In *ICCV*, 2021.

[28] H. Luo, Y. Gu, X. Liao, S. Lai, and W. Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *CVPR Workshops*, 2019.

[29] K. Musgrave, S. Belongie, and S.-N. Lim. A metric learning reality check. In *ECCV*, 2020.

[30] R. R. A. Pramono, Y. T. Chen, and W. H. Fang. Empowering relational network by self-attention augmented conditional random fields for group activity recognition. In *ECCV*, 2020.

[31] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[32] K. Sendo and N. Ukita. Heatmapping of people involved in group activities. In *MVA*, 2019.

[33] T. Shu, S. Todorovic, and S.-C. Zhu. Cern: Confidence-energy recurrent network for group activity recognition. In *CVPR*, 2017.

[34] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014.

[35] stagNet: An Attentive Semantic RNN for Group Activity Recognition. M. qi and j. qin and a. li and y. wang and j. luo and l. van gool. In *ECCV*, 2018.

[36] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017.

[39] P. Voigtlaender, M. Krause, A. Ŏsep, and J. Luiten. Mots: Multi-object tracking and segmentation. In *CVPR*, 2019.

[40] M. Wang, B. Ni, and X. Yang. Recurrent modeling of interaction context for collective activity recognition. In *CVPR*, 2017.

[41] Z. Wang, L. Zheng, Y. Liu, and S. Wang. Towards real-time multi-object tracking. *arXiv preprint arxiv:1909.12605*, 2019.

[42] X. Wei, T. Zhang, Y. Li, Y. Zhang, and F. Wu. Multi-modality cross attention network for image and sentence matching. In *CVPR*, 2020.

[43] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017.

[44] J. Wu, L. Wang, L. Wang, J. Guo, and G. Wu. Learning actor relation graphs for group activity recognition. In *CVPR*, 2019.

[45] R. Yan, L. Xie, J. Tang, X. Shu, and Q. Tian. Social adaptive module for weakly-supervised group activity recognition. In *ECCV*, 2020.

[46] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. H. Hoi. Deep learning for person re-identification: A survey and outlook. *TPAMI*, 2021.

[47] H. Yuan, D. Ni, and M. Wang. Spatio-temporal dynamic inference network for group activity recognition. In *ICCV*, 2021.

[48] Y. Zhang, C. Wang, X. Wangy, W. Zeng, and W. Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *arXiv preprint arXiv:2004.01888*, 2020.

[49] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.