Hierarchical Deep Network for Group Discovery and Multi-level Activity Recognition

Ashish Goyal
Indian Institute of Technology, Bombay
Mumbai, Maharashtra
goyal26@outlook.com

Neha Bhargava
Indian Institute of Technology, Bombay
Mumbai, Maharashtra
neha.iitb@gmail.com

Subhasis Chaudhuri
Indian Institute of Technology, Bombay
Mumbai, Maharashtra
sc@ee.iitb.ac.in

Rajbabu Velmurugan
Indian Institute of Technology, Bombay
Mumbai, Maharashtra
rajbabu@ee.iitb.ac.in

ABSTRACT

We present a deep network based hierarchical framework to recognize activities at various levels of granularity - individual, group and overall (or scene level). Most of the existing work focus on scene activity and ignore any intermediate analysis. In this work, we extend the existing methods by adding an extra layer that finds the groups (or clusters) present in a scene and their activities. We then utilize these group activities along with the scene context to recognize the scene activity. To discover these groups, we propose a min-max criteria within the framework to learn pairwise similarity between any two individuals, which is used by a clustering algorithm. The group activity is captured by an LSTM module whereas the individual and scene activities are captured by CNN-LSTM based modules. These modules along with the grouping layer form the proposed network. We evaluate the network on publicly available dataset to indicate the usefulness of our approach.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Machine learning**.

ACM Reference Format:
Ashish Goyal, Neha Bhargava, Subhasis Chaudhuri, and Rajbabu Velmurugan. 2018. Hierarchical Deep Network for Group Discovery and Multi-level Activity Recognition. In *11th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP 2018), December 18–22, 2018, Hyderabad, India.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3293353.3293390

1 INTRODUCTION

An activity video can be understood at different levels of granularity that can be related to a hierarchical representation. This hierarchy comes naturally in crowd videos since people tend to interact with each other and form different groups. These groups collectively influence the scene activity. Therefore, a video can be represented as a hierarchical graph where leaf nodes correspond to the individuals, the root node corresponds to the scene, and the intermediate layer is comprised of various groups. The suitable activities can be assigned at these different levels in the hierarchy - *action* to an individual, *group activity* to a group and *scene activity* to the scene. The spatio-temporal interaction among the individuals leads to different group activities and these group activities along with the scene context influence the scene activity. An illustration of a hierarchy in an activity video is given in Figure 1 - there are six *standing* individuals and interaction among them generates two *talking* groups in the scene. Since the major activity is *talking*, the scene activity is also *talking*.

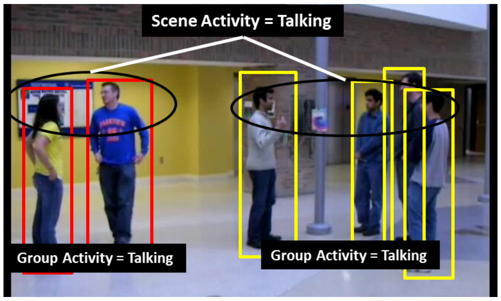


Figure 1: Illustration of hierarchy present in a video. There are 6 *standing* individuals forming two *talking* groups. The overall scene activity is *talking*.

The groups play an important role in activity analysis. Together, the groups influence the overall activity of the crowd. Individually, a group guides the actions of its members. Therefore, group discovery and group level analysis are important in activity recognition. Once the groups are known, the latent hierarchical structure in the video becomes identifiable. We propose a model that discovers groups as well as recognizes activities at different levels of granularity. The groups are discovered using a clustering algorithm that takes similarity matrix as its input. We learn a pairwise similarity measure from a fully connected neural network. The objective function to learn this measure is based on maximizing the

ICVGIP 2018, December 18–22, 2018, Hyderabad, India
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6615-1/18/12.
https://doi.org/10.1145/3293353.3293390

intra-group similarity score and minimizing the inter-group similarity score. To recognize a group activity, our framework utilizes group level features that depend upon features of each member of the group. Along with the scene context, these group activities are then used to identify the scene level activity.

In this paper, the term action refers to an atomic movement (*e.g. stand, walk*) of an individual, the term group activity denotes an activity performed by a group and scene activity refers to the main activity happening in the scene. The paper is organized as follows. The next section discusses the related work. The proposed method is explained in Section 3. The implementation and experimentation details are given in Section 4 and Section 5, respectively. Finally, the paper concludes in Section 6.

2 RELATED WORK

Activity recognition is an active area of research due to its various applications. There are numerous approaches present in the literature to recognize scene activity. Some of the recent and interesting works are [2–4, 9, 13, 15, 16, 19, 20]. In [3], Choi *et al.* recognize the collective activity by extracting local spatio-temporal descriptors from people and the surroundings. In [4], they extend the algorithm by automatically capturing the crowd context and use it for classification. In [2], the authors go one step further and present a unified framework for target tracking and activity recognition. Ryoo and Aggarwal in [19] model a group activity as a stochastic collection of individual activities. The method proposed in [9] is based on multi-instance cardinality model with hand crafted features.

Recently, deep learning based methods have also been explored to recognize activities [6, 7, 10, 12]. In [7], Deng *et al.* proposed a deep model to capture individual actions, pairwise interactions and group activity. The authors in [6] first estimate the individual and scene activities that are further refined using a message passing algorithm under a framework of recurrent neural network. In [12], the authors proposed a two-staged LSTM model where first stage captures individual temporal dynamics followed by scene activity recognition based on aggregated individual information.

Most of the existing approaches focus on scene activity recognition and ignore any intermediate analysis. Hence such methods are not suitable for the videos with multiple groups performing different activities. We think that this group-level information is important to understand the scene in its completeness. Such group level information can be utilized for many high-level applications such as abnormal activity detection.

Keeping these short-comings in the existing approaches, we make the following contributions in this paper:

(1) We propose a hierarchical framework to analyze a video in its entirety - from *individual action* to *group activity* to *scene activity*.
(2) Usually scene activity recognition provides only the top-level activity ignoring activities of individual groups and their contributions in the scene activity. We propose a method for group discovery and group activity recognition. Hence, we add one more layer in the existing hierarchical methods.
(3) As a minor contribution, we also present an objective function to learn the pairwise similarity measure under the proposed framework of deep network. We also present an architecture to combine the scene context with group activities to learn the scene activity.

3 PROPOSED METHOD

In this section, we discuss the proposed method in detail. The model has various stages as shown in Figure 2. The overview of the framework is as follows. The inputs to our model are bounding boxes of the individuals as well as the scene image. We first extract features for all the individuals. The feature encodes action, pose, and location of an individual. To extract these features, the bounding boxes are fed to an *individual unit* described in Section 3.1. These individual features ϕ's are utilized by a clustering algorithm to group the individuals in the scene. Once the groups are known, we use an LSTM based model to recognize group activities. To recognize the scene activity, the model utilizes group activity distribution and the scene context. In this section, we discuss the individual modules of the proposed model in detail.

3.1 Individual feature descriptor

The descriptor encodes the spatio-temporal information of the individual. To capture the appearance information, we pass the bounding boxes of an individual through a CNN. We use pretrained ResNet [11] for this CNN framework. The feature vectors thus obtained are passed through LSTM to extract the spatio-temporal descriptor for an individual. An LSTM consists of many memory cells which enable it to capture the temporal behaviour. The outputs of the LSTM are the predicted action a and pose θ of the individual. A similar CNN-LSTM pipeline has shown promising results in video classification problem [17]. The final feature vector ϕ_i for the i^{th} individual consists of pose p_i, action a_i and its bounding box $[x_i, y_i, w_i, h_i]$. These components in a feature vector are important for group detection as discussed next.

3.2 Group discovery

When individuals form a group, their features (poses, actions and locations) interact in a specific manner. For example, group members tend to have spatial closeness and similar atomic action. The pose compatibility is also desirable in a group. For example, two persons standing back to back are not likely to be in a same group but if they face each other then they are more likely to be in a group. Instead of defining the group formation rules explicitly, we learn the pairwise similarity measure by training a fully connected network. In a group, not all members necessarily interact with each other. For example, the individuals standing in a queue are considered in a single group but two individuals standing far apart from each other may not be interacting directly but via chain of other individuals. Moreover, pose and action are non-numerical data. Thus, learning such a measure directly from group information becomes non-trivial if the pairwise interactions are not known.

We define an objective function f to learn this pairwise similarity measure which tries to maximize the intra-group similarity score and minimize the inter-group score. Let $\mathcal{I} = \{1, 2, ..., n\}$ denote a set of individuals. The goal of clustering is to divide the

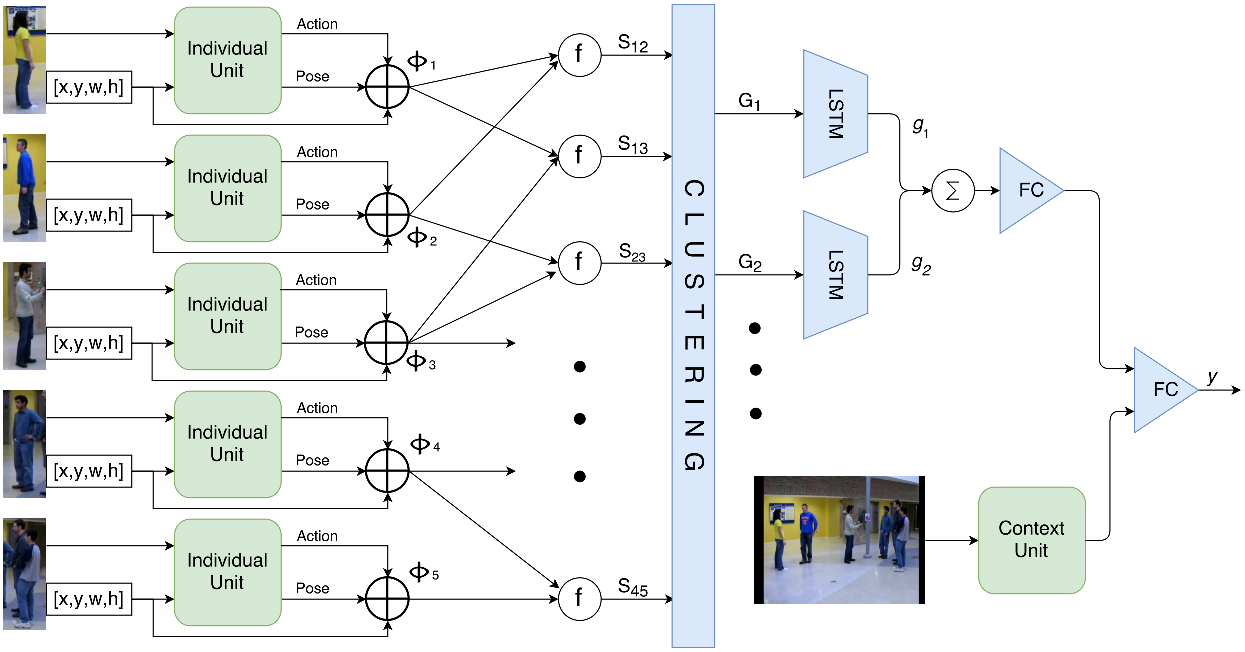


Figure 2: Hierarchical learning in deep network. First, we capture dynamics a_i, θ_i of an individual through an LSTM. This along with the bounding box information constitutes the feature vector ϕ_i for the individual which is then used for clustering. f is a pairwise similarity function and S_{ij} is the similarity score between i^{th} and j^{th} persons. After clustering, the group features are passed through an LSTM to estimate group activity g. To estimate scene activity y, we learn scene context from the *context unit* which along with group activities are used to recognize scene activity y. Note that FC stands for fully connected.

individuals in a partition $\mathcal{P}$, such that $\bigcup_{\omega_i \in \mathcal{P}} \omega_i = \mathcal{I}$ where ω_i is the i^{th} group. To do this, we learn a pairwise similarity function $f_w(\phi_i, \phi_j)$ parameterized by w, where ϕ_i is the feature vector for the i^{th} person. Let P_i denote the set of individuals interacting with i (*i.e.* they are in the same group as of i). We define two types of costs for i^{th} individual - an inclusion cost α_i and an exclusion cost β_i as follows:

$$\alpha_i = \max_{j \neq i,\ j \in P_i} f_w(\phi_i, \phi_j)$$

$$\beta_i = \max_{j \neq i,\ j \notin P_i} f_w(\phi_i, \phi_j)$$

Where α_i is the maximum similarity score of i with any other individual in P_i and β_i is the maximum similarity score of i with an individual not in P_i. It is desirable to maximize α_i and minimize β_i for the i^{th} individual. Hence, we minimize an objective function defined as $J_i = \beta_i - \alpha_i$ to learn w.

To perform clustering on the pairwise similarity matrix S where S_{ij} is a pairwise similarity score for i^{th} and j^{th} individuals, we use DBSCAN [8] method. It is a density based clustering algorithm which groups the close points. The algorithm is useful when the number of clusters is not known such as in our case. Once we have the groups, the next stage in Figure 2 is recognition of activities of the predicted groups. The method is discussed next.

3.3 Scene activity recognition

We define scene activity as the major activity happening in the scene. It is highly dependent on the group activities. Additionally, the scene context also plays an important role in determining the scene activity. For instance, recognition of *crossing* activity depends on whether a road is present in the scene or not. Hence, the scene activity can be recognized from two cues - (a) scene context, and (b) group activities. We propose a *context unit* that is used to obtain the scene context and is illustrated in Figure 3. The input to the unit is the scene. It is fed to a CNN followed by an LSTM. To avoid over-fitting of the network, we use dropout [21]. The idea is to drop some units randomly during training so that they do not co-adapt too much. The output is then passed through a fully connected layer to get a context vector of length 32. To get the second cue of group activity feature, we use the histogram of group activities (length of 4) that is passed to a fully connected network that outputs a feature vector of length 32. Both these vectors of length 32 are then combined and sent to a fully connected network that predicts the scene activity.

4 IMPLEMENTATION

Our hierarchical network is trained in multiple stages and is implemented using Keras [5]. All CNNs used in our model are initialized by ResNet-50 [11] with pruned fully-connected layers . All LSTMs use *sigmoid* function for activation and *tanh* for recurrent activation. Unless stated otherwise, we use *softmax* activation for prediction layer with cross-entropy loss. We use Adam optimizer [14] to train the networks, with a learning rate of 0.001. The details of the stages in our network are given in the following subsections.

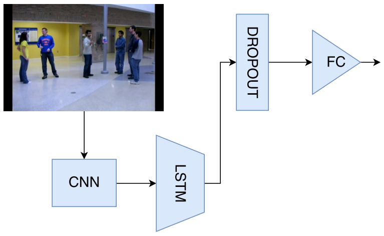


Figure 3: Context unit to capture the visual information in the scene. The scene image is passed though a CNN followed by a LSTM with a technique of dropout.

4.1 Individual unit

Each individual unit is composed of two detectors as described below:

(1) Pose detector: The poses are quantized in 8 directions. The detector is trained on Parse-27k dataset [22] using a CNN. During training, we modify the standard cross-entropy loss in the network to penalize the predictions that are deviating from the true pose by more than 90°.

(2) Action detector: The action detector is trained using ground truth trajectories along with the corresponding bounding box images (see Figure 4). The training is done in an end-to-end fashion with fixed CNN weights.

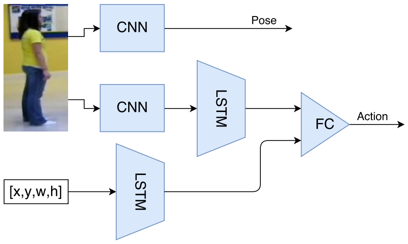


Figure 4: Individual unit for pose and action detection. To detect pose, the image is fed to a CNN. To detect action, the image is fed to a CNN followed by an LSTM with 256 hidden units. The trajectories are directly fed to an LSTM with 256 hidden units. The two outputs are concatenated and fed to a prediction layer.

4.2 Similarity measure unit

We learn our pairwise similarity measure f_w under the framework of neural networks. The network consists of two fully-connected layers having 32 neurons each and *tanh* as the activation function. The prediction layer of the network is a single neuron with sigmoid activation function to ensure that the score is always in the range $(0, 1)$.

4.3 Group and scene activity stage

As explained before, we need to fix a number K of members to be used in group activity recognition. Since the average group cardinality in the dataset is around 10, we keep $K = 10$ for the fixed number approach in group activity recognition. We use 512 hidden units in the LSTM followed by a fully connected layer of 128 neurons with *tanh* as the activation function and a prediction layer. For context unit, we keep 256 hidden units in the LSTM and dropout rate is fixed as 0.9. The subsequent fully-connected layer is comprised of 16 hidden units with *tanh* as the activation function. The output is concatenated to the output of fully-connected layer of group activity. This concatenated output is fed to the prediction layer for scene activity prediction.

5 EXPERIMENTATION

In this section, we discuss the experimental results for group detection, group activity, and scene activity. We also explain the dataset used for experimentation with the additional annotations we provide for the grouping layer. We provide results on videos in the supplementary material.

5.1 Dataset

We use Collective Activity Dataset [3] to demonstrate the performance of our model. The dataset consists of 44 videos covering different challenging videos. The authors of [3] have provided the annotations for five scene activities (*walking*, *waiting*, *queuing*, *talking* and *crossing*) after every 10 frames. Since we are interested in finding groups and group activities also, we annotate the group labels and group activities (*walking*, *waiting*, *queuing* and *talking*) after every 10 frames.

NMI	Purity	Rand Index
0.82	0.88	0.89

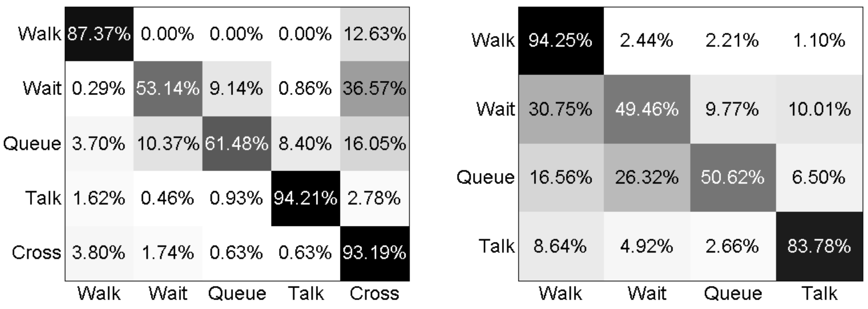
Table 1: Performance of group detection algorithm on Collective Activity dataset.

5.2 Group detection

To evaluate the group detection performance, we use the following widely used measures - (a) Normalized Mutual Information (*NMI*) [23], (b) *Purity* [1], and (c) *Rand Index* [18]. *NMI* is inspired by information theory ideas and finds the mutual information between the two clustering outputs. *Purity* is defined as the average percentage of the dominant class label in each cluster. *Rand Index* penalizes both false positive and false negative decisions during clustering. These measures take values in $[0, 1]$ where 1 indicates the perfect clustering. The values of these measures obtained using the proposed approach are shown in Table 1.

5.3 Group activity recognition

The confusion matrix for the group activity is shown in Figure 5. We get average accuracy as 80.2%. If we run the framework with true poses and true actions, we achieve 88.9% and with only true actions, we get 84.0% of accuracy. We observe confusion between *wait* and *queue*. We suspect that the network is unable to capture the relationship between relative locations and the associated poses, which is required to discriminate between *wait* and *queue*.



	Walk	Wait	Queue	Talk	Cross
Walk	87.37%	0.00%	0.00%	0.00%	12.63%
Wait	0.29%	53.14%	9.14%	0.86%	36.57%
Queue	3.70%	10.37%	61.48%	8.40%	16.05%
Talk	1.62%	0.46%	0.93%	94.21%	2.78%
Cross	3.80%	1.74%	0.63%	0.63%	93.19%

	Walk	Wait	Queue	Talk
Walk	94.25%	2.44%	2.21%	1.10%
Wait	30.75%	49.46%	9.77%	10.01%
Queue	16.56%	26.32%	50.62%	6.50%
Talk	8.64%	4.92%	2.66%	83.78%

Figure 5: Confusion matrices for the scene activity and the group activity, obtained from the proposed method.

Method	Scene activity accuracy	Group activity accuracy
Cardinality kernel [9]	83.4%	-
Deep structured model [7]	80.6%	-
Structure Inference Machine [6]	81.2%	-
Hierarchical deep temporal network [12]	81.5%	-
Proposed method (A)	80.5%	80.2%
Proposed method (B)	84.0%	82.1%
Proposed method (C)	88.9%	86.2%

Table 2: Comparison of our method with the state-of-the-art methods for scene and group activity. (A) True poses and actions of individuals are unknown, (B) Only true poses are known, (C) Both true actions and poses are known

5.4 Scene activity recognition

We compare the scene activity recognition performance with [6], [12], [7] and [9]. The average accuracies for comparison are mentioned in Table 2. The confusion matrix obtained from our method is shown in Figure 5. The Cardinality Kernel [9] although achieves the highest accuracy, but uses hand crafted features. If we run the framework with true poses and true actions, the accuracy for scene activity reaches 86.2% and with true actions alone it goes to 82.1%. The average accuracy of our method is at par with the state-of-the-art methods. Additionally, we provide activities at various levels unlike others.

Some frames from different activity videos illustrating the qualitative results of our algorithm are presented in Figure 6. It is clear from these figures that our approach is able to provide meaningful activity labels at various levels.

6 CONCLUSION

We proposed a method for hierarchical and multi-stage analysis of activity videos. We learn the temporal dynamics of the scene at various levels - *individual*, *group* and *scene*. We also estimate the hierarchical structure present in the scene by discovering the groups. Overall, we provide a novel approach to analyze a video in its entirety. The results on Collective Activity dataset are competitive with the state-of-the-art methods and at the same time provide activity information at various levels.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support provided by National Center of Excellence in Technology for Internal Securities (NCETIS) (an initiative by Indian Institute of Technology Bombay and DeitY).

REFERENCES

[1] Charu C Aggarwal. 2004. A human-computer interactive method for projected clustering. *IEEE transactions on knowledge and data engineering* 16, 4 (2004), 448–460.

[2] W. Choi and S. Savarese. 2012. A Unified Framework for Multi-Target Tracking and Collective Activity Recognition. In *ECCV*.

[3] Wongun Choi, Khuram Shahid, and Silvio Savarese. 2009. What are they doing?: Collective activity classification using spatio-temporal relationship among people. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 1282–1289.

[4] Wongun Choi, Khuram Shahid, and Silvio Savarese. 2011. Learning context for collective activity recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 3273–3280.

[5] François Chollet. 2015. Keras.

[6] Zhiwei Deng, Arash Vahdat, Hexiang Hu, and Greg Mori. 2016. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4772–4781.

[7] Zhiwei Deng, Mengyao Zhai, Lei Chen, Yuhao Liu, Srikanth Muralidharan, Mehrsan Javan Roshtkhari, and Greg Mori. 2015. Deep structured models for group activity recognition. *arXiv preprint arXiv:1506.04191* (2015).

[8] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd*, Vol. 96. 226–231.

[9] Hossein Hajimirsadeghi, Wang Yan, Arash Vahdat, and Greg Mori. 2015. Visual recognition by counting instances: A multi-instance cardinality potential kernel. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2596–2605.

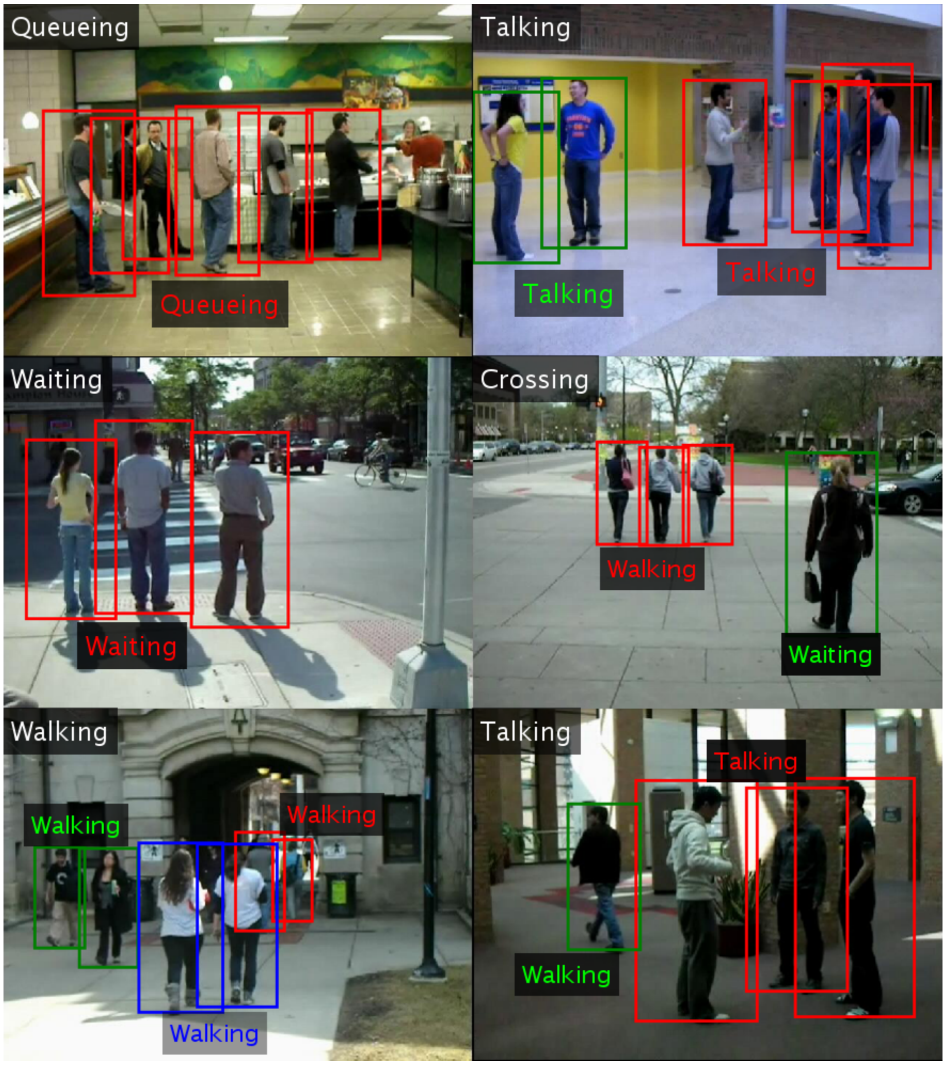


Figure 6: Demonstration of outputs of our proposed method at group and scene levels on various videos. The scene activity is mentioned at the *top-left* while the group activities are mentioned near the groups. Best viewed in color.

[10] Hossein Hajimirsadeghi, Wang Yan, Arash Vahdat, and Greg Mori. 2015. Visual recognition by counting instances: A multi-instance cardinality potential kernel. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2596–2605.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385 (2015). http://arxiv.org/abs/1512.03385

[12] Mostafa S Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori. 2016. A hierarchical deep temporal model for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1971–1980.

[13] Saad M Khan and Mubarak Shah. 2005. Detecting group activities using rigidity of formation. In *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, 403–406.

[14] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). http://arxiv.org/abs/1412.6980

[15] Tian Lan, Yang Wang, Weilong Yang, and Greg Mori. 2010. Beyond actions: Discriminative models for contextual group activities. In *Advances in neural information processing systems*. 1216–1224.

[16] Ruonan Li, Rama Chellappa, and Shaohua Kevin Zhou. 2009. Learning multimodal densities on discriminative temporal interaction manifold for group activity recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2450–2457.

[17] Joe Yue-Hei Ng, Matthew J. Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. 2015. Beyond Short Snippets: Deep Networks for Video Classification. *CoRR* abs/1503.08909 (2015). http://arxiv.org/abs/1503.08909

[18] William M Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association* 66, 336 (1971), 846–850.

[19] MS Ryoo and JK Aggarwal. 2011. Stochastic representation and recognition of high-level group activities. *International journal of computer Vision* 93, 2 (2011), 183–200.

[20] Michael S Ryoo and Jake K Aggarwal. 2009. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *Computer vision, 2009 ieee 12th international conference on*. IEEE, 1593–1600.

[21] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[22] Patrick Sudowe, Hannah Spitzer, and Bastian Leibe. 2015. Person Attribute Recognition with a Jointly-trained Holistic CNN Model. In *ICCV'15 ChaLearn Looking at People Workshop*.

[23] Mingrui Wu and Bernhard Schölkopf. 2006. A local learning approach for clustering. In *NIPS*. 1529–1536.