A Flow Model for Joint Action Recognition and Identity Maintenance

Sameh Khamis, Vlad I. Morariu, Larry S. Davis
University of Maryland, College Park
{sameh,morariu,lsd}@umiacs.umd.edu

Abstract

We propose a framework that performs action recognition and identity maintenance of multiple targets simultaneously. Instead of first establishing tracks using an appearance model and then performing action recognition, we construct a network flow-based model that links detected bounding boxes across video frames while inferring activities, thus integrating identity maintenance and action recognition. Inference in our model reduces to a constrained minimum cost flow problem, which we solve exactly and efficiently. By leveraging both appearance similarity and action transition likelihoods, our model improves on state-of-the-art results on action recognition for two datasets.

1. Introduction

We introduce a novel framework for human action recognition from videos. We are motivated by the fact that actions in a video sequence typically follow a natural order. Consider the illustration in Figure 1. The person outlined in the left image is queueing, while the person outlined in the right image is waiting to cross. Given the appearance and stance resemblance, a classifier might return similar scores for both actions. However, we can take advantage of their actions at a later time, when the person on the right will be crossing while the person on the left will still be queueing; their actions then become more distinguishable.

One issue that remains with this idea is identity maintenance. A simple approach would be to build the tracks of people detections using appearance models, and then construct an action recognition model that makes use of the identities established from the tracking step. This approach assumes that such tracks are accurate and disregards the advantage of jointly solving both problems under one framework. This is most evident with similar appearances and overlapping bounding boxes, where the likelihood of a transition between compatible actions can improve the inference of the identities.

We develop a novel representation of the joint problem. We initially train a linear SVM on the Action Context (AC) descriptor[17], which explicitly accounts for group actions to recognize an individual's action. We use the normalized classifier scores for the action likelihood potentials. We then train an appearance model for identity association. Our association potentials incorporate both appearance cues and action consistency cues. Our problem is then represented by a constrained multi-criteria objective function. Casting this problem in a network flow model allows us to perform inference efficiently and exactly. Finally, we report results that outperform state-of-the-art methods on two group action datasets.

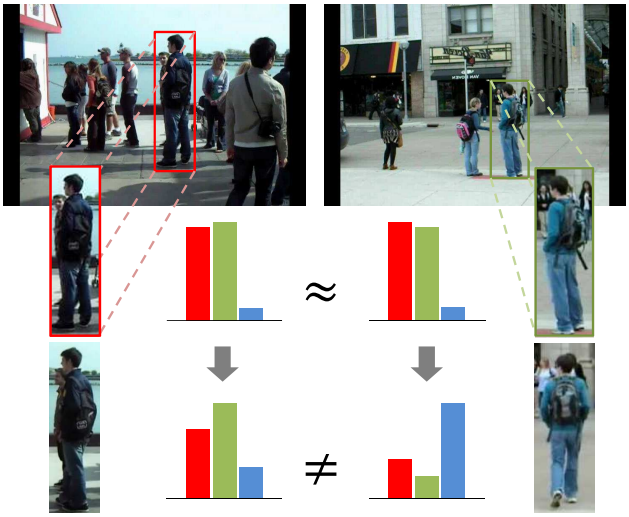

Figure 1. How tracking can improve action recognition. While the person outlined on the left is queueing and the person outlined on the right is waiting to cross, a classifier might initially return similar scores for both given the resemblance in their appearance and stance. However, the actions become more distinguishable after the person on the right is tracked to subsequent frames and is observed to be crossing. We present a framework to solve both problems jointly and efficiently.

Our contribution in this work is three-fold:

- We propose jointly solving action recognition and identity maintenance under one framework.

- We formulate inference as a flow problem and solve it exactly and efficiently.

- Our action recognition performance improves on the state-of-the-art results for two datasets.

The rest of this paper is structured as follows. In Section 2 we survey the action recognition literature and discuss our contribution in its light. We introduce our approach and focus on the problem formulation in Section 3. We then discuss the system in details in Section 4. We present the datasets in Section 5, and report our results quantitatively and qualitatively. And last, we conclude in Section 6.

2. Related Work

In recent work on action recognition, researchers have explicitly modeled interactions amongst actions being observed, jointly solving multiple previously independent vision problems. Such interactions include those between scenes and actions (e.g., *road* and *driving*) [19], objects and actions [14, 30] (e.g., *spray bottle* and *spraying*, *tennis raquet* and *swinging*) or actions performed by two or more people [6, 18, 17, 7] (e.g., two people standing versus two people queueing). More complex high level interactions have also been modeled, e.g., by dynamic Bayesian networks (DBNs) [29], CASE natural language representations [16], Context-Free Grammars (CFGs) [22], AND-OR graphs [15], and probabilistic first-order logic [20, 5].

To reason about actions over time, most of these approaches require that people or objects are already detected and tracked [14, 6, 15, 17, 7, 20, 5, 22]. These tracks can be obtained by first detecting people and objects using detectors such as Felzenszwalb *et al*. [11] and then linking the resulting detections to form tracks. For example, the detection based tracking approach of Zhang *et al*. [31] links detections into tracklets using a global data association framework based on network flows. Pirsiavash *et al*. [21] extend this approach while maintaining global-optimality by performing shortest path computations on a flow network. Berclaz *et al*. divide the scene into a network flow problem on a spatio-temporal node grid [2], which they solve using the *k-shortest path* algorithm. This approach, while not requiring the detection of bounding boxes before tracking, results in a significantly larger state-space than [31]. Ben Shitrit *et al*. extend this work by introducing a global appearance model, reducing the number of track switches for overlapping tracks [25]. While performing tracking and activity recognition sequentially simplifies action recognition, since the problem of identity maintenance can be ignored during the recognition step, mistakes performed during the tracking step cannot be overcome during recognition. Motivated by the improved results of explicitly modeling the interactions of multiple vision problems jointly (person-object, person-person, *etc*.), we perform joint identity maintenance and activity recognition.

Our work is closely related to previous work on modeling collective behavior [6, 17, 7]. Choi *et al*. [6] initially introduced this problem, proposing a spatio-temporal local (STL) descriptor that relies on an initial 2.5D tracking step which is used to construct histograms of poses (facing left, right, forward, or backward) at binned locations around an anchor person. These descriptors are aggregated over time, used as features for a linear SVM classifier with a pyramid-like kernel, and combined with velocity-based features to infer the activity of each person. Collective activity is modeled through the construction of the STL feature. In later work, Choi *et al*. [7] extend the STL descriptor by using random forests to bin the attribute space and spatio-temporal volume adaptively, in order to better discriminate between collective activities. An MRF applied over the random forest output regularizes collective activities in both time and space. Lan *et al*. [17] propose a slightly modified descriptor, the action context (AC) descriptor, which, unlike the STL descriptor, encodes the actions instead of the poses of people at nearby locations. The AC descriptor stores for each region around a person a k-dimensional response vector obtained from the output of k action classifiers.

We adopt the AC descriptor to model human actions in the context of actions performed by nearby people; however, to reason about these actions over time, we solve the problem of identity maintenance and activity recognition simultaneously in a single framework, instead of precomputing track associations. Similar to [31, 21], given human detections, we pose the problem of identity maintenance as a network flow problem, which allows us to obtain the solution exactly and efficiently, while focusing on our final goal of activity recognition.

3. Approach

3.1. Overview

Our focus in this work is to improve human action recognition. We assume that humans have already been localized, *e.g*., with a state-of-the-art multi-part model [11], or with background subtraction if the camera is stationary. Our representation for a detected human figure is based on Histogram of Oriented Gradients (HOG) [8], for which we use the popular implementation from Felzenszwalb *et al*. [11]. We augment our representation with an appearance model for tracking by blurring and subsampling the three color channels of the bounding box in *Lab* color space. We use this representation to train the action and association likelihoods used in our model. Figure 2 illustrates the overall flow of analysis, and the details are presented in Section 4.

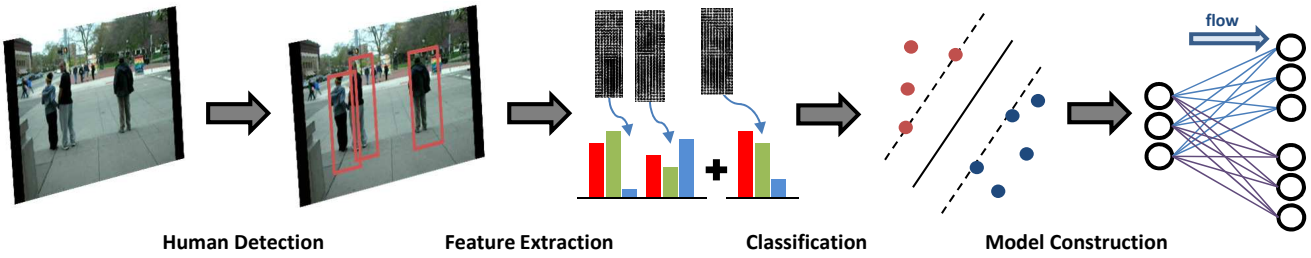

Figure 2. An overview of our system. Since our focus is human action recognition, we assume a video sequence with detected humans as bounding boxes. We then run a two-stage classification process with the Action Context (AC) descriptor [17] on top of HOG features [8] as the underlying representation. We finally use the normalized classifier scores to build our network flow-based model. See Section 3.2 for details.

3.2. Formulation

We use i, j, and k to denote the indices of human detections in a video sequence, while a, b, and c are used to denote actions. We also define $\mathcal{P}(i)$ to be the set of candidate predecessors for detection i from prior frames, and similarly $\mathcal{S}(i)$ to be the set of candidate successors of detection i from subsequent frames. We indicate the action and the identity of a detected person i by y_i and z_i, respectively. We can then formulate our model as a cost function over actions and identities represented as

$$F(\mathbf{y},\mathbf{z}) = \sum_i \sum_a \Big[u_a(i) + v'_a(i)\Big] \mathbb{1}(y_i = a), \qquad (1)$$

where $u_a(i)$ is the classification cost associated with assigning action a to person i, and $v'_a(i)$ is the associated tracking cost. Commonly, $\mathbb{1}(.)$ is defined as the indicator function.

We define the classification cost $u_a(i)$ to be the normalized negative classification score of person i performing action a. The details of the classifier training procedure is in Section 4.2.

Since a detection could designate a new person entering the scene, we define our tracking cost as

$$v'_a(i) = \begin{cases} v_{ab}(i,j) & \textit{if } \exists j \in \mathcal{P}(i) \textit{ s.t. } z_i = z_j,\ y_j = b, \\ \lambda_0 & \textit{otherwise,} \end{cases} \qquad (2)$$

where $v_{ab}(i,j)$ is the transition cost that links "person i performing action a" to a previously tracked "person j performing action b". If the newly detected person i does not sufficiently match any of the people previously tracked, the model incurs a penalty represented by the tuning parameter λ_0, and a new track is established. We define the transition cost $v_{ab}(i,j)$ as

$$v_{ab}(i,j) = \lambda_d \, d(i,j) - \lambda_c \log(p_{ab}), \qquad (3)$$

which is a mixture of an appearance term and an action consistency term. The appearance term measures the similarity between person i and person j with a distance metric $d(i,j)$, and the action consistency term measures the prior probability p_{ab} of a person performing action a followed by action b. The tuning parameters λ_d and λ_c weigh the importance of those two terms. The models for calculating both the appearance distance metric $d(i,j)$ and the action co-occurrences p_{ab} are provided in Section 4.3.

Maximum-a-posteriori (MAP) estimation in our model can be formulated as the minimum of an integer linear program (ILP). We define the following program

$$\begin{aligned} \min_{\{\mathbf{e},\mathbf{t},\mathbf{x}\}} \quad & \sum_i \sum_a \Big[(u_a(i) + \lambda_0) e_a(i) + \\ & \qquad \sum_{j \in \mathcal{P}(i)} \sum_b (u_a(i) + v_{ab}(i,j)) t_{ab}(i,j)\Big], \\ s.t. \quad & e_a(i) + \sum_{j \in \mathcal{P}(i)} \sum_b t_{ab}(i,j) = \\ & \qquad x_a(i) + \sum_{k \in \mathcal{S}(i)} \sum_c t_{ca}(k,i) \quad \forall i, a \\ & \sum_a \Big[e_a(i) + \sum_{j \in \mathcal{P}(i)} \sum_b t_{ab}(i,j)\Big] = 1 \quad \forall i \\ & \{\mathbf{e},\mathbf{t},\mathbf{x}\} \in \mathbb{B}^n, \end{aligned} \qquad (4)$$

where variable $e_a(i)$ denotes the entrance of person i into the scene performing action a, while variable $t_{ab}(i,j)$ denotes the transition link of person i performing action a to person j performing action b. Finally, variable $x_a(i)$ denotes person i exiting the scene after performing action a. The entrance, transition, and exit variables are defined to be binary indicators. The costs $u_a(i)$ and $v_{ab}(i,j)$ are as previously defined.

Minimizing the program in Equation 4 is equivalent to inference in the model from Equation 1. A detected human figure would always encounter a classification cost, whether it is linked to a previously tracked detection, or is entering

the scene for the first time. Consequently, it will either incur the transition cost to link it to the previously tracked detection, or incur the penalty of not having a sufficiently matching predecessor. The two constraints enforce a valid assignment according to Equations 1 and 2.

The variables $\mathbf{e}$, $\mathbf{t}$, and $\mathbf{x}$ always recover a unique assignment for $\mathbf{y}$ and $\mathbf{z}$. Specifically, if detection i just entered the scene, it will be assigned action $y_i = a$ for which $e_a(i) = 1$ and its identity z_i will be assigned to an unused track number. Otherwise, detection i will be instead linked to a previous detection; in that case, it will be assigned action $y_i = a$ for which $t_{ca}(k, i) = 1$ and the identity will propagate from that previous detection: $z_i = z_k$.

The ILP in Equation 4 represents a network flow problem. In fact, the first constraint of the ILP is the "flow conservation constraint" (or *Kirchoff's Laws*). However, the second constraint, which we refer to as the "explanation constraint", is not typically encountered in the minimum cost flow problem. In our case, it enforces that an action and an identity be assigned to every person detected in the video. Figure 3 illustrates the flow graph of an example with 3 frames, 5 detections, and 3 possible actions per person. Each person is represented by a subset of nodes, and is connected to people from the previous frame, or more generally, from any previous frame. The connection between two people is a complete bipartite subgraph between their nodes. The flow of the minimum cost in the network uniquely assigns actions and identities to every detected person in the video sequence.

3.3. Inference

While minimim cost flow problems with side constraints can generally be solved by *Lagrangian Relaxation* (also known as *Dual Decomposition*) [1], the form of our constraints allows us to provide fast alternative solutions. As shown in Equation 4 and Figure 3, our formulation uses constraints on sets of nodes, which motivates us to explore the link between our model and the so-called Neoflow problems [12], a set of equivalent generalized network flow problems that includes submodular flow. Our model is a special case of the submodular flow problem. The submodular flow problem, introduced by Edmonds and Giles [9], generalizes the flow conservation constraints of classical network flows to submodular functions on sets of nodes. The max-flow min-cut theorem still holds in this more general setting [12], and polynomial-time algorithms to solve this class of problems exist [13, 24].

While we could use any general submodular flow algorithm available [13, 24], we emphasize that constraining the ILP in Equation 4 to the submodular polyhedron implies a totally unimodular constraint matrix [12]. Consequently, we can relax the binary constraint to an interval constraint and still guarantee an integer solution to the linear program. We therefore opted for a fast interior-point solver. To improve the inference speed, we only connect people with overlapping bounding boxes in consecutive frames. Solving the cost function exactly takes an average of 1.2 seconds for an average sequence length of 520 frames, where each sequence is subsampled every ten frames during model construction.

4. Learning the Potentials

4.1. Piecewise Training

Since inference in our model is exact and latent variables are absent, global training approaches become not only possible, but deterministic. However, for practical reasons, we chose to use piecewise training [27]. Piecewise training involves dividing the model into several components, each of which is trained independently. We are motivated by recent theoretical and practical results. Theoretically speaking, piecewise training minimizes an upper bound on the log partition function of the model, which corresponds to maximizing a lower bound on the exact likelihood. In practice, the experiments of [27, 26] show that piecewise training sometimes outperforms global training, even when joint full inference is used. We choose to divide our model training across potentials, i.e., we train the three groups of potentials–unary action, binary action consistency, and binary appearance consistency–independently from each other. The tuning parameters that weigh the importance of the individual terms were set manually through visual inspection.

4.2. Action Potentials

We now describe how we train our action likelihood potentials. We use the AC descriptor from Lan *et al.* [17]. We utilize HOG features as the underlying representation. We then train a multi-class linear SVM using *LibLinear* [10]. Next, a bag-of-words style representation for the action descriptor of each person is built. Each person is represented by the associated classifier scores, and the strongest classifier response for every action in a set of defined neighborhood regions in their context.

The descriptor of the i-th person becomes the concatenation of their action scores and context scores. The action scores for person i, given A possible actions, become $\mathbf{F}_i = [s_1(i), s_2(i), \ldots, s_A(i)]$, where $s_a(i)$ is the score of classifying person i to action a. The context score, defined over M neighborhood regions, is

$$\mathbf{C}_i = \Big[\max_{j \in \mathcal{N}_1(i)} s_1(j), \ldots, \max_{j \in \mathcal{N}_1(i)} s_A(j), \ldots, \max_{j \in \mathcal{N}_M(i)} s_1(j), \ldots, \max_{j \in \mathcal{N}_M(i)} s_A(j) \Big], \quad (5)$$

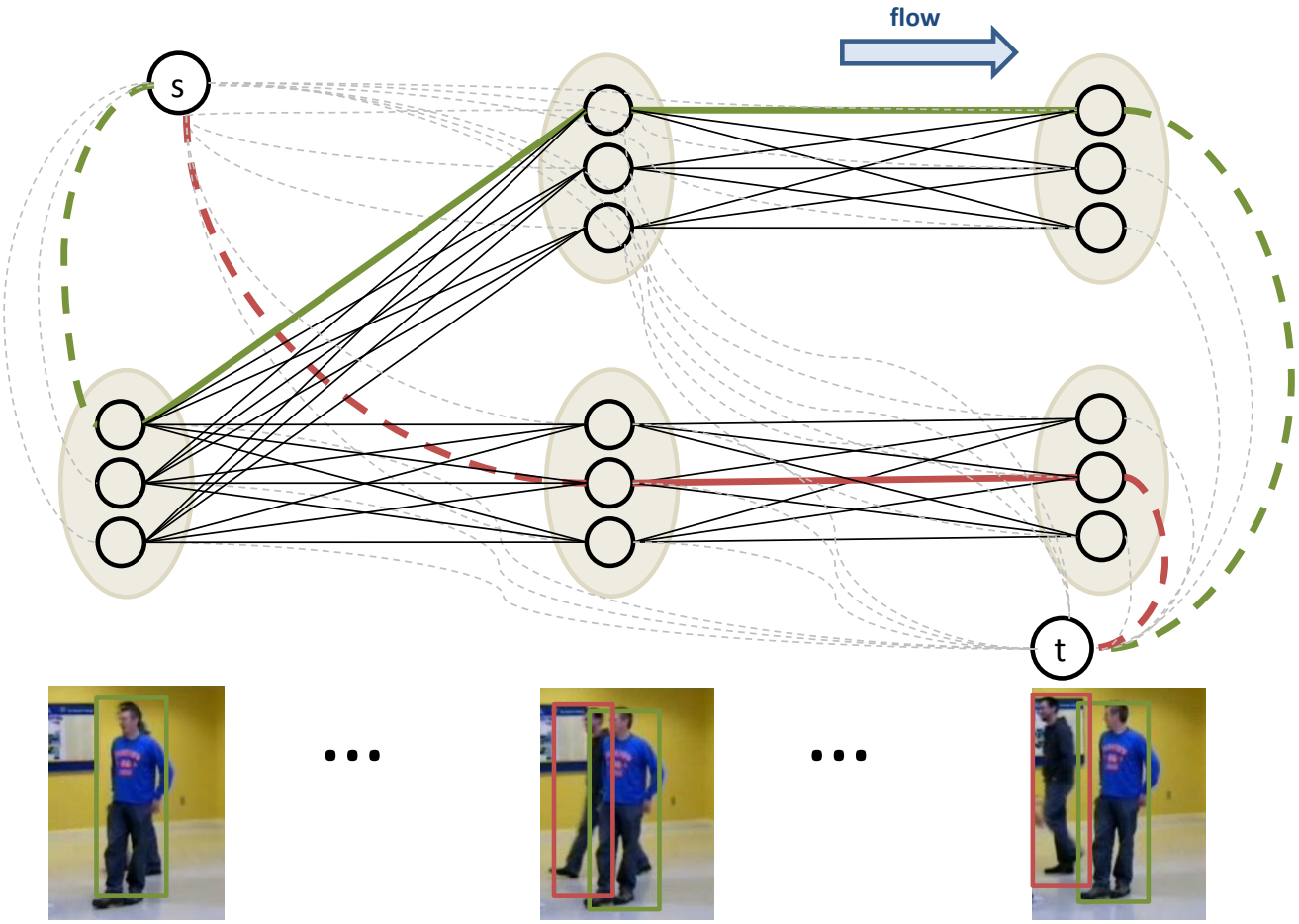

Figure 3. An illustration of our flow model. Every grouped subset of nodes represents a detection, and the nodes in the subset are potential actions for that detection. Every detection forms a complete bipartite graph with its predecessors and successors. Here people in every frame are connected to those in the previous frame, but that can be generalized to any subset of people in any number of frames. The flow goes from the source node to the sink node assigning actions and identities that minimize our integer linear program in Equation 4. By enforcing the "explanation constraint", we are guaranteed an action and an identity for every person in the graph. The colored arcs in the diagram represent a valid complete assignment in the frame sequence at the bottom. The person outlined in green enters in the first frame, performs the first action for the entire sequence, and exits in the final frame, while the person outlined in red enters in the second frame, performs the second action, before exiting at the final frame. Section 3.2 provides the technical details.

where $\mathcal{N}_m(i)$ is a list of people in the m-th region in the neighborhood of the i-th person. We use the same "sub-context regions" as [17]. We then run a second-stage classifer on the extracted AC descriptor using the same multi-class linear SVM implementation of *LibLinear* [10]. The classifier scores are negated and then normalized using a softmax function, and finally incorporated as the unary action likelihood potentials $u_a(i)$, which assign action a to person i.

4.3. Association Potentials

To track the identities of the targets in our video sequences, we train identity association potentials and incorporate them in our model. Our association potentials use both appearance and action consistency cues. The appearance cues are trained using the subsampled color channels as features. We train for a Mahalanobis distance matrix M to estimate the similarity between detections across frames. The distance matrix is learned so as to bring detections from the same track closer, and those from different tracks apart [4]. This is formulated as

$$\mathbf{M}^* = \arg\min_{\mathbf{M}} \sum_{\mathcal{T}_k} \Big[\sum_{i,j \in \mathcal{T}_k} (\mathbf{f}_i - \mathbf{f}_j)^T \mathbf{M} (\mathbf{f}_i - \mathbf{f}_j) - \sum_{i' \in \mathcal{T}_k, j' \notin \mathcal{T}_k} (\mathbf{f}_{i'} - \mathbf{f}_{j'})^T \mathbf{M} (\mathbf{f}_{i'} - \mathbf{f}_{j'}) \Big], \quad (6)$$

where $\mathcal{T}_k$ is the k-th track and $\mathbf{f}_i$ is the feature vector of the i-th person. We solve for M using the fast Large Margin Nearest Neighbor (LMNN) implementation of [28]. The distance between two people i and j can then be defined as

$$d(i,j) = (\mathbf{f}_i - \mathbf{f}_j)^T \mathbf{M} (\mathbf{f}_i - \mathbf{f}_j). \quad (7)$$

The action consistency cues are estimated using the groundtruth action labels from the training set. We count pairwise co-occurrences of actions on the same track plus a small additive smoothing parameter α. The counts are normalized into the pairwise co-occurrence probabilities p_{ab} of action pairs a and b.

5. Experiments

5.1. Datasets

We use the group actions dataset from [6] and its augmentation from [7] to evaluate our model. The datasets are appropriate since they have multiple targets in a natural setting, while most action datasets, like KTH [23] or Weizmann [3], have a single person performing a specific action. The original dataset includes 5 action classes: *crossing*, *standing*, *queueing*, *walking*, and *talking*. The augmented dataset includes 6 action classes: *crossing*, *standing*, *queueing*, *talking*, *dancing*, and *jogging*. The *walking* action was removed from the augmented dataset because it is ill-defined [6]. We only use the bounding boxes, the associated actions, and the identities. We did not use any of the 3-D trajectory information.

Our main focus here is action recognition, and tracking is used only to improve the performance in the full model. While we show that joint optimization improves action recognition through tracking, it is intuitive that tracking performance will also improve through action recognition. However, such an evaluation is outside the scope of our work. We evaluate our results similar to [6, 7]. For each dataset, we perform a leave-one-video-out cross-validation scheme. This means that when we classify the actions in one video, we use all the other videos in the dataset for training and validation. Our action potentials are based on [17], which we also compare against to analyze the efficacy of our approach.

5.2. Results

Our confusion matrices for the 5-class and the 6-class datasets are shown in Figure 4. It is clear that removing the *walking* activity improves the classification performance, possibly due to the apparent ambiguity between *walking* and *crossing*. Our average classification accuracy is 70.9% on the former dataset and 83.7% on the latter.

We outperform the state-of-the-art methods on the two datasets, as shown in Table 1. Classification using the AC descriptor that we employ was reported in [17], which we improve upon. The model from [7] yields the same performance as our model for the first dataset. However, it employs additional trajectory information, including the 3D location and the pose of every person [7].

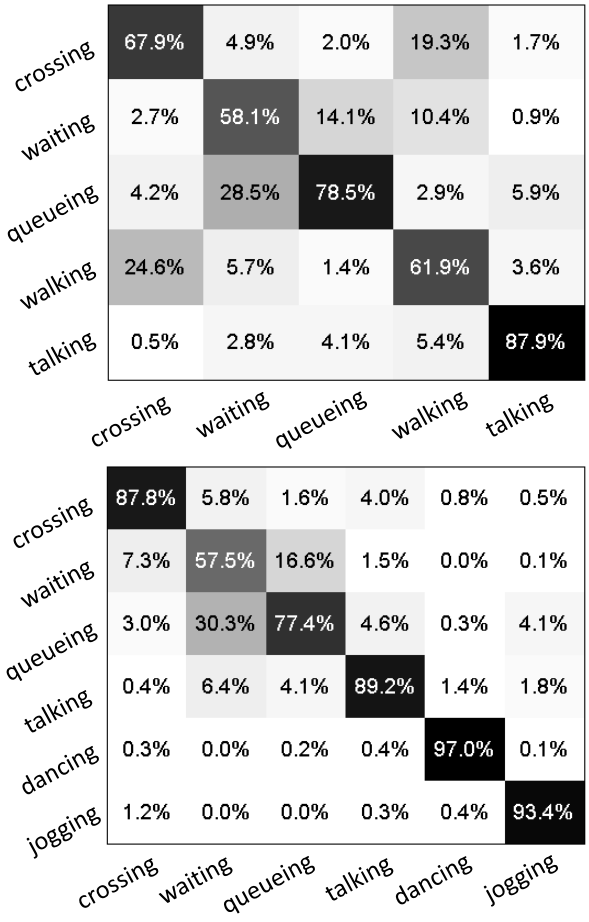

Figure 4. Our confusion matrices for the 5-class [6] and the 6-class [7] datasets. The confusion matrices were obtained using the full model. Our classification accuracy is 70.9% on the 5-class dataset and 83.7% on the 6-class dataset.

We also report qualitative results on the 6-activity dataset in Figure 5. Each row in the figure represents a different video sequence. The first 3 sequences are successful cases where the full model improves the action classification results in an adjacent frame, while the final row represents one failure case where the high confidence of the action classifier in the wrong label causes the full model to misclassify the action in the consecutive frame.

6. Conclusion

We evaluated how tracking identities helps recover consistent actions across frames, and we unified action classifi-

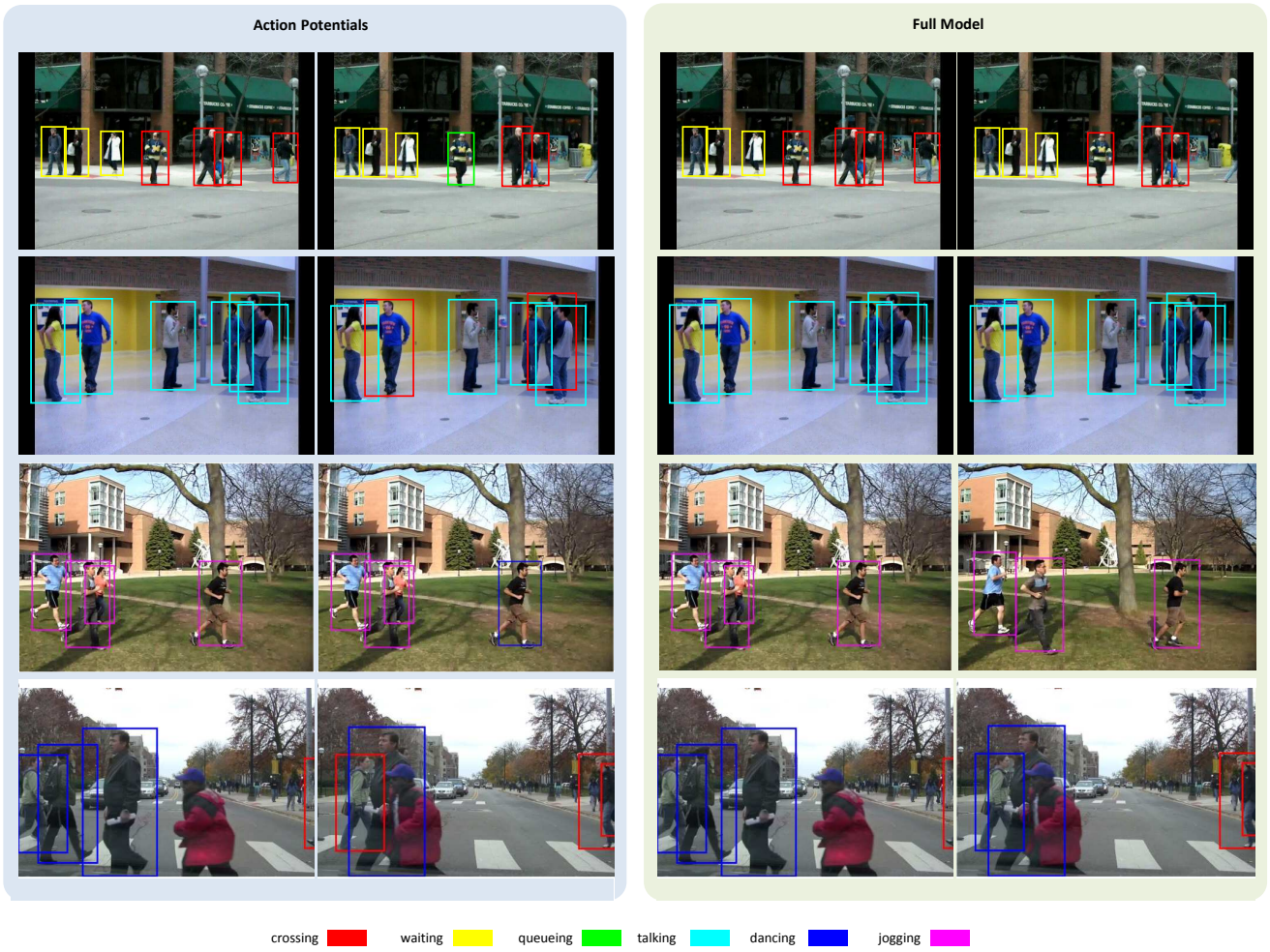

Figure 5. Qualitative results with and without our full model. The first two columns are the results of two consecutive frames from the same video sequence using only the action potentials, and the next two columns are the results of the same two frames, but using our full model. Each row represents a different video sequence. The first row shows a video sequence where the misclassification of *crossing* as *queueing* is fixed with correct tracking. The second shows the same case for *talking* being misclassified as *crossing*, and the third for *jogging* being misclassified as *dancing*. The fourth row is a case where the full model actually decreases the classification accuracy due to the high confidence of the action classifier in the wrong label.

Approach/Dataset	5 Activities	6 Activities
AC [17]	68.2%	-
STV+MC [6]	65.9%	-
RSTV [7]*	67.2%	71.7%
RSTV+MRF [7]*	70.9%	82.0%
AC	68.8%	81.5%
AC+Flow	**70.9%**	**83.7%**

Table 1. A comparison of classification accuracies of the state-of-the-art methods on the two datasets. * While the full model from [7] yields similar results to our model, their model training employs additional trajectory information, including the 3D location and the pose of every person.

cation and identity maintenance in a single framework. We proposed an efficient flow model to jointly solve both problems, which could be solved by a myriad of polynomial-time algorithms. In practice, we can assign actions and identities to every person in one video sequence in roughly one second. We reported our action recognition results on two datasets, and outperformed the state-of-the-art approaches using the same leave-one-out validation scheme. Our model generalizes minimum cost flow and is theoretically linked to other Neoflow problems [12]. It is general, fast, and can be easily adapted to other problems in computer vision.

Acknowledgements

This research was partially supported by ONR MURI grant N000141010934 and by a grant from Siemens Corporate Research in Princeton, NJ. The first author would like to thank Tian Lan for the prompt email correspondences about his work.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993. 4

[2] J. Berclaz, F. Fleuret, E. Tretken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819, 2011. 2

[3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *International Conference on Computer Vision*, 2005. 6

[4] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum-weight independent set. In *Conference on Computer Vision and Pattern Recognition*, 2011. 5

[5] W. Brendel, S. Todorovic, and A. Fern. Probabilistic event logic for interval-based event recognition. In *Conference on Computer Vision and Pattern Recognition*, 2011. 2

[6] W. Choi, K. Shahid, and S. Savarese. What are they doing?: Collective activity classification using spatio-temporal relationship among people. In *International Workshop on Visual Surveillance*, 2009. 2, 6, 7

[7] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *Conference on Computer Vision and Pattern Recognition*, 2011. 2, 6, 7

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Conference on Computer Vision and Pattern Recognition*, 2005. 2, 3

[9] J. Edmonds and R. Giles. A min-max relation for submodular functions on graphs. *Annals of Discrete Mathematics*, 1:185–204, 1997. 4

[10] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. 4, 5

[11] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Conference on Computer Vision and Pattern Recognition*, 2008. 2

[12] S. Fujishige. *Submodular Functions and Optimization*. Elsevier Science, 2005. 4, 7

[13] S. Fujishige and S. Iwata. Algorithms for submodular flows. *IEICE Transactions on Information and Systems*, E83-D:322–329, 2000. 4

[14] A. Gupta and L. S. Davis. Objects in action: An approach for combining action understanding and object perception. In *Conference on Computer Vision and Pattern Recognition*, 2007. 2

[15] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *Conference on Computer Vision and Pattern Recognition*, 2009. 2

[16] A. Hakeem and M. Shah. Learning, detection and representation of multi-agent events in videos. *Artificial Intelligence*, 2007. 2

[17] T. Lan, Y. Wang, G. Mori, and S. N. Robinovitch. Retrieving actions in group contexts. In *International Workshop on Sign, Gesture, and Activity*, 2010. 1, 2, 3, 4, 5, 6, 7

[18] T. Lan, Y. Wang, W. Yang, and G. Mori. Beyond actions: Discriminative models for contextual group activities. In *Neural Information Processing Systems*, 2010. 2

[19] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009. 2

[20] V. I. Morariu and L. S. Davis. Multi-agent event recognition in structured scenarios. In *Conference on Computer Vision and Pattern Recognition*, 2011. 2

[21] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Conference on Computer Vision and Pattern Recognition*, 2011. 2

[22] M. S. Ryoo and J. K. Aggarwal. Stochastic representation and recognition of high-level group activities. *International Journal of Computer Vision*, 93(2):183–200, 2010. 2

[23] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *International Conference on Pattern Recognition*, 2004. 6

[24] M. Shigeno and S. Iwata. A cost-scaling algorithm for 0-1 submodular flows. *Discrete Applied Mathematics*, 73(3):261–273, 1997. 4

[25] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *International Conference on Computer Vision*, 2011. 2

[26] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision*, 2006. 4

[27] C. Sutton and A. McCallum. Piecewise training for undirected models. In *Conference on Uncertainty in Artificial Intelligence*, 2005. 4

[28] K. Q. Weinberger and L. K. Saul. Fast solvers and efficient implementations for distance metric learning. In *International Conference on Machine Learning*, 2008. 6

[29] T. Xiang and S. Gong. Beyond tracking: modelling activity and understanding behaviour. *International Journal of Computer Vision*, 67:21–51, 2006. 2

[30] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. *Conference on Computer Vision and Pattern Recognition*, 2010. 2

[31] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Conference on Computer Vision and Pattern Recognition*, 2008. 2