Semi-supervised Learning for Identifying Players from Broadcast Sports Videos with Play-by-Play Information

Wei-Lwun Lu, Jo-Anne Ting, James J. Little, and Kevin P. Murphy
University of British Columbia, Vancouver, BC, Canada
{vailen,jating,little,murphyk}@cs.ubc.ca

Abstract

Tracking and identifying players in sports videos filmed with a single moving pan-tilt-zoom camera has many applications, but it is also a challenging problem due to fast camera motions, unpredictable player movements, and unreliable visual features. Recently, [26] introduced a system to tackle this problem based on conditional random fields. However, their system requires a large number of labeled images for training. In this paper, we take advantage of weakly labeled data in the form of publicly available play-by-play information. This, together with semi-supervised learning, allows us to train an identification system with very little supervision. Experiments show that by using only 1500 labels with the play-by-play information in a dataset of 75000 images, we can train a system that has a comparable accuracy as a fully supervised model trained by using 75000 labels.

1. Introduction

Automatic player tracking and identification in sports videos filmed with a single moving pan-tilt-zoom camera has many applications for video retrieval and analysis. However, it is a very challenging computer vision problem. Tracking players in this scenario is particularly hard because their motion patterns are unpredictable and camera movement is fast (in contrast, tracking pedestrians is easier due to static cameras and a simpler motion pattern). Identifying players is even a harder problem because biometric features are unreliable and player dimensions are difficult to measure. For example, faces are blurry and low-resolution, making it impossible even for humans to identify players from only faces. Numbers on jerseys appear infrequently and, when visible, are often deformed due to player movement. This, along with the fact that detecting numbers in far-field shots is hard, makes number recognition tricky. Colors are also not strong cues because players on a team have the same jersey color, and many have the same hair/skin color.

[26] presented an approach to solve this problem based on conditional random fields (CRFs). The basic idea is to use a tracking-by-detection approach, followed by learning a mapping from the visual features of detections to player identities. To overcome local ambiguity, they propagated label information along tracklets representing each player's trajectory; they also enforced mutual exclusion constraints, not allowing a player to appear more than once at the same time. They showed that this approach significantly outperformed classifying each detection independently.

The main drawback of [26] is the amount of labeled data required to train the classifier. The labeled images consist of manually drawing bounding boxes around the player and assigning the true identity. Acquiring such labeled data is very time consuming. For example, even with a suitable software package, it takes an experienced human expert about 2 hours to label a video clip of 500 frames, assuming each frame has ~ 10 detections. Since a typical sports game lasts about 1 hour, this means it would take more than 300 hours to label a full game (assuming 30 frames/ second).

One solution to the label acquisition problem is to use crowd sourcing services such as Amazon's Mechanical Turk (e.g., Vondrick *et al*. [35]), but quality control of labels is not guaranteed and this still requires human effort. We adopt a different approach, favoring a solution that can be easily deployed for multiple sports with little human effort or cost. An alternative solution is the use of *weak labels*. A typical source of weak labels are the captions/subtitles that come with videos, which specify what or who is in the image, but not where. Such weakly labeled data is often cheaply available in large quantities. However, a hard correspondence problem between labels and objects has to be solved. An early example of this approach is [3], who learned a correspondence between captions and regions in a segmented image. [11, 12, 14, 33] learned a mapping between names in subtitles of movies to appearances of faces, extracted by face detection systems. Others have also attempted to learn action recognition systems from subtitles [13, 23, 27] or a storyline [19].

In this paper, we consider semi-supervised learning of appearance models for sports players, by combining our CRF with EM-based training. Our scenario is more difficult than prior work [11, 12, 14, 33], since each frame has 8-10 players, whereas most movies only contain 1-2 dominant faces per frame. In addition, it is generally much harder to track sports players than to track faces.

We use a source of weak labels, namely play-by-play text data, which has not received much attention. Play-by-play data is freely available for most broadcast sport games, including basketball, soccer and hockey. Descriptions in play-by-plays usually come in chronological order and contain the time of the event, the players involved, and text describing the event (see Section 3 for details). Although play-by-play has been previously used as features for event tactic analysis [38], it has not been used, as far as we know, to train a vision-based system.

The second major problem with [26] is that the approximate inference method used in the CRF sometimes violated mutual exclusion constraints. We propose a slightly different graphical model, which collapses tracklets into a single node. We propose a modified approximate inference algorithm and show our approach gives improved results.

Our focus in this paper is on tracking and player identification in NBA basketball games. However, our techniques are quite general, and we are currently working on extending them to ice hockey and soccer.

2. Related work

We can divide the prior work into three main categories: tracking, player identification, and weakly labeled learning. We have already discussed some approaches to weakly labeled learning; we summarize some relevant approaches to tracking and player identification below.

Reviewing all relevant tracking papers is beyond the scope of this paper. Interested readers can consult [37] for a survey of tracking systems. Our tracking algorithm is similar to [9], who use bi-partite matching to associate detections with targets and then use a Boosted Particle Filter [30] for tracking. We also borrow techniques from [17, 24], who use data-driven MCMC to create tracklets from detections.

Previous player identification systems in the sports domain have focused on videos shot with a close-up camera, and they relied on recognizing frontal faces or numbers on the jersey. For instance, [2, 4, 5] trained face recognition systems to identify players. [32, 36] developed systems to identify players by recognizing their jersey numbers. However, these systems only apply to close-up and frontal-view images where facial features are clear and/or jersey numbers are visible. In order to tackle videos filmed from a far-range distance containing partial views of players, [26] proposed a robust appearance model with a CRF that incorporated temporal consistency and mutex constraints. They demonstrated promising identification results using images taken from a medium-distance camera.

00:42.3 [LAL 51-29]	**Bynum Slam Dunk Shot: Made (5 PTS) Assist: Bryant (5 AST)**
00:35.8	Bynum Foul : Personal (1 PF)
00:35.8	Bryant Substitution replaced by Blake
00:31.1	Brown Foul : Personal (1 PF)

Figure 1. Snapshot of play-by-play of a basketball game. Play-by-plays show the time and people involved in important sport events. Play-by-play information is available for most broadcast sport games and can be freely downloaded from websites of leagues and teams [1].

In the surveillance community, there is a related problem called pedestrian re-identification, where the goal is to find a pedestrian of some appearance over a network of cameras (e.g., [15, 18, 21]). However, since most of these systems rely on color, shape or texture, they cannot be directly applied to sport videos due to the uniformity of jersey colors in a team. Some systems even use the geometric configuration of cameras to help re-identify pedestrians (e.g., [21]), which is also not applicable in our case because we only have a single moving camera.

3. Play-by-play data

As mentioned in Section 1, play-by-play consists of text summarizing important events in sport games, specifying who did what to whom, when and roughly where. Figure 1 shows an example of play-by-play downloaded from the NBA website [1]. We see that it shows event types (e.g., "Dunk", "Foul", "Substitution"), player names (e.g., "Bryant", "Bynum"), and the timestamps (e.g., "00:42.3"). In this paper, we focus on player identity, rather than actions, since we want to train a player identification system.

The timestamps shown in play-by-play are measured by the game clock, which do not match the internal clock of the video recording device. Therefore, in order to use play-by-play data, we have to first *synchronize* them with sports videos. To do this, we exploit the fact that there is usually an information bar overlaid on most broadcast sport videos for showing game time, team names, and scores (see Figure 5). We then run an off-the-shelf OCR system [34] on the overlaid clock region to recognize the game time of every video frame. The OCR system has nearly perfect accuracy because the clock region has a fixed location and background of the clock is homogeneous.

By combining the synchronized video and play-by-play text, we obtain information about who is present on the court/field at any given moment by using the substitution events. However, we do not know their locations. In this paper, we assume that the mentioned players are somewhere in the current frame; relaxing this assumption is left to future work.

4. Player tracking

We adopt a similar approach to [26] to detect and track multiple players in sports videos. The strategy is tracking-by-detection, where we first run an object detector to locate players before associating detections with tracklets.

We use the Deformable Part Model (DPM) [16] to detect players. Our DPM consists of six parts and three aspect ratios and is able to achieve a precision of 73% and a recall of 78% in test videos. Most false positive detections arise from spectators/referees, who have similar shapes to players.

In order to reduce the number of false positive detections, we use the fact that players of the same team wear jerseys of the same colors. We collect training images of players from different teams, extract RGB color histograms, and then train a logistic regression classifier [6] that maps images to team labels. We can then filter out false positive detections (spectators, referees) and, at the same time, also group detections into their respective teams. After performing this step, we significantly boost precision to 97% while retaining a recall level of 74%.

We perform tracking by associating detections with tracklets and use an one-pass approach similar to Cai *et al*. [9]. Starting from the current frame, we assign detections to existing tracklets. To ensure the assignment is one-to-one, we use bi-partite matching where the matching cost is the Euclidean distances between centers of detections and predictive locations of tracklets. Then, we use a Kalman Filter [22] to update the current state estimate, based on a locally linear motion model that is updated online. A new tracklet is initialized for any unassigned detection. However, the tracklet is dropped/removed if it fails to have a sufficient number of detections associated with it after a short time. Existing tracklets are also removed if they are close to image boundaries or if they fail to have any detections associated with them for a sufficient period of time.

The tracking algorithm has a 98% precision with an improved recall of 82%. This is because the original detections are temporally sparse and tracking helps to bridge the temporal gap between disjointed detections. It is worthwhile to note that problem of tracking in broadcast sport videos is non-trivial due to fast camera movements and unpredictable player motions. We have tried off-the-shelf trackers (e.g., [30]), but they yield worse results.

5. Player identification

Once we run automatic tracking on all training and test video clips, the next step is to identify the player each tracklet represents. In this section, we describe a multi-class appearance model that is learned over all player classes in a semi-supervised framework with weak play-by-play labels. We show how the appearance model can be used to predict player identities on unlabeled test clips.

5.1. Player appearance models

Due to the moving pan-tilt-zoom camera filmed at a distance, facial recognition is impractical since faces of players are blurry and of low resolution. Number recognition is possible but difficult since numbers are small (10×10 pixels), appear infrequently and are often deformed due to player movements as mentioned in [26].

We identify players in their entirety and use the same features as [26]. We extract MSER regions [28] and SIFT interest points [25], and compute their 128-dim. SIFT descriptors [25]. We then quantize them into 300 MSER words and 500 SIFT words from a codebook learned with k-means clustering. The final image representation consists of a 800-dim. bag-of-words bit vector (where 1 indicates presence of a visual word in the image and 0 otherwise) and a 30-bin RGB color histogram (with 10 bins for each channel). [26] found that using number recognition alone was insufficient and using all three features was most discriminative.

There are many other features that could be informative about player identity, such as player location, size, motion style, etc. However, it is hard to extract such features, since the moving camera makes computing homographies quite difficult. This is the subject of on-going work.

5.2. Supervised learning

[26] learned a player appearance model from a fully labeled training set, using an L1-regularized logistic regression classifier that mapped image feature vectors to player class labels. Their training set had 9318 frames containing 34798 labeled image patches.

Note that a label is defined on a per detection (not frame) basis. A *labeled image* consists of not only the *true player class identity* but also a *manually drawn bounding box around the player*. Unlabeled samples are images that have no player class labels and are obtained using automatic player detection (as described in Section 4). We present images to the human annotator to label, randomly sampling detections from the set of training clips (i.e., images to be labeled are drawn from a "bag-of-detections", with each detection treated independently, regardless of frame and clip). Though we sampled detections randomly, smarter sampling strategies (e.g., active learning), could have been used, and we leave this for future work.

5.3. Semi-supervised learning

We now discuss how to exploit weak labels during training. Given a set of training clips, we construct a CRF for each clip (see Figure 2), as done in [26]. The observed variable $\mathbf{x}_{td}$ represents the feature vector for detection d in frame t. The hidden variable y_{td} is the detection's identity (with C possible values, where C is the number of possible player classes). Detections belonging to the same tracklet

are connected with temporal edges having the potential:

$$\psi_{time}(y_{tj}, y_{t+1,k}) = \begin{cases} 1-\epsilon & \text{if } y_{tj} = y_{t+1,k} \\ \epsilon & \text{otherwise} \end{cases} \quad (1)$$

where ϵ is a fixed parameter reflecting the amount of linking errors in the tracker. Setting ϵ to 0 forces the detection identities in a tracklet to be the same. Since no one can be in two places at a time, we enforce mutual exclusion in detection identities by introducing pairwise edges between y nodes in each frame, using the potential:

$$\psi_{mutex}(y_{tj}, y_{tk}) = \begin{cases} 1 & \text{if } y_{tj} \neq y_{tk} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The log posterior of detection identities is then:

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{x}, \mathbf{L}, \boldsymbol{\theta}) \propto & \sum_{t=1}^{T} \sum_{d=1}^{D_t} \log p(y_{td}|\mathbf{x}_{td}, \boldsymbol{\theta}) p(L_{td}|y_{td}) \\ & + \sum_{t=1}^{T} \sum_{d=1}^{D_t} \sum_{j=1, j\neq d}^{D_t} \log \psi_{mutex}(y_{td}, y_{tj}) \\ & + \sum_{t=1}^{T} \sum_{d=1}^{D_t} \sum_{j:succ(d,t)=j} \log \psi_{time}(y_{td}, y_{t+1,j}) \end{aligned} \quad (3)$$

where $succ(d,t)$ is the next node (if it exists) that is connected to y_{td} in the tracklet, $\boldsymbol{\theta}$ are model parameters, D_t is the number of detections in frame t, and T is the total number of frames in the training set.

To handle labels, let us associate a label node L_{td} with each hidden identity node y_{td}. In the supervised setting, L_{td} is the known identity of y_{td}. In the unsupervised setting, L_{td} is the empty set, which implies no constraints on y_{td}. In the weakly supervised setting, L_{td} is a set, and y_{td} is constrained to only have non-zero probability mass for labels which are part of that set. We can handle these label nodes by incorporating $p(L_{td}|y_{td})$ as soft local evidence into the model i.e., as a unary potential.

Having specified the model, we now discuss how to train it. The basic idea is to use EM. We initialize the logistic regression parameters using some labeled data. For the E step, we perform inference in the CRF. We use loopy belief propagation (BP) [29], since exact inference is intractable due to large clique size in the graphical model. The output of the E step are the node marginals, $p(y_{td}|\mathbf{x}, \mathbf{L}, \boldsymbol{\theta})$. We can then use these as "soft targets" in the M step, which reduces to a weighted form of L1-regularized logistic regression.

Since EM is prone to local maxima, we perform multiple random restarts. We are currently exploring more sophisticated approaches, based on annealing and other semi-supervised learning strategies such as co-training [8] and graph-based methods [7, 10]).

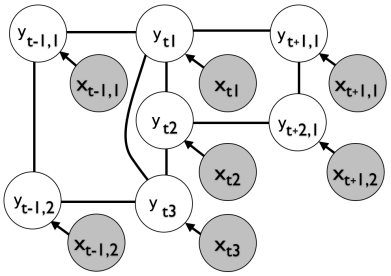


Figure 2. Graphical model for training clips. $\mathbf{x}$ are detections. Mutex arcs exist between detection identities $\mathbf{y}$ in a frame. Temporal arcs exist between $\mathbf{y}$ nodes across frames.

5.4. Prediction at test time

Given the learned model, we can use it to identify players in unlabeled test videos by running inference exactly as in the E step. In particular, we can run loopy BP and then compute the max marginals, as described in [26]. In practice, however, we discovered that this strategy sometimes generates solutions that violate mutual exclusion constraints.

To resolve this problem, we propose to perform inference in a modified model. The basic idea is to transform the graphical model in Figure 2 into a *tracklet graph*, where we have one node per tracklet (set of connected y_{td} nodes), as in Figure 3, where v_i is the identity of the tracklet. We no longer need to enforce that all the connected y_{td}'s have the same label (temporal edges), because they have all been collapsed into a single node. However, we still need to enforce mutual exclusion constraints. In particular, we add an edge between nodes v_i and v_j if both tracklets appear in the same frame at any time. We impose a hard mutex constraint on this edge. We currently assume all the observations in a tracklet are iid, so we use a local evidence potential of the form

$$\phi(v_i = k, \mathbf{x}_i) = \prod_{t,d \text{ in tracklet}(i)} p(y_{td}|\mathbf{x}_{td}, \boldsymbol{\theta}) \quad (4)$$

where $\mathbf{x}_i$ are all features from all detections in tracklet i, and $p(y_{td}|\mathbf{x}_{td}, \boldsymbol{\theta})$ is the local evidence computed by the logistic regression classifier. Consequently, the posterior over tracklet assignments can be written as follows:

$$p(\mathbf{v}|\mathbf{x}) \propto \prod_i \phi(v_i, \mathbf{x}_i) \quad \mathbf{v} \text{ is a valid assignment} \quad (5)$$

An additional advantage of the tracklet graph, as opposed to the original detection-level graph, is that we can easily use motion features; however, we leave this to future work.

Since the number of tracklets is much smaller than the number of detections, the search space in the tracklet graph is also smaller. We use greedy local search with random

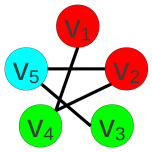


Figure 3. A tracklet graph example. Every node v_i represents the identity of a tracklet. There is an edge between v_i and v_j if they appear in the same frame and, thus, cannot have the same identity. Different colors indicate different identities. The goal is to find the optimal assignment that maximizes the posterior in Eq. (3).

restarts [20] to find a local optimum. Starting with a random valid assignment, we search for the local move that best improves the posterior. A local move can be computed by flipping the current identity of node v_i to any other valid one. We perform successive moves until reaching the local optimum and then restart the search with another random assignment. We repeat this procedure multiple times to improve the solution quality and reduce variance.

Since tracking may contain errors, we also search for different tracklet graph configurations. We use MCMC search with two kinds of proposals: splitting tracklets and merging tracklets (This is similar to the DDMCMC approach described in [24].) We start from the initial tracklet graph generated by the tracking algorithm described in Section 4 and compute the local optimal assignment using greedy local search. Then, we randomly split one tracklet into two or merge two tracklets into one and search for the best assignment in the new tracklet graph. The new graph will be accepted with probability computed by the Metropolis criterion [31]. We repeat this procedure multiple times and return the best tracklet graph and assignment among all samples. Algorithm 1 summarizes the search algorithm.

This algorithm is guaranteed to produce a solution that satisfies mutual exclusion constraints. We also find that it leads to more accurate identification compared to that achieved by taking the maximum of marginals generated by loopy BP [26]. Unfortunately this method is slower, making it less suitable for using inside the E step.

Algorithm 1 MCMC search

$G^\star$ = initial tracklet graph
$\mathbf{z}^\star$ = GREEDYLOCALSEARCH($G^\star$)
for $\tau = 1$ to T **do**
  ρ = UNIFORMSAMPLE(0,1)
  if $\rho > 0.5$ **then**
    G^τ = randomly split one tracklet in $G^\star$
  else
    G^τ = randomly merge two traklets in $G^\star$
  end if
  $\mathbf{z}^\tau$ = GREEDYLOCALSEARCH(G^τ)
  $A = min\left(1, \frac{p(\mathbf{z}^\tau)}{p(\mathbf{z}^\star)}\right)$
  ρ = UNIFORMSAMPLE(0,1)
  if $\rho < A$ **then**
    $G^\star = G^\tau$, $\mathbf{z}^\star = \mathbf{z}^\tau$
  end if
end for
return $\mathbf{z}^{opt} = arg\,max\ p(\mathbf{z}^\tau)$

6. Results

6.1. NBA Basketball Videos

We used data from the 2010 NBA Championship series (Los Angeles Lakers vs. Boston Celtics). The video consists of different kind of shots (close-up, medium-distance, commercials), and we extracted clips from medium-distance shots. Although each team has 12 players, only 10 players from the Lakers and 9 players from the Celtics played. As a result, the number of player class labels for the Lakers and Celtics was 10 and 9, respectively. The maximum number of player detections in a frame is 10. Also, using the highly accurate team classifier in Section 4, we were able to separate detections into teams and perform identification for each team individually.

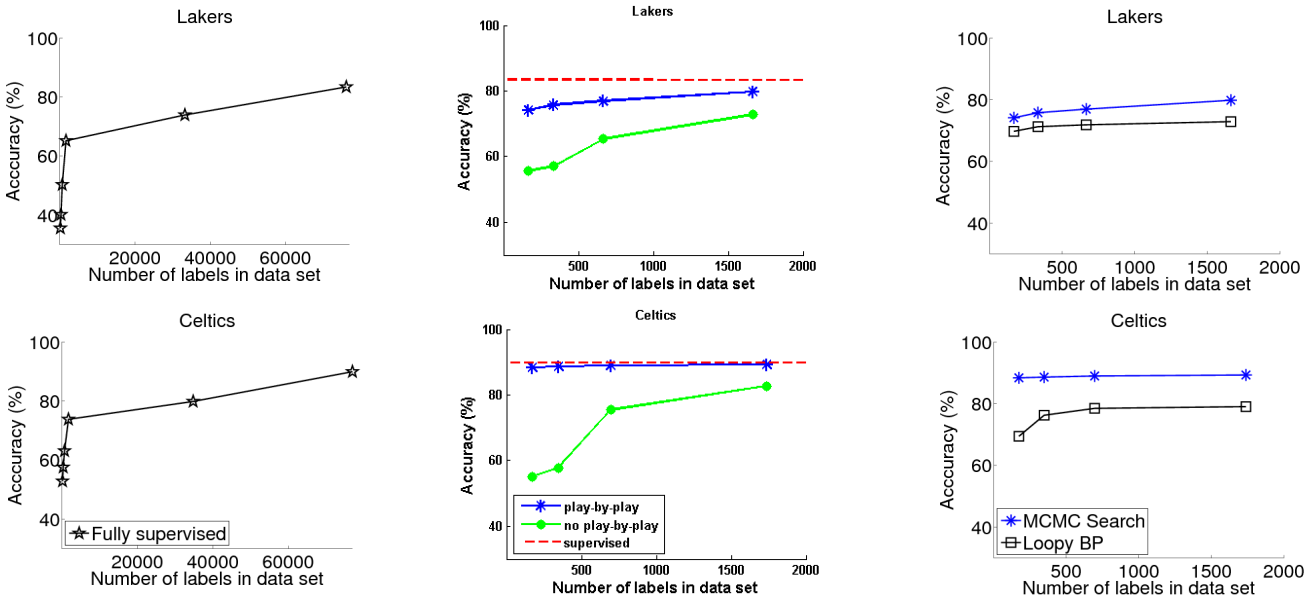
We used a labeled training set which is twice as large as [26]. Specifically, we use 153160 detections (77004 of Celtics players, 76156 of Lakers players) across 21306 frames. We evaluated the learned models on 15 test clips, consisting of a total of 9751 frames with 64174 detections (32227 of Celtics players, 31947 of Lakers players). The test clips varied in length, with the shortest at 300 frames and longest at 1400 frames. They also varied in level of identification difficulty. Labeling both training and test sets took us considerable effort, on the order of 200+ hrs. The size of this training data set is comparable or larger than others in the weakly labeled learning literature. For example, in previous work on high-resolution movies, [14] trained/tested on 49447 faces, and [11] trained on about 100000 faces.

6.2. Performance with supervised learning

Figure 4(a) shows player classifications, averaged over all test clips, for both teams when fully supervised learning is done, as a function of number of training set size. We can see that increasing training data improves the performance of a supervised model. Results for a fully unsupervised model were omitted since they were almost as bad as a randomly assigned solution ($\sim 20\%$).

Results are for $\epsilon = 0.001$ and do not change for values of $0 < \epsilon \leq 0.01$ (equivalent to assuming 1% error in tracking). Reported values were averaged over multiple learning trials (since labels were sampled randomly from the training data) and over multiple MCMC trials for search.

Interestingly, performance for the Celtics is better than



a) Supervised learning b) Semi-supervised learning c) Loopy BP vs. MCMC search

Figure 4. Average test accuracies for the Lakers and Celtics, as a function of number of labels in the training set. a) shows results of supervised learning; b) shows results of semi-supervised learning with and without play-by-play in the training set; c) shows results of using loopy BP vs. MCMC search for prediction (the model is learned in a semi-supervised way with play-by-play).

for the Lakers (90% vs 80%). Closer examination of the confusion matrix for the Lakers shows that one of the players, Lamar Odom (jersey 7) is often mistaken for his teammates Andrew Bynum (jersey 17) and Ron Artest (jersey 37). All three players not only have similar stature and appearance but also wear jersey numbers that, when not viewed entirely, could lead to mistaken identities.

6.3. Performance with semi-supervised learning

Figure 4(b) shows average player classification accuracies for semi-supervised learning, vs the number of fully labeled patches available in the training set. Since we were only interested in using a small number of labels with semi-supervised learning, we capped the number of labels considered to 2000. Given a set of labels, we compare the performance of semi-supervised learning in two scenarios: one where the remaining images have play-by-play, and another where no play-by-play is available.

We see that the weakly labeled data (play-by-play) improves the performance considerably. In fact, it seems that performance is almost invariant to the amount of labeled data. However, it turns out that performance of semi-supervised learning drops dramatically once the number of labeled examples goes below about 50 (result not shown). We conjecture that this is because it is hard to create a good initial estimate of the parameters for EM unless we have some labeled data.

Additionally, from Figures 4(a) and 4(b), we observe that semi-supervised learning with only 150 labels in a 75000-image training set outperforms a fully-supervised model learned on a 30000-image training set. We can conclude that adding more weakly labeled data helps to improve performance.

6.4. BP vs MCMC

In Figure 4(c), we compare identification accuracies of the MCMC search presented in this paper and of loopy BP (as used in [26]). We see that MCMC search consistently gives better accuracy than using loopy BP. The differences range from 5% to 15%, depending on the team and the number of labels used during learning. Unfortunately, MCMC search is about 10-20 times slower than loopy BP.

6.5. Qualitative results

Figure 5 shows tracking and identification results on the basketball sequences. We see that the proposed system is able to track and identify multiple basketball players effectively. Please refer to the video attachment for more details.

7. Conclusions and future work

In this paper, we introduce the use of semi-supervised learning to reduce the amount of labels needed during training. We use play-by-play information that is publicly available on websites to increase the number of videos used for

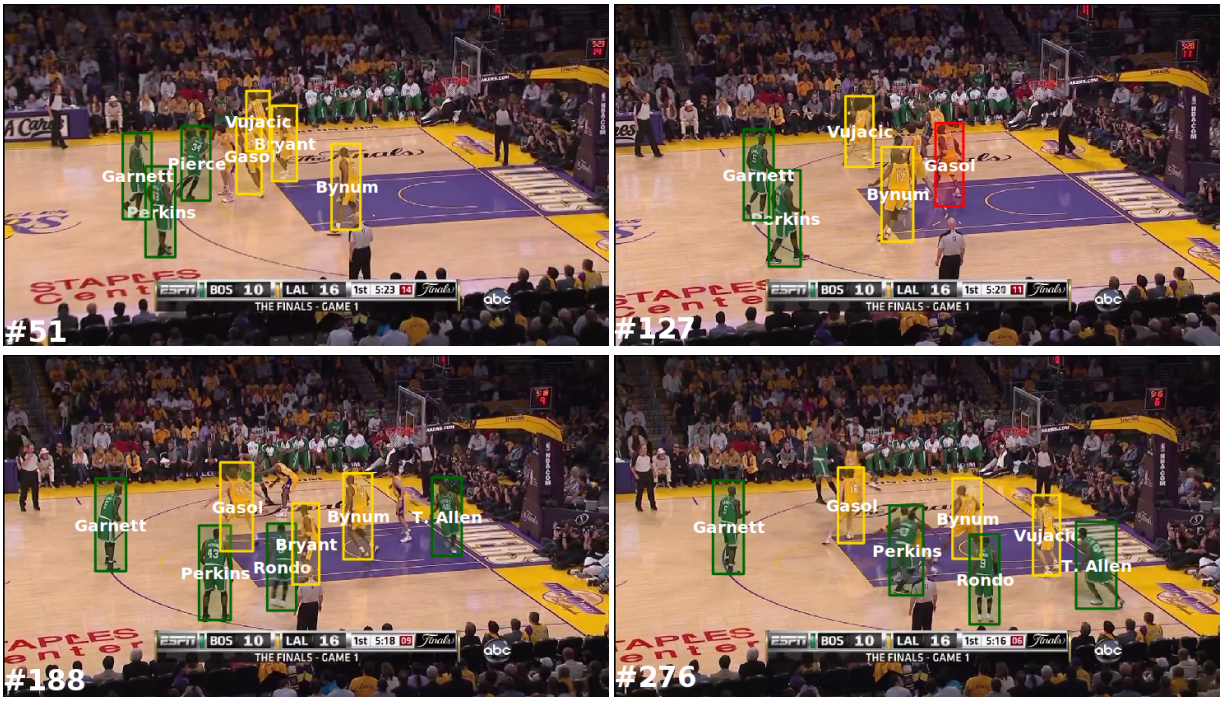


Figure 5. Automatic tracking and identification results in a broadcast basketball video. Green boxes represent Celtics players, and yellow boxes represent Lakers players. Text within boxes are identification results (player's name), while red boxes highlight misclassifications.

learning. We evaluate our system by training large-scale videos with play-by-play and demonstrate its effectiveness against models learned on smaller but fully labeled training data. We plan to scale up to even larger training sets, given the amount of unlabeled video available.

We hope to couple tracking and identification by including tracking-related information during MCMC search, to improve the player model by using player motion as additional features, and to incorporate court position in identification. With the help of semi-supervised learning and these additional improvements, we hope to apply the system to videos of other sports like hockey and soccer.

References

[1] http://www.nba.com. 2

[2] L. Ballan, M. Bertini, A. D. Bimbo, and W. Nunziati. Soccer players identification based on visual local features. In *CIVR*, 2007. 2

[3] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching Words and Pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003. 1

[4] M. Bertini, A. D. Bimbo, and W. Nunziati. Player Identification in Soccer Videos. In *MIR*, 2005. 2

[5] M. Bertini, A. D. Bimbo, and W. Nunziati. Automatic Detection of Player's Identity in Soccer Videos using Faces and Text Cues. In *ACM Multimedia*, 2006. 2

[6] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 3

[7] A. Blum, J. Lafferty, M. Rwebangira, and R. Reddy. Semi-supervised learning using randomized mincuts. In *In ICML*, 2004. 4

[8] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *In COLT*, 1998. 4

[9] Y. Cai, N. de Freitas, and J. J. Little. Robust Visual Tracking for Multiple Targets. In *ECCV*, 2006. 2, 3

[10] O. Chapelle, J. Weston, and B. Scholkopf. Cluster kernels for semi-supervised learning. In *In NIPS*, 2002. 4

[11] T. Cour, B. Sapp, C. Jordan, and B. Taskar. Learning from Ambiguously Labeled Images. In *CVPR*, 2009. 1, 2, 5

[12] T. Cour, B. Sapp, A. Nagle, and B. Taskar. Talking Pictures: Temporal Grouping and Dialog-Supervised Person Recognition. In *CVPR*, 2010. 1, 2

[13] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic Annotation of Human Actions in Video. In *ICCV*, 2009. 1

[14] M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is... Buffy" - Automatic Naming of Characters in TV Video. In *BMVC*, 2006. 1, 2, 5

[15] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person Re-Identification by Symmetry-Driven Accumulation of Local Features. In *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010. 2

[16] P. Felzenszwalb, D. McAllester, and D. Ramanan. A Discriminatively Trained, Multiscale, Deformable Part Model. In *CVPR*, 2008. 3

[17] W. Ge and R. T. Collins. Multi-target Data Association by Tracklets with Unsupervised Parameter Estimation. In *BMVC*, 2008. 2

[18] D. Gray and H. Tao. Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features. In *Proceedings of the 10th European Conference on Computer Vision*, volume 5302 of *LNCS*, pages 262–275, 2008. 2

[19] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding Videos, Constructing Plots: Learning a Visually Grounded Storyline Model from Annotated Videos. In *CVPR*, 2009. 1

[20] H. H. Hoos and T. Stützle. *Stochastic Local Search - Foundations Applications*. Morgan Kaufmann, 2004. 5

[21] O. Javed, K. Shafique, Z. Rasheed, and M. Shah. Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views. *Computer Vision and Image Understanding*, 109:146–162, 2008. 2

[22] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASMEJournal of Basic Engineering*, 82(Series D):35–45, 1960. 3

[23] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 1

[24] J. Liu, X. Tong, W. Li, T. Wang, Y. Zhang, and H. Wang. Automatic player detection, labeling and tracking in broadcast soccer video. *Pattern Recognition Letters*, 30:103–113, 2009. 2, 5

[25] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 3

[26] W.-L. Lu, J.-A. Ting, K. P. Murphy, and J. J. Little. Identifying Players in Broadcast Sports Videos using Conditional Random Fields. In *CVPR*, 2011. 1, 2, 3, 4, 5, 6

[27] M. Marszalek, I. Laptev, and C. Schmid. Actions in Context. In *CVPR*, 2009. 1

[28] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *BMVC*, 2002. 3

[29] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy Belief Propagation for Approximate Inference: An Empirical Study. In *UAI*, 1999. 4

[30] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe. A Boosted Particle Filter: Multitarget Detection and Tracking. In *ECCV*, 2004. 2, 3

[31] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2nd edition, 2005. 5

[32] M. Saric, H. Dujmic, V. Papic, and N. Rozic. Player Number Localization and Recognition in Soccer Video using HSV Color Space and Internal Contours. In *ICSIP*, 2008. 2

[33] J. Sivic, M. Everingham, and A. Zisserman. "Who are you" - Learning person specific classifiers from video. In *CVPR*, 2009. 1, 2

[34] Tesseract-OCR. http://code.google.com/p/tesseract-ocr/. 2

[35] C. Vondrick, D. Ramanan, and D. Patterson. Efficiently Scaling Up Video Annotation with Crowdsourced Marketplaces. In *ECCV*, 2010. 1

[36] Q. Ye, Q. Huang, S. Jiang, Y. Liu, and W. Gao. Jersey number detection in sports video for athlete identification. In *SPIE*, 2005. 2

[37] A. Yilmaz and O. Javed. Object Tracking: A Survey. *ACM Computing Surveys*, 38(4):No. 13, 2006. 2

[38] G. Zhu, C. Xu, Q. Huang, Y. Rui, S. Jiang, W. Gao, and H. Yao. Event Tactic Analysis Based on Broadcast Sports Video. *IEEE Transactions on Multimedia*, 11(1):49–66, 2009. 2