Action Attribute Detection from Sports Videos with Contextual Constraints

Xiaodong Yu[1]
xiaodong_yu@cable.comcast.com
Ching Lik Teo[2]
cteo@cs.umd.edu
Yezhou Yang[2]
yzyang@cs.umd.edu
Cornelia Fermüller[2]
fer@cfar.umd.edu
Yiannis Aloimonos[2]
yiannis@cs.umd.edu

[1] Comcast Corporation
Washington DC, USA
[2] University of Maryland,
College Park, MD, USA

Abstract

In this paper, we are interested in detecting action attributes from sports videos for event understanding and video analysis. Action attribute is a middle layer between low level motion features and high level action classes, which includes various motion patterns of human limbs and bodies and the interaction between human and objects. Successfully detecting action attributes provides a richer video description that facilitates many other important tasks, such action classification, video understanding, automatic video transcript, etc.

A naive approach to deal with this challenging problem is to train a classifier for each attribute and then use them to detect attributes in novel videos independently. However, this independence assumption is often too strong, and as we show in our experiments, produces a large number of false positives in practice. We propose a novel approach that incorporates the contextual constraints for activity attribute detection. The temporal contexts within an attribute and the co-occurrence contexts between different attributes are modelled by a factorial conditional random field, which encourages agreement between different time points and attributes. The effectiveness of our methods are clearly illustrated by the experimental evaluations.

1 Introduction

While traditional computer vision research focuses on categorizing objects, scenes and actions, more and more people are interested in going beyond naming these entities and devising approaches to infer attributes from them. The capacity to detect attributes allows us to describe a visual entity in greater detail; compare two visual entities at multiple levels, and helps us to categorize unseen visual classes [4, 6]. The attribute layer can be considered as an intermediate layer between the low-level features and action class labels, and this property can facilitate novel applications such as *zero-shot* learning and *one-shot* learning [4, 8, 16].

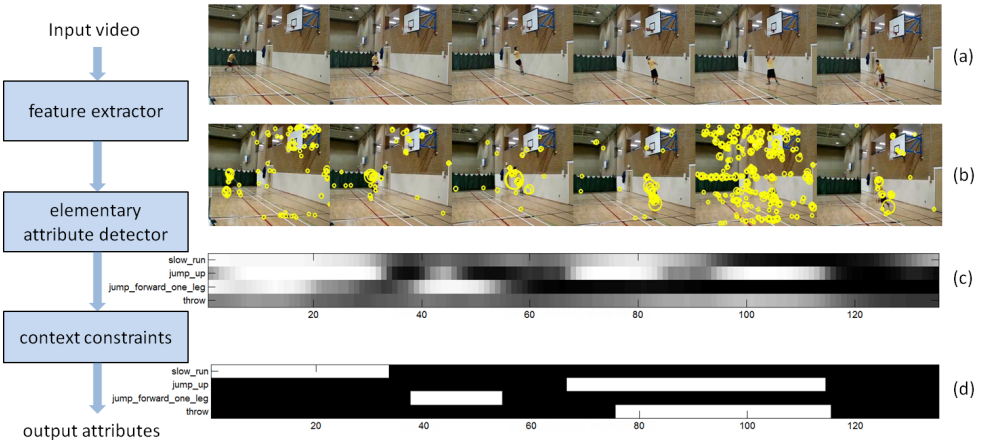


Figure 1: Overview of the proposed system including the key components (left) and example inputs/outputs (right): row (a) shows example video frames along the timeline; row (b) shows the detected STIP interest points; row (c) shows the probability output from the elementary attribute detector (the brighter the color, the higher the probability that corresponding attribute presents in that frame); row (d) shows the final attribute label after applying context constraints (bright color represents positive label and black represent negative). The frame numbers are illustrated on the bottom of row (c) and (d).

In this paper, we study the problem of detecting action attributes from sport videos. Action attributes include atomic components of action classes (such as the motion patterns of human limbs and body), contextual components of action classes (such as the objects and scenes involved in the action), and non-semantic attributes, a.k.a data-driven attributes [9]. A common property of action attributes is that they can be generalized into different action classes. This is especially true in sports videos. For example, *bend*, as an action attribute that describes the motion of human body, is present in the action `tennis serve`, `bowling`, `snatch`, etc. That is why they can be learned even from training sets that contain only a few examples for each action class (*one-shot* learning) or even no example for some action classes (*zero-shot* learning). We focus on the action attributes related to motion patterns of human body in this paper; however, our model can be easily extended to detect the other types of action attributes as well.

The concept of action attribute was first introduced in [9]. However, the approach proposed in [9] has several severe limitations that restrict the applicability of action attributes. One of the most noticeable problems is that action attributes are labelled at the level of an action class, instead of being labelled at the level of a frame or at the level of a video. As a result, all videos from the same action class are assumed to have the same set of action attributes, regardless of the exact content of a specific video, and the temporal structure of the action attributes is totally discarded. But in reality, not every video belonging to the same action class have the same set of action attributes; more often than not, real-world videos would have some exceptions. For example, in a video of the `snatch` activity, the athlete may not be able to completely lift the barbell above his head at the end so we cannot say this video has an action attribute *two arms raise pose*. Furthermore, the temporal structure is a unique property of action attributes and we lose lots of descriptive capacity if we ignore it.

For example, given a video of `basketball layup`, a description *"the athlete starts with a slow run and lasts for half a second, then jumps forward with single leg in the next second, finally jumps up and throws the ball (into the basket), and maintains a slow run at the end of the video"* will be more useful than simply saying *"there are slow running, jumping forward, jumping up, throwing in this video"*. The goal of this paper is thus to detect the key action attributes at each frame from a given video so that we can generate video descriptions at a much finer granularity than those from previous work. Figure 1 shows an example of the action attributes proposed in this paper for a particular activity (see row (d)).

While having great advantages as discussed above, it is obviously a much more challenging task to locate the temporal occurrences of every action attributes in a given video. As a high-level semantic concept, a particular action attribute may exhibit significant variability due to viewpoint changes, photometric measurements, intra-class variability (e.g. attribute *two arms open* can have different opening angles between two arms, *jump up* can have different height and velocity, etc). Naive detectors that rely entirely on local features will easily be overwhelmed by a large number of false positives and/or false negatives. So we must take into account the contextual constraints in both the temporal and semantic domains, thereby reducing the noise in the local feature space to produce more reliable results. This is exactly the theme of this paper.

2 Detecting Action Attributes using Contextual Constraints

2.1 Systematic Overview

A systematic overview of our approach is shown in Figure 1. It is composed of three parts: feature extractor, which extracts low-level features from a video; elementary attribute detector, which detects attributes in a video using local cues only; and context constraints, which combine outputs from the elementary detectors of all attributes so as to determine a set of globally optimized attribute labels. We use off-the-shelf algorithms in the first two parts and a factorial conditional random field to incorporate the contextual constraints in the last part. The details are presented in the rest of this section.

2.2 Low-level feature extraction

Since our goal is to detect human action attributes, we need a module to detect human and extract motion features within the human bounding box. This problem has been thoroughly studied and numerous algorithms have been proposed in the last few decades. Interested readers can refer to [1, 2, 3, 5, 10] for a few exemplar implementations. Since this is beyond the scope of this paper, we assume that this step has been done and simply use the annotated bounding box as the one produced by any algorithm that does human detection and tracking. In this way, we can measure the upper bound on the performance of various attribute detectors described in the next section.

The low-level features for representing motions in a video are HOG and HOF, which are extracted at the detected interest points using the author's latest implementation of STIP [13] with a default noise threshold and video resolution of 320×240. All descriptors within the human bounding box are quantized into one of the 400 visual words, which are computed using k-means from 40,000 randomly selected descriptors. At the end, each frame is

represented by a histogram of visual words, $\mathbf{x} \in \mathbb{X}^d$, which counts the number of quantized descriptors within the bounding box of the human in a frame and its two closest neighbours.

2.3 Elementary attribute detectors

Our elementary attribute detector is an SVM with a χ^2 kernel, which takes the histogram of visual words as inputs and predicts the probability of a particular action attribute occurring in each frame of an unseen video. The kernel function $K(\mathbf{h}_i, \mathbf{h}_j)$ is given as in Section 3.4 of [17]

During training, we have training videos with attributes labelled at each frame. For each attribute, we select features from all positive frames as positive examples and randomly select an equal number of negative examples to train the SVM. For testing, we first extract histogram of visual words from each frame t in an unseen video, denoted as $\mathbf{x}_t$. Then the SVM for attribute a predicts the presence of this attribute in frame t, and the output probability value is denoted as $N_a(\mathbf{x}_t)$. This value will be used as the node feature of the conditional random field described in the next section. The SVM also predicts a binary value to indicate the presence/absence of attribute a at frame t by applying a threshold of 0.5 over $N_a(\mathbf{x}_t)$. This will be used as a baseline in our evaluation. See Section 3.2 for more details.

2.4 Incorporating Contextual Constraints

2.4.1 Factorial Conditional Random Field Model

We take into account two types of contextual constraints in this paper: the temporal context and the semantic context. To promote agreement between the different attribute labels at different frames, we model the contextual constraints with a conditional random field (CRF) as shown in Figure 2(b). The features extracted from T frames in a video are denoted as a vector of T local observations, $\mathbf{X} = \{x_1, x_2, ..., x_T\}$; and at each local observation at frame t, by $\mathbf{x}_t$, the histogram of visual words as discussed in the preceding section. At each frame t, we wish to detect the presence of A action attributes $\mathbf{y}_t = \{y_t^1, y_t^2, ..., y_t^A\}$, which are the states in the CRF model. In the literature, this is also known as a factorial CRF [14]. To better understand this CRF model, we can compare it with a linear chain CRF [7] as shown in Figure 2(a), which has been used in action class recognition from video streams [13]. In a linear chain CRF, the state of each time point is dependent on its immediate neighbors only (Markovian assumption) and we enforce agreement between states in adjacent time points after accounting for the correlation between the neighboring temporal states. In a factorial CRF, we have multiple states at each time point, $\mathbf{y}_t = \{y_t^1, y_t^2, ..., y_t^A\}$, and there are edges between every pair of y_t^i and y_t^j, $(i, j) \in \{1, 2, ..., A\}$[1]. To avoid clutter, we only show two attributes at each time point in Figure 2(b). In the experimental dataset used, there are 24 attributes at each time point. The between-chain edges are designed to promote agreement between different attributes at the same time point. The intuition is that some attributes tend to occur together, e.g.*two arms oscillate* and *fast run* while others don't, e.g. *slow run* and *fast run*. Thus the between-chain edges take into account the co-temporal correlation among attributes.

[1]For clarity, we call the edges between states of the same time points as *between-chain edges* and the edges between states of adjacent time points as *within-chain edges*. In Figure 2(b), the former are colored in red and the latter in black

The factorial CRF is defined as follows

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})} \left(\sum_{t=1}^{T} \sum_{a=1}^{A} \Upsilon_t(y_{t,a}, \mathbf{X}) \right) \left(\sum_{t=1}^{T-1} \sum_{a=1}^{A} \Psi_t(y_{t,a}, y_{t+1,a}, \mathbf{X}) \right) \left(\sum_{t=1}^{T} \sum_{a,b \in \{1,\ldots,A\}, a \neq b} \Phi_t(y_{t,a}, y_{t,b}, \mathbf{X}) \right), \quad (1)$$

where $Z(\mathbf{X})$ is the partition function, $\{\Upsilon_t\}$ the local (node) potential functions, $\{\Psi_t\}$ the within-chain potential functions, and $\{\Phi_t\}$ the between-chain potential functions. The potential functions are defined by a set of feature functions $\{f_k\}$ together with corresponding weights $\{\lambda_k\}$ as:

$$\begin{aligned} \Upsilon_t(y_{t,a}, \mathbf{X}) &= \exp\left(\sum_k \lambda_k f_k(y_{t,a}, \mathbf{X}) \right) \\ \Psi_t(y_{t,a}, y_{t+1,a}, \mathbf{X}) &= \exp\left(\sum_k \lambda_k f_k(y_{t,a}, y_{t+1,a}, \mathbf{X}) \right) \\ \Phi_t(y_{t,a}, y_{t,b}, \mathbf{X}) &= \exp\left(\sum_k \lambda_k f_k(y_{t,a}, y_{t,b}, \mathbf{X}) \right). \end{aligned} \quad (2)$$

The feature functions for the above potential functions are defined as follows:

$$\begin{aligned} f_k(y_{t,a}, \mathbf{X}) &= \mathbb{I}[y_{t,a} = m] \log N_a(\mathbf{x}_{t-j}) \\ f_k(y_{t,a}, y_{t+1,a}, \mathbf{X}) &= \mathbb{I}[y_{t,a} = m \wedge y_{t+1,a} = n] \\ f_k(y_{t,a}, y_{t,b}, \mathbf{X}) &= \mathbb{I}[y_{t,a} = m \wedge y_{t,b} = n] \phi(a,b) \end{aligned} \quad (3)$$

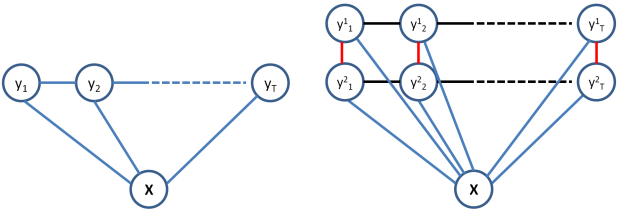
where $j \in [-W, W]$[2], $m, n \in \{0,1\}, a, b \in \{1,\ldots,A\}$, $\mathbb{I}[x = A]$ is an indicator function so that $\mathbb{I}[x = A] = 1$ if $x = A$, or 0 otherwise. Intuitively, the node feature functions encode correlations from the presence of attribute a and the observation confidence forward or backward within a time window; the within-chain edge feature functions encode the transition probability between adjacent time point for attribute a; the between-chain edge feature functions encode the compatibility of two attributes, a and b, measured by $\phi(a,b)$ and is obtained by normalizing the co-occurrence frequency between these two attributes.

2.4.2 Learning Model Parameters

In general, given a training set consisting of N sequences $\mathcal{D} = \{\mathbf{X}^n, \mathbf{Y}^n\}_{i=1}^N$, we want to find the optimal parameter $\Lambda^* = \{\lambda_k^*\}$ that maximizes the following objective function, as discussed in [7],

$$L(\Lambda) = \sum_{n=1}^{N} \log P(\mathbf{Y}^n | \mathbf{X}^n, \Lambda), \quad (4)$$

[2] where $W = 1$ is the size of the sliding window around a video frame



(a) (b)

Figure 2: (a) A linear chain CRF model and (b) a factorial CRF model. To avoid clutter, we only show two attributes at each time points. In reality, we can have as many as attributes at each time points and they are fully connected to each other.

where the right hand side is the conditional log-likelihood of the training data. The partial derivative of the log-likelihood w.r.t λ_k associated with clique index c is [14]

$$\frac{\partial L}{\partial \lambda_k} = \sum_n \sum_t f_k(\mathbf{Y}^n_{t,c}, \mathbf{X}^n) - \sum_n \sum_t \sum_{\mathbf{Y}_{t,c}} p(\mathbf{Y}_{t,c}|\mathbf{X}^n) f_k(\mathbf{Y}^n_{t,c}\mathbf{X}^n) \tag{5}$$

where $\mathbf{Y}^n_{t,c}$ is the assignment to $\mathbf{Y}_{t,c}$ in $\mathbf{Y}^n$, and $\mathbf{Y}_{t,c}$ ranges over assignments to the clique c at time point t. The first term in the right hand side is easy to compute. The second term computes marginal probabilities $p(\mathbf{Y}_{t,c}|\mathbf{X}^n)$, which will be discussed in Section 2.4.3.

To reduce overfitting, we define a spherical Gaussian prior [14] to the parameter, which has zero mean and covariance matrix $\Sigma = \sigma^2\mathbf{I}$, i.e.,

$$\Lambda \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}), \tag{6}$$

and this is equivalent to optimizing $L(\Lambda)$ using ℓ^2 regularization:

$$L_r(\Lambda) = \sum_{n=1}^{N} \log P(\mathbf{Y}^n|\mathbf{X}^n, \Lambda) - \frac{1}{2\sigma^2} \parallel \Lambda \parallel^2, \tag{7}$$

and the gradient becomes

$$\frac{\partial L_r(\Lambda)}{\partial \lambda_k} = \frac{\partial L}{\partial \lambda_k} - \frac{\lambda_k}{\sigma^2} \tag{8}$$

To speed up the training process, we use stochastic gradient ascent to search for the optimal parameters [12]. At each iteration, we randomly pick a training sequence, evaluate the gradient w.r.t λ_k on that training sequence, and update λ_k by taking a small step in the direction of the negative gradient

$$\lambda_k \leftarrow \lambda_k + \alpha\Big(\sum_t f_k(\mathbf{Y}^n_{t,c}, \mathbf{X}^n) - \sum_t \sum_{\mathbf{Y}_{t,c}} p(\mathbf{Y}_{t,c}|\mathbf{X}^n) f_k(\mathbf{Y}^n_{t,c}, \mathbf{X}^n) - \frac{\lambda_k}{\sigma^2}\Big), \tag{9}$$

where α is a learning rate parameter, which is set to a small value. The iteration continues until we reach the maximum iteration or the change of objective function is below a threshold for 10 iterations. In our experiments, the optimization procedure usually completes within $10N$ iterations.

2.4.3 Inference

Two types of inference tasks are addressed during training and testing. In training, we need to compute the marginal probability of each clique $p(\mathbf{Y}_{t,c}|\mathbf{X}^n)$. In testing, we need to perform Viterbi decoding, i.e. estimate the most probable attribute sequence $\mathbf{Y}^*$ for an unseen sequence $\mathbf{X}$ that maximizes the conditional probability

$$\mathbf{Y}^* = arg\max_{\mathbf{Y}} P(\mathbf{Y}|\mathbf{X}, \Lambda^*) \tag{10}$$

where the parameter $\Lambda^* = \{\lambda_k\}$ are learned from training examples. Both tasks can be achieved by Loopy Belief Propagation (LBP) [14]. In this section, we briefly discuss the LBP procedure for computing the marginal probability. The Viterbi decoding can be performed in a similar fashion by replacing the summation in Equation (11) with a maximization.

Belief propagation algorithms iteratively update a vector $\mathbf{m} = (m_i(v_j))$, which are called messages between pair of vertices v_i and v_j. The message $m_i(v_j)$ sent from vertex v_i to its neighbor v_j is given by:

$$m_i(v_j) \leftarrow \sum_i \left(\Upsilon(v_i)\Omega(v_i, v_j) \prod_{k \neq j} m_k(v_i) \right), \tag{11}$$

where $\Upsilon(v_i)$ is the local potential, $\Omega(v_i, v_j)$ the edge potential between v_i and v_j, $m_k(v_i)$ is the message sent to v_i from its neighbors except v_j. A random schedule is adopted and messages propagate through the CRF until convergence or a maximum number of iterations is reached. Then the marginal probability of nodes v_i and v_j are computed as:

$$p(v_i, v_j) \propto \Upsilon(v_i)\Upsilon(v_j)\Omega(v_i, v_j) \prod_{k \neq j} m_k(v_i) \prod_{k \neq i} m_k(v_j). \tag{12}$$

Note that we have omitted $\mathbf{X}$ in the above two equations for clarity, and we use $\Omega(v_i, v_j)$ to refer either $\Psi_t(y_{t,a}, y_{t+1,a}, \mathbf{X})$ or $\Phi_t(y_{t,a}, y_{t,b}, \mathbf{X})$, which can be determined by the clique that involves v_i and v_j.

3 Experiments

3.1 Dataset and action attributes

We tested our approach on the Olympic Sports Dataset [11]. This dataset includes 16 action classes and 783 videos. The original purpose of this dataset is for recognizing action classes and thus there is no attribute labels available. Liu et al [9] defined 39 attributes on this dataset on the action class level, i.e., each action class has a list of fixed attributes across all videos of this class. As we argued in Section 1, this level of action attribute labels is neither sufficient to describe the dynamics in the action videos nor capture the unique characteristics within a particular video. Thus we create a new dataset to evaluate the performance of action attribute detection using the videos from the Olympic Sport Dataset. In particular, we defined 24 action attributes, which include 9 leg motion patterns, 6 arm motion patterns, 6 whole body motion patterns, and 3 human-object interactions. For each action class, we randomly select 20 videos from the Olympic Sports Dataset to create the Action with Attribute dataset with 320 videos. In each video, we labelled the presence/absence of each action attribute as well as the bounding box of the athlete in each frame.

To evaluate the performance of action attribute detector, we divide the Action with Attribute dataset into two disjoint subsets, which include a training set with 240 videos and a testing set with 80 videos. Both sets include all 16 action classes.

3.2 Baseline algorithms

The first baseline algorithm is the elementary attribute detectors described in Section 2.3. This algorithm treats each frame as an independent sample and does not take into account any contextual constraints.

The second baseline algorithm is a linear CRF as illustrated in Figure 2(a). The node feature functions and edge feature functions are defined in Equation (3). The learning algorithm is the same as the one presented in Section 2.4.2, except that there are no between-chain feature functions involved. Since this is a chain structured CRF, exact inference can be achieved by the forward-backward algorithm and Viterbi algorithm [14]. This algorithm takes into account the temporal contextual constraints but treats each attribute independently: the semantic context is still ignored.

3.3 Experimental Results

We measure the performance of attribute detection by precision, recall and F1-score at the frame level. Every frame is treated as an individual sample for this purpose. A positive sample that is correctly (incorrectly) detected as positive (negative) is counted as true positive, a.k.a., TP, (false negative, a.k.a., FN); A negative sample that is correctly (incorrectly) detected as negative (positive) is counted as true negative, a.k.a., TN, (false positive, a.k.a., FP).

We compute the three measurements for each attribute and their mean and standard deviation for the baselines and the proposed factorial CRF (FCRF). Their overall detection performances are summarized in Figure 3(a). This figure shows that the proposed FCRF significantly outperforms the baseline algorithms SVM and LCRF in precision and F1-score, while the recall of all three algorithms are similar. To test the significance of these performance results, we did a t-test of the null hypothesis that the mean performances of FCRF are the same as those of SVM and LCRF, against the alternative that the mean performances of FCRF are higher than those of SVM and LCRF, with significance $\alpha = 0.05$. The result of t-test failed to reject the null hypothesis for recall but rejected the null hypothesis for precision and F1-score, with p-value < 0.0005.

Figure 3(b) compares the precision scores across each attribute. It shows that FCRF improves the detection precision for every attribute. In particular, the baseline algorithms achieve very low precision for a few attributes, e.g. *walk forward*, *stand up*, *one arm open*, *put down* and *throw*. FCRF removes a large number false positives using contextual constraints which results in significantly better precision scores.

Figure 4 illustrates the outputs of the three algorithms on a video of the activity `tennis serve`. It shows that the decision values of SVM are quite noisy and there are a large number of false positives for the attribute *bend* and false negatives for the attribute *one arm swing*. The LCRF, which only considers temporal constraints, is only able to smooth out noisy detections that last for short periods but is unable to deal with persistent false positives. On the other hand, FCRF corrects many of such errors by using semantic contextual constraints: *stand still* and *bend* are less likely to co-occur while *bend* and *one arm swing* are more likely to co-occur.

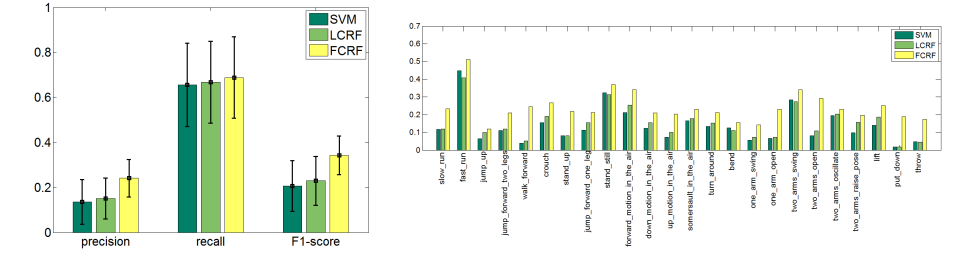


Figure 3: (a) Overall performance of action attribute detection and (b) precision of action attribute detection with three algorithms with three algorithms: SVM, linear CRF (LCRF) and factorial CRF (FCRF)

4 Conclusion and Future Work

In this paper, we study the problem of detecting action attributes from sport videos. Although the concept of action attributes has been discussed in previous work, to our knowledge, there does not exist an action attribute detection method that report presence/absence of attributes at the frame level. Our work not only answers the question "is there an action attribute *a* in this video?", but also addresses the question "when does this attribute occur?". The attribute annotation we propose at this granularity will benefit lots of interesting applications in video understanding and event detection. In this work, we proposed an approach that uses contextual constraints as post-processing to an off-the-shelf discriminative model for attribute detection. We observed that semantic context, i.e. the co-occurrence of attributes is an important constraint that can compensates for the ambiguity that arises from noisy video features. As a result, our approach is able to produce attribute labels that maximizes the agreement among labels at different time points and different attributes.

In our ongoing work, we are exploring more action attributes, such as the scene (e.g. *swimming pool*, *track*) and objects (e.g. *basketball/tennis ball*, *pole/javelin*). We will also incorporate more contextual constraints into our model. In particular, we are interested in the absolute and relative temporal order of attributes. An example for the absolute temporal order is in `high jump`: the athlete must first do a *slow run*, then *jump forward with one leg*, followed by *jumping up* and *moving up in the air*, and finally *move down in the air*; an example for the relative temporal order for `high jump` is that *slow run* must occur before the other attributes, *moving up in the air* must occur before *move down in the air*, etc. We believe these two types of temporal orders are both strong contextual cues that will facilitate the detection of action attributes more accurately.

References

[1] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting People Using Mutually Consistent Poselet Activations. In *European Conference on Computer Vision (ECCV)*, 2010.

[2] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection.

Figure 4: Detailed detection results of the activity `tennis serve`, where imposing contextual constraints in the FCRF improves the attribute detection precision compared to the other two baseline approaches: SVM and LCRF.

In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *CVPR*, volume 2, pages 886–893, June 2005.

[3] Liam Ellis, Nicholas Dowson, Jiri Matas, and Richard Bowden. Linear Regression and Adaptive Appearance Models for Fast Simultaneous Modelling and Tracking. *IJCV*, 95:154–179, 2011.

[4] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing Objects by Their Attributes. In *CVPR*, 2009.

[5] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32, 2010.

[6] V. Ferrari and A. Zisserman. Learning Visual Attributes. In *NIPS*, December 2007.

[7] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[8] H. Nickisch Lampert, C. H. and S. Harmeling. Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer. In *CVPR*, 2009.

[9] Jingen Liu, Benjamin Kuipers, and Silvio Savarese. Recognizing Human Actions by Attributes. In *CVPR*, 2011.

[10] S. M. Shahed Nejhum, Jeffrey Ho, , and Ming-Hsuan Yang. Online visual tracking with histograms and articulating blocks. *Computer Vision and Image Understanding (CVIU)*, 114:901–914, 2010.

[11] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification. In *ECCV*, 2010.

[12] Vishwanathan Schraudolph and Schmidt Murphy. Accelerated Training of Conditional Random Fields with Stochastic Gradient Methods. In *ICML*, 2006.

[13] Cristian Sminchisescu, Atul Kanaujia, and Dimitris Metaxas. Conditional Models for Contextual Human Motion Recognition. *CVIU*, 104:210–220, 2006.

[14] Charles Sutton, Andrew McCallum, and Khashayar Rohanimanesh. Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data. *Journal of Machine Learning Research*, 8:693–723, 2007.

[15] Heng Wang, Muhammad Muneeb Ullah, Alexander Kläser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[16] Xiaodong Yu and Yiannis Aloimonos. Attribute-based Transfer Learning for Object Categorization with Zero or One Training Example. In *ECCV*, 2010.

[17] Jianguo Zhang, Marcin Marszalek, Svetlana Lazebnik, and Cordelia Schmid. Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *IJCV*, 73(2):213–238, 2007.