HUMAN ACTION RECOGNITION BASED ON BAG OF FEATURES AND MULTI-VIEW NEURAL NETWORKS

Alexandros Iosifidis, Anastasios Tefas and Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
Email: {tefas,pitas}@aiia.csd.auth.gr

ABSTRACT

In this paper, we employ Single-hidden Layer Feedforward Neural networks in order to perform human action recognition based on multiple action representations. In order to determine both optimized network and action representation combination weights, we propose an optimization process that jointly minimizes the overall network training error and the within-class variance of the training data in the corresponding hidden layer spaces. The proposed approach has been evaluated by using the state-of-the-art Bag of Features-based action video representation on three publicly available action recognition databases, where it outperforms two commonly used video representation combination approaches, as well as the best single-descriptor classification outcome.

Index Terms— Single-hidden Layer Feedforward Neural networks, Multi-view Learning, Human Action Recognition, Bag of Features

1. INTRODUCTION

Human action recognition is intensively studied nowadays due to its importance in many real-life applications, like intelligent visual surveillance, human-computer interaction and video games, to name a few. Perhaps the most well studied and successful approach for action representation is based on the Bag of Visual Features (BoF) model. According to this model, sets of shape and/or motion descriptors are evaluated on spatiotemporal locations of interest of a video and multiple (one for each descriptor type) video representations are obtained by applying vector quantization. The descriptors that provide the current state-of-the-art performance in most action recognition databases are: the Histogram of Oriented Gradients (HOG), the Histogram of Optical Flow (HOF) and the Motion Boundary Histogram (MBH). These descriptors are evaluated on the trajectories of densely sampled video frame interest points, which are tracked for a number of consecutive video frames. Tracking can be performed in various ways [1, 2, 3]. The normalized location of the tracked interest points is also employed in order to form another descriptor type, referred to as Trajectory [4]. An advantage of this approach is that no human silhouette extraction is needed [5, 6].

Since different descriptor types express different properties of interest for actions, it is not surprising the fact that a combined action representation exploiting all the above mentioned (single-descriptor based) video representations enhances action classification performance [4]. Such combined action representations are usually obtained by employing unsupervised combination schemes, like the use of concatenated representations (either on the descriptor, or on the video representation level), or by combining the outcomes of classifiers trained on different representation types, e.g., by using the mean classifier outcome in the case of SLFN networks [7, 8]. However, the adoption of such combination approaches may decrease the generalization ability of the adopted classification schemes, since all the available action representations equally contribute to the classification result.

Extreme Learning Machine (ELM) [9] is a relatively new algorithm for fast Single-hidden Layer Feedforward Neural (SLFN) networks training, requiring low human supervision. Conventional SLFN training algorithms require adjustment of the network weights and the bias values, using a parameter optimization approach, like gradient descent. However, gradient descent learning techniques are, generally, slow and may lead to local minima. In ELM, the input weights and the hidden layer bias values are randomly chosen, while the network output weights are analytically calculated. ELM not only tends to reach a small training error, but also a small norm of output weights, indicating good generalization performance [10]. ELM has been successfully applied to many classification problems, including human action recognition [11, 12, 13, 14, 15, 16].

In this paper we employ the ELM algorithm in order to perform human action recognition from videos. We adopt the state-of-the-art BoF-based action representation [4], in order to represent videos depicting actions, called action videos hereafter. In order to enhance the performance of the ELM network and properly combine the information provided by different descriptor types, we extend the ELM algorithm in order to incorporate multiple video representations in the corre-

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 316564 (IMPART).

sponding ELM spaces and jointly minimize their within-class variance and the overall network training error for network output weights optimization.

The remainder of the paper is structured as follows. In Section 2, we briefly describe the ELM algorithm. The proposed optimization scheme is described in Section 3. Experimental results evaluating its performance are illustrated in Section 4. Finally, conclusions are drawn in Section 5.

2. EXTREME LEARNING MACHINE

ELM has been proposed for single-view classification [9]. Let $\mathbf{x}_i$ and $c_i,\ i = 1, ..., N$ be a set of the labeled action vectors (e.g., BoF-based action video representations) and the corresponding action class labels, respectively. For a classification problem involving the D-dimensional action vectors $\mathbf{x}_i$, each belonging to one of the C action classes, the network should consist of D input, H hidden and C output neurons. The network target vectors $\mathbf{t}_i = [t_{i1}, ..., t_{iC}]^T$, each corresponding to one labeled action vector $\mathbf{x}_i$, are set to $t_{ij} = 1$ for vectors belonging to action class j, i.e., when $c_i = j$, and to $t_{ij} = -1$ otherwise.

In ELM, the network input weights $\mathbf{W}_{in} \in \mathbb{R}^{D \times H}$ and the hidden layer bias values $\mathbf{b} \in \mathbb{R}^H$ are randomly chosen, while the output weights $\mathbf{W}_{out} \in \mathbb{R}^{H \times C}$ are analytically calculated. Let $\mathbf{v}_j$ denote the j-th column of $\mathbf{W}_{in}$, $\mathbf{u}_k$ the k-th row of $\mathbf{W}_{out}$ and u_{kj} be the j-th element of $\mathbf{u}_k$. For a given hidden layer activation function $\Phi(\cdot)$ and by using a linear activation function for the output neurons, the output $\mathbf{o}_i = [o_1, \ldots, o_C]^T$ of the ELM network corresponding to training action vector $\mathbf{s}_i$ is given by:

$$o_{ik} = \sum_{j=1}^{H} u_{kj}\ \Phi(\mathbf{v}_j, b_j, \mathbf{x}_i), \quad k = 1, ..., C. \tag{1}$$

By storing the hidden layer neuron outputs $\boldsymbol{\phi}_i \in \mathbb{R}^H$ in a matrix $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N]$, equation (1) can be written in a matrix form as $\mathbf{O} = \mathbf{W}_{out}^T \mathbf{\Phi}$. Finally, by assuming that the predicted network outputs $\mathbf{O}$ are equal to the desired ones, i.e., $\mathbf{o}_i = \mathbf{t}_i,\ i = 1, ..., N$, $\mathbf{W}_{out}$ can be analytically calculated by solving for $\mathbf{W}_{out}^T \mathbf{\Phi} = \mathbf{T}$, where $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]$ is a matrix containing the network target vectors. The network output weights are, thus, given by $\mathbf{W}_{out} = \left(\mathbf{\Phi}\mathbf{\Phi}^T\right)^{-1} \mathbf{\Phi}\,\mathbf{T}^T$.

The original ELM algorithm described above assumes zero training error. An extension allowing small training errors and incorporating the within-class variance of the training data in the ELM space has been proposed in [14], where the network output weights are obtained, according to a regularization paramter $c > 0$, by:

$$\mathbf{W}_{out} = \left(\mathbf{\Phi}\mathbf{\Phi}^T + \frac{1}{c}\mathbf{S}_w\right)^{-1} \mathbf{\Phi}\,\mathbf{T}^T. \tag{2}$$

$\mathbf{S}_w$ is the within class scatter matrix evaluated on $\boldsymbol{\phi}_i$.

After calculating the network output weights $\mathbf{W}_{out}$, a test action vector $\mathbf{x}_t$ can be introduced to the trained network and be classified to the action class corresponding to the maximal network output, i.e. $c_t = arg \max_j o_{tj},\ j = 1, ..., C$.

The above described ELM algorithms can be employed for single-view (i.e., single-representation) action classification. In the next section, we describe an optimization process that can be used for multi-view action classification, i.e., in the cases where each action video is represented by multiple action vectors $\mathbf{x}_i^v,\ v = 1, \ldots, V$.

3. PROPOSED OPTIMIZATION SCHEME

Let us assume that the N training action videos are represented by the corresponding action vectors $\mathbf{x}_i^v \in \mathbb{R}^{D_v},\ i = 1, \ldots, N,\ v = 1, \ldots, V$. We would like to employ them, in order to train V SLFN networks, each operating on one view (descriptor type). To this end we map the action vectors of each view v to one ELM space $\mathbb{R}^{H_v}$, by using randomly chosen input weights $\mathbf{W}_{in}^v \in \mathbb{R}^{D_v \times H_v}$ and input layer bias values $\mathbf{b}^v \in \mathbb{R}^{H_v}$. H_v is the dimensionality of the ELM space related to view v and may vary between views.

In order to determine both the networks output weights $\mathbf{W}_{out}^v \in \mathbb{R}^{H_v \times C}$ and appropriate descriptor type combination weights $\boldsymbol{\alpha} \in \mathbb{R}^V$ we can formulate the following optimization problem:

$$\textbf{Minimize:} \quad \mathcal{J} = \frac{1}{2}\sum_{v=1}^{V} \|\mathbf{S}_w^{v\frac{1}{2}} \mathbf{W}_{out}^v\|_F^2 + \frac{c}{2}\sum_{i=1}^{N} \|\boldsymbol{\xi}_i\|_2^2 \tag{3}$$

$$\textbf{s.t.:} \quad \left(\sum_{v=1}^{V} \alpha_v \mathbf{W}_{out}^{v\,T} \boldsymbol{\phi}_i^v\right) - \mathbf{t}_i = \boldsymbol{\xi}_i,\ i = 1, ..., N, \tag{4}$$

$$\|\boldsymbol{\alpha}\|_2^2 = 1, \tag{5}$$

where $\boldsymbol{\phi}_i^v \in \mathbb{R}^{H_v}$ is the representation of $\mathbf{x}_i^v$ in the corresponding ELM space and $\mathbf{S}_w^v$ is the within-class scatter matrix of the training data evaluated on $\boldsymbol{\phi}_i^v$. $\boldsymbol{\xi}_i \in \mathbb{R}^C$ is the error vector related to the i-th action video and c is a regularization parameter expressing the importance of the training error in the optimization process. By using $\mathbf{\Phi}^v = [\boldsymbol{\phi}_1^v, \ldots, \boldsymbol{\phi}_N^v]$, the network responses corresponding to the entire training set are given by $\mathbf{O} = \sum_{v=1}^{V} \alpha_v \mathbf{W}_{out}^{v\,T} \mathbf{\Phi}^v$.

By substituting (4) in (3) and taking the equivalent dual problem, we obtain:

$$\begin{aligned}\mathcal{J}_D(\boldsymbol{\alpha}) &= \frac{1}{2}\sum_{v=1}^{V} \|\mathbf{S}_w^{v\frac{1}{2}} \mathbf{W}_{out}^v\|_F^2 + \frac{c}{2}\boldsymbol{\alpha}^T \mathbf{P} \boldsymbol{\alpha} - c\mathbf{r}^T \boldsymbol{\alpha} \\ &+ \frac{c}{2} tr\left(\mathbf{T}^T \mathbf{T}\right) + \frac{\lambda}{2}\boldsymbol{\alpha}^T \boldsymbol{\alpha},\end{aligned} \tag{6}$$

where $\mathbf{P} \in \mathbb{R}^{V \times V}$ is a matrix having its elements equal to $[\mathbf{P}]_{kl} = tr\left(\mathbf{W}_{out}^{k\,T} \mathbf{\Phi}^k \mathbf{\Phi}^{l\,T} \mathbf{W}_{out}^l\right)$ and $\mathbf{r} \in \mathbb{R}^V$ is a vector having its elements equal to $\mathbf{r}_v = tr\left(\mathbf{T}^T \mathbf{W}_{out}^{v\,T} \mathbf{\Phi}^v\right)$. By solving for $\frac{\vartheta \mathcal{J}_D(\boldsymbol{\alpha})}{\vartheta \boldsymbol{\alpha}} = 0$, $\boldsymbol{\alpha}$ is given by $\boldsymbol{\alpha} = \left(\mathbf{P} + \frac{\lambda}{c}\mathbf{I}\right)^{-1} \mathbf{r}$.

By substituting (4) in (3) and taking the equivalent dual problem, we can also obtain:

$$\begin{aligned} \mathcal{J}_D(\mathbf{W}_{out}^v) &= \frac{1}{2}\sum_{v=1}^{V} tr\left(\mathbf{W}_{out}^{v\,T}\mathbf{S}_w^v\mathbf{W}_{out}^v\right) + \frac{c}{2} tr\left(\mathbf{T}^T\mathbf{T}\right) \\ &+ \frac{c}{2} tr\left(\sum_{v=1}^{V}\sum_{l=1}^{V}\alpha_v\alpha_l\mathbf{W}_{out}^{v\,T}\mathbf{\Phi}^v\mathbf{\Phi}^{l\,T}\mathbf{W}_{out}^l\right) \\ &- c\sum_{v=1}^{V} tr\left(\alpha_v\mathbf{W}_{out}^{v\,T}\mathbf{\Phi}^v\mathbf{T}^T\right) + \frac{\lambda}{2}\boldsymbol{\alpha}^T\boldsymbol{\alpha}. \quad (7) \end{aligned}$$

By solving for $\frac{\vartheta \mathcal{J}_D(\mathbf{W}_{out}^v)}{\vartheta \mathbf{W}_{out}^v} = 0$, $\mathbf{W}_{out}^v$ is given by:

$$\mathbf{W}_{out}^v = \left(\frac{2}{\alpha_v c}\mathbf{S}_w^v + \alpha_v\mathbf{\Phi}^v\mathbf{\Phi}^{v\,T}\right)^{-1}\mathbf{\Phi}^v(2\mathbf{T} - \mathbf{O})^T. \quad (8)$$

As can be observed in (**??**), (8), $\boldsymbol{\alpha}$ is a function of $\mathbf{W}_{out}^v$, $v = 1, \ldots, V$ and $\mathbf{W}_{out}^v$ is a function of α_v. Thus, a direct optimization of $\mathcal{J}_D$ with respect to both $\{\alpha_v, \mathbf{W}_{out}^v\}_{v=1}^V$ is intractable. Therefore, we employ an iterative optimization scheme formed by two optimization steps. In the following, we introduce a index t denoting the iteration of the proposed iterative optimization scheme.

Let us denote by $\mathbf{W}_{out,t}^v$, $\boldsymbol{\alpha}_t$ the network output and combination weights determined for the iteration t, respectively. We initialize $\mathbf{W}_{out,1}^v$ by using (2) and set $\boldsymbol{\alpha}_{1,v} = 1/V$ for all the action video representations $v = 1, \ldots, V$. By using $\mathbf{W}_{out,t}^v$, $\boldsymbol{\alpha}_{t+1}$ is updated by using (**??**). After the calculation of $\boldsymbol{\alpha}_{t+1}$, $\mathbf{W}_{out,t+1}^v$ are updated by using (8). The above described process is terminated when $(\mathcal{J}_D(t) - \mathcal{J}_D(t+1))/\mathcal{J}_D(t) < \epsilon$, where ϵ is a small positive value equal to $\epsilon = 10^{-10}$ in our experiments.

After the determination of the set $\{\alpha_v, \mathbf{W}_{out}^v\}_{v=1}^V$, the network response for a given set of test action vectors $\mathbf{x}_l^v$ is given by:

$$\mathbf{o}_l = \sum_{v=1}^{V}\alpha_v\mathbf{W}_{out}^{v\,T}\boldsymbol{\phi}_l^v. \quad (9)$$

4. EXPERIMENTS

In this section, we present experiments conducted in order to evaluate the performance of the proposed classification scheme in human action recognition. We have employed three publicly available databases, namely the Olympic Sports, the Hollywood2 and the Hollywood 3D databases. In the following subsections, we describe the databases and evaluation measures used in our experiments. Experimental results are provided in subsection 4.4.

We compare the performance of the proposed approach to that of two commonly used unsupervised video representation combination schemes, i.e., the concatenation of all the available video representations before training a SLFN network by using (2) and the mean output of V SLFN networks, each trained by using one video representation using (2).

Regarding the parameter values used in our experiments, they have been determined by following a grid search strategy using values $c = 10^q, q = 0, \ldots, 3$ and $\lambda = 10^l, l = 0, \ldots, 3$. The dimensionality of the BoF-based action video representations has been set equal to $D_v = D = 4000$, $v = 1, \ldots, V$. The number of hidden layer neurons has been set to $H_v = H = 1000$, $v = 1, \ldots, V$ in all the cases. For the hidden layer neurons, we have employed the χ^2 activation function:

$$\Phi_{\chi^2}(\mathbf{v}_j, b, \mathbf{x}_i) = exp\left(-\frac{1}{2b}\sum_{d=1}^{D}\frac{(\mathbf{x}_{id} - \mathbf{v}_{jd})^2}{\mathbf{x}_{id} + \mathbf{v}_{jd}}\right), \quad (10)$$

which has been found to outperform other choices, like the RBF and the sigmoid function. The parameter b has been set equal to the mean value of the χ^2 distances between the training action vectors and the network input weights. Since we employ a BoF-based action video representation, the network input weights have been chosen to have the form of histograms, i.e., to be nonnegative and with unit l_1 norm. For fair comparison, we employ the same network input weights in all the experiments.

4.1. The Olympic Sports database

The Olympic Sports database [18] consists of 783 videos depicting athletes practicing 16 sports: high-jump, long-jump, triple-jump, pole-vault, basketball lay-up, bowling, tennis-serve, platform, discus, hammer, javelin, shot-put, springboard, snatch, clean-jerk and vault. Example video frames of the database are illustrated in Figure 1a. We used the standard training-test split provided by the database (649 training and 134 test videos). The performance is evaluated by computing the average precision (AP) for each action class and reporting the mean AP over all classes (mAP), as suggested in [18].

4.2. The Hollywood2 database

The Hollywood2 database [19] consists of 1707 videos depicting 12 actions: answering the phone, driving car, eating, ghting, getting out of car, hand shaking, hugging, kissing, running, sitting down, sitting up and standing up. The videos have been collected from 69 different Hollywood movies. Example video frames of the database are illustrated in Figure 1b. We used the standard training-test split provided by the database (823 training and 884 test videos). Training and test videos come from different movies. The performance is evaluated by computing the mean Average Precision (mAP) over all classes, as suggested in [19].

Table 1. Action Recognition Performance (mAP) on the Hollywood2, Olympic Sports and Hollywood 3D databases.

	Olympic Sports	Hollywood2	Hollywood 3D (mAP)	Hollywood 3D (CR)
Method [4]	74.1 %	**58.2** %	-	-
Method [17]	-	-	15 %	21.8 %
Best Descriptor	66.48 %	52.04 %	20.8 %	23.05 %
Concatenation	67.4 %	55.97 %	20.6 %	21.75 %
Mean Output	73.15 %	56.26 %	26.16 %	26.96 %
Proposed Scheme	**82.12** %	**58.2** %	**30.79** %	**35.71** %

(a) (b) (c)

Fig. 1. *Video frames of the: a) Olympic Sports, b) Hollywood2 and c) Hollywood 3D databases.*

4.3. The Hollywood 3D database

The Hollywood 3D database [17] consists of 951 stereoscopic videos (left and right channel) depicting 12 actions: dance, drive, eat, hug, kick, kiss, punch, run, shoot, sit down, stand up, swim and use phone. Another class referred to as 'no action' is also included in the database. In our experiments we have used only one (the left) channel of each stereoscopic video. Example video frames of this database are illustrated in Figure 1c. We used the standard (balanced) training-test split provided by the database (643 videos are used for training and performance is measured in the remaining 308 videos). Training and test videos come from different movies. The performance is evaluated by computing the mean Average Precision (mAP) over all classes and the Classification Rate (CR), as suggested in [17].

4.4. Experimental Results

Table 1 illustrates the performance obtained by using different descriptor type combination approaches on the the Olympic Sports, Hollywood2 and the Hollywood 3D databases. We also report the best performance obtained by using one of the available descriptors. As can be seen, the use of the mean SLFN network output outperforms the use of an action video representation obtained by concatenating all the available action vectors. This seems reasonable, since in the case of concatenated video representations all the descriptors equally contribute to the discriminative ability of the combined representation. On the other hand, by training multiple networks, each one on a different descriptor type, the discriminative power of each video representation is not affected. The proposed optimization scheme, by properly combining the contribution of each representation on the final classification result, achieves the highest performance in all the cases, providing $2-9\%$ increase on the performance of the remaining combination schemes. In Table 1, we also provide the performance reported in [4] for the same action video representations and the best performance reported in [17]. In both cases, classification is performed by using SVM and a combined action video representation obtained by (element-wise) kernel matrix multiplication. As can be seen, the proposed approach outperforms them in most cases.

5. CONCLUSIONS

In this paper, we proposed an optimization scheme that can be employed for neural network-based action classification. Proper regularization terms have been incorporated in the ELM optimization problem in order to extend it to multi-view action classification. In order to determine both optimized network and action representation combination weights, we proposed an iterative optimization process. The proposed algorithm has been evaluated on three publicly available action recognition databases, where its performance has been compared with that of the best single-descriptor choice and two commonly used video representation combination approaches, i.e., the vector concatenation before learning and the network output combination by using networks trained on different descriptor types independently.

6. REFERENCES

[1] I. Pitas, "Digital video and television," *Createspace*, 2013.

[2] H. Baltzakis, A. Argyros, M. Lourakis, and P. Trahanias, "Tracking of human hands and faces through probabilistic fusion of multiple visual cues," *International Conference on Computer Vision Systems*, 208.

[3] O. Zoidi, A. Tefas, and I. Pitas, "Visual object tracking based on local steering kernels and color histograms," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 5, pp. 870–882, 2013.

[4] H. Wang, A. Klaser, C. Schmid, and C.L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 60, pp. 1–20, 2013.

[5] P. Spagnolo, T.D. Orazio, M. Leo, and A. Distante, "Moving object segmentation by background subtraction and temporal analysis," *Image and Vision Computing*, vol. 24, no. 5, pp. 411–243, 2006.

[6] A. Iosifidis, S.G. Mouroutsos, and A. Gasteratos, "A hybrid static/active video surveillance sytstem," *International Journal of Optomechatronics*, vol. 5, no. 1, pp. 80–95, 2011.

[7] A. Iosifidis, A. Tefas, and I. Pitas, "View-invariant action recognition based on artificial neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 3, pp. 412–424, 2012.

[8] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[9] G. Huang, Q. Zhu, and C. Siew, "Extreme learning machine: a new learning scheme for feedfowrard neural networks," *International Joint Conference on Neural Networks*, 2004.

[10] P.L. Bartlett, "The sample complexity of pattern classification with neural networks: the size of the weights is more importantthan the size of te network," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 525–536, 1998.

[11] R. Minhas, A. Baradarani, S. Seifzadeh, and Q.J. Wu, "Human action recognition using extreme learning machine based on visual vocabularies," *Neurocomputing*, vol. 73, no. 10, pp. 1906–1917, 2010.

[12] A. Iosifidis, A. Tefas, and I. Pitas, "Semi-supervised classification of human actions based on neural networks," *International Conference on Pattern Recognition*, 2014.

[13] A. Iosifidis, A. Tefas, and I. Pitas, "Multi-view human action recognition under occlusion based on fuzzy distances and neural networks," *European Signal Processing Conference*, 2012.

[14] A. Iosifidis, A. Tefas, and I. Pitas, "Minimum class variance extreme learning machine for human action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 11, pp. 1968–1979, 2013.

[15] A. Iosifidis, A. Tefas, and I. Pitas, "Dynamic action recognition based on dynemes and extreme learning machine," *Pattern Recognition Letters*, vol. 34, pp. 1890–1898, 2013.

[16] A. Iosifidis, A. Tefas, and I. Pitas, "Person identification from actions based on artificial neural networks," *IEEE Symposium Series on Computational Intelligence*, 2013.

[17] S. Hadfield and R. Bowden, "Hollywood 3d: Recognizing actions in 3d natural scenes," *Conference on Computer Vision and Pattern Recognition*, 2013.

[18] J.C. Nieble, C.W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable mition segemnts for activity classification," *European Conference on Computer Vision*, 2010.

[19] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," *Conference on Computer Vision and Pattern Recognition*, 2009.