Evaluating new variants of Motion Interchange Patterns

Yair Hanani Noga Levy Lior Wolf
Tel-Aviv University, Israel

Abstract

Action Recognition in videos is an active research field that is fueled by an acute need, spanning several application domains. Still, existing systems fall short of the applications' needs in real-world scenarios, where the quality of the video is less than optimal and the viewpoint is uncontrolled and often not static. In this paper, we extend the Motion Interchange Patterns (MIP) framework for action recognition. This effective framework encodes motion by capturing local changes in motion directions and additionally uses mechanisms to suppress static edges and compensate for global camera motion. Here, we suggest to apply the MIP encoding on gradient-based descriptors to enhance invariance to light changes and achieve a better description of the motion's structure. We compare our method using Patterns of Oriented Edge Magnitudes (POEM) and Difference of Gaussians (DoG) as gradient-based descriptors to the original MIP on two challenging large-scale datasets.

1. Introduction

Real world applications of human action recognition in video are yet to emerge. This, in spite of the growing success of commercial systems that are based on recent advances in other computer vision domains such as object recognition and face recognition. A current trend, attending directly to the needs of real world video analysis, is the shift from developing algorithms for benchmarks that are based on staged videos taken under controlled settings, to working with collections of unconstrained video. Compared to the first benchmarks, the more recent ones show a much larger variation in both scene parameters and imaging parameters, including the actors' identity and clothes, the scene background and illumination, camera viewpoint and motion, and image resolution and quality.

In order to work with unconstrained video, new video descriptors have emerged. The recently suggested Motion Interchange Patterns (MIP) method described in [11] encodes motion interchanges, i.e., the creation of a signature that captures at every time point and at every image location both the preceding motion flow and the next motion component. This is done using a patch-based approach, sometimes known as "self-similarity", and local pattern encoding. To decouple static image edges from motion edges, MIP incorporates a unique suppression mechanism, and to overcome camera motion, it employs a motion compensation mechanism. A bag-of-words approach is then used to pool this information from the entire video clip, followed, when appropriate, by a learned metric technique that mixes and reweighs the various features. In this work, we created new variants of the Motion Interchange Patterns (MIP) family by incorporating gradient-based descriptors.

2. Related Work

Action Recognition is an on-going research in Computer Vision, that is addressed by various approaches [22, 25]. One line of research extracts the high-level information of the human shape in motion by building explicit models of bodies [30], silhouettes [2] or 3D volumes [6]. In a recent paper [23] a bank of action templates is collected, and the templates are used for high-level action representation applied to challenging action recognition datasets.

A central family of approaches uses low-level representation schemes of the information in a video. These approaches can be further categorized as local descriptors, optical flow and dynamic-texture methods. The MIP framework, which is the basis of the current work, belongs to the dynamic-texture based representations.

Local descriptors. Recent methods use local descriptors for recognition [13, 28, 17]. The locality of the human motion in time and space is captured by a local spatio-temporal environment representation, using feature point descriptors borrowed from images or adapted to include temporal information. As a first stage, pixels that are potentially significant to understand the scenario are detected. These are often referred to as space-time interest points (STIP) [15]. The region around each interest point is represented by a local descriptor. To represent the entire video, these descriptors are processed and combined using, for example, a bag-of-words representation [16]. This approach was tested successfully on recent real-world datasets (e.g., [12]). However, a major drawback of this approach is the sensitivity to the number of interest points detected. In videos with subtle

978-0-7695-4990-3/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPRW.2013.46

motion, only a small number of interest points is detected, providing insufficient information for recognition. Videos with too much motion (textured motion such as waves in a swimming pool or leaves moving in the wind) may provide a lot of information irrelevant for recognition thus masking the relevant discriminative points required for recognition.

Optical-flow based methods. The optical flow between successive frames [1, 24], sub-volumes of the video [9], or surrounding the central motion [4, 5] is highly valuable for Action Recognition. A drawback of optical flow methods, is committing too soon to a particular motion estimate at each pixel. When these estimates are mistaken, they affect subsequent processing by providing incorrect information.

The Motion Boundary Histograms (MBH) descriptor proposed in [3] computes oriented histogram of differential optical flow, thus capturing motion while being more robust to camera motion than optical flow. An alternative method is based on dense trajectories [28, 29]. The trajectories are extracted efficiently with optical flow and represent the local motion information in the video. The descriptors are then computed as HOG, HOF or MBH on a spatio-temporal volume defined by the trajectory. This method is state of the art on the HMDB51 [14] dataset but was not tested before on the ASLAN [12] dataset.

Dynamic-texture representations. Existing techniques for recognizing textures in 2D images extend the textures to time-varying "dynamic textures" (e.g., [10, 7]). One such technique is Local Binary Patterns (LBP) [21], that extracts texture using local comparisons between a pixel and the pixels surrounding it, and encodes these relations as a short binary string. The frequencies of these binary strings are combined to represent the entire image region. In [10, 32], an extension of the LBP descriptor to 3D video data was successfully applied to facial expression recognition tasks.

The Local Trinary Patterns (LTP) descriptor of [31] is another LBP extension to videos. An LTP code of a pixel p is computed as follows: a spatial patch around p is defined as the central patch. In the next frame, a circle centered at the pixel corresponding to p is sketched, and spatial patches are uniformly distributed around it. A similar circle of patches is sketched in the preceding frame. Every pair of patches, one patch from the former frame and one from the next, is compared to the central patch in the current frame. A trinary bit represents the comparison result - whether the central patch is more similar to the patch in the preceding frame, the succeeding frame or if the two similarities are approximately the same. The comparisons conducted for all pairs of former and succeeding patches are represented in a trinary string. A video is partitioned into a regular grid of non-overlapping cells and the histograms of the LTP codes in each cell are then concatenated to represent the entire video. The Motion Interchange Patterns (MIP) (described in detail in Section 3) is closely related to LTP.

In this work, a single frame is locally encoded either based on the soft version of Patterns of Oriented Edge Magnitudes [27, 26], or by the Difference of Gaussians [19].

Patterns of Oriented Edge Magnitudes (POEM). An efficient image gradient-based descriptor, suggested in [27] and further investigated in [26]. POEM computes the gradient orientation for every pixel in the image and quantizes it. For each pixel p, the matching descriptor is the orientations histogram over a patch centered at p. The gradient magnitude of every pixel in the patch is assigned to the histogram bin corresponding to the nearest orientation. In the soft version of POEM, the magnitude assignment of every pixel in the patch is distributed between the bins corresponding to the two nearest orientations.

Difference of Gaussians (DoG). An image descriptor obtained by applying s Gaussian kernels differing in their σ values, thus constructing s blurred versions of the image [18]. For each pair of adjacent σ values, the difference between the corresponding blurred images is calculated. The DoG descriptors are $(s-1)$-dimensional vectors containing the computed differences per pixel.

3. Motion Interchange Patterns

Given an input video, the MIP encoding [11] assigns to every pixel of every frame eight strings of eight trinary digits each. Every single digit compares the compatibility of two motions with the local patch similarity pattern: one motion in a specific direction from the previous frame to the current frame, and one motion in another direction from the current frame to the next one. Figure 1 illustrates the motion structure extracted from comparing different patches. A value of -1 indicates that the former motion is more likely, 1 indicates that the latter is more likely. A value of 0 indicates that both are compatible in approximately the same degree.

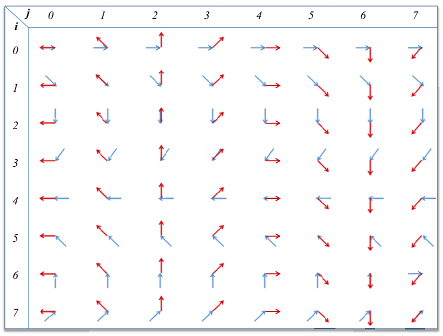

Figure 1. Representation of motion comparisons between two patches. For a given pixel and frame, blue arrows show the motion from a patch in the preceding frame and red arrows show the motion to a patch in the succeeding frame.

A 3×3 patch is centered around the given pixel. Eight possible locations in each of the previous and the next

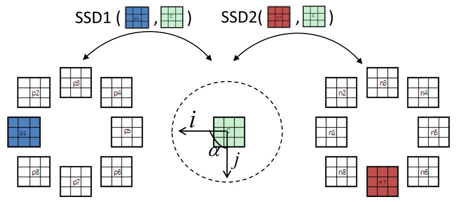

Figure 2. Each trinary digit in the MIP encoding represents a comparison of two SSD scores, both referring to the same central patch (in green). SSD1 is computed between the central patch and a patch in the previous frame (in blue), and SSD2 is computed between the central patch and a patch in the next frame (in red).

frames are denoted i and j (respectively) and numbered from 0 to 7. All 64 combinations of i and j are considered, and the resulting code is denoted by $S(p)$. Each digit $S_{i,j}(p)$ corresponds to one combination of patch locations in the previous and next frames (respectively).

The sum of squared differences (SSD) patch-comparison operator is used to set the matching bit. Denote by SSD1 (SSD2) the sum of squared differences between the patch in the previous (next) frame and the patch in the current frame, as depicted in Figure 2. Each trit, $S_{i,j}(p)$, is computed as follows, for some threshold parameter θ:

$$S_{i,j}(p) = \begin{cases} 1 & if \quad SSD1 - \theta > SSD2 \\ 0 & if \quad |SSD2 - SSD1| \leq \theta \\ -1 & if \quad SSD1 < SSD2 - \theta \end{cases} \tag{1}$$

MIP compares all eight motions to the eight subsequent motions, obtaining a comprehensive characterization of the change in motion at each video pixel.

To overcome the ambiguity introduced by camera motion even in motionless parts of the scene, a motion compensation module finds the alignment parameters that maximize the number of zero encoded pixels in the video. To avoid implausible motion patterns such patterns are suppressed.

Computing Similarity Positive and negative parts of each string are separated, obtaining 2 UINT8 per pixel, for each of the eight possible values of the angle between direction i and direction j. These 16 values represent the complete motion interchange pattern for that pixel. For each fixed value of α, the histograms of these MIP codes are pooled from a 16×16 patches around each image pixel, thus creating 512-dimensional code words. A bag-of-words is applied by employing k-means clustering on the code words obtained for the training images, $k = 5000$. Each local string is assigned to the closest word, and a video clip is then represented by eight histograms denoted as u^{α}.

Applying MIP in learning tasks For the vanilla supervised multi-class Action Recognition, the feature vector u representing a video clip is a concatenation of the eight u^{α} of all channels. Linear SVM is then used to build a suitable classification model.

In the action pair-matching task, the input comprises of pairs of video clips, labeled as describing the same action or a dissimilar action. This setting is cheaper to label, as it does not require specifying an actual action, and only refers to the similarity between two videos. Once a suitable similarity measure between a pair of actions is learned, this setting generalizes easily to measuring distance between previously unseen actions.

For this task, one can use the histograms directly (employing the L_2 similarity) or employ a metric learning step. The Cosine Similarity Metric Learning (CSML) algorithm [20] was previously shown to be effective for MIP encoding of the ASLAN benchmark. It is employed to each of the eight u^{α} vectors described above, and learns eight corresponding transformations T_{α}.

CSML is computationally demanding, therefore, before learning the CSML transformations, PCA is trained for each channel separately on a subset of the training data and the 50 most significant dimensions are used. The resulting transformation maps the feature vectors to a 30-dimensional space, and concatenating the channels, the final representation of a single video clips is a 240-dimensional feature vector. The feature vector representing the similarity between a pair of video clips is the element-wise multiplication of their transformed feature vectors. Finally, a binary SVM trains a similarity model on the feature vectors representing the training set pairs.

CSML metric is learned on a training set $\{(v_i, v_i'), l_i\}_{i=1}^{n}$ consisting of n pairs of samples labeled as same ($l_i = 1$) or not same ($l_i = -1$). The CSML optimization problem finds a transformation T which minimizes

$$CSML(T, \{(v_i, v_i')\}, \{l_i\}) = \sum_{\{i|l_i=1\}} CS(T, v_i, v_i') - \beta_1 \sum_{\{i|l_i=-1\}} CS(T, v_i, v_i') - \beta_2 ||T - I|| \, , \tag{2}$$

with I being the identity matrix, and the transformed cosine similarity defined as: $CS(T, v, v') = \frac{(Tv)^{\top}(Tv')}{||Tv|| \, ||Tv'||}$.

In the MIP paper, as well as here, the regularization parameter β_1 is set to one, and the parameter β_2 is optimized using a coarse to fine scheme as suggested in [20].

4. Overview of the new variants

We suggest two variants of the original MIP scheme. These variants are based on replacing the patch representation employed in MIP by representations that are based on the gradients within each video frame. We call the first variant histMIP, as it encodes each frame as a histogram of gradient orientations, and second variant DoGMIP after the Difference of Gaussians representation.

histMIP Given an input video, the texture of each frame is encoded separately as a collection of local histograms. These are then compared using the MIP scheme. For every pixel p in the frame, let $\theta(p)$ and $m(p)$ denote the gradient orientation and magnitude respectively. The orientation space $0-\pi$ is evenly discretized to d orientations $\phi_1,\ldots,\phi_d$ (indiscernible to opposing orientations). In our experiments we use $d=3$. Consider pixel p_i with orientation $\theta(p_i)$ and the nearest discrete orientations ϕ_{i_1} and ϕ_{i_2}, arranged $\phi_{i_1}\leq\theta(p_i)\leq\phi_{i_2}$. A d-dimensional vector $[\hat{m}_1(p_i),\ldots,\hat{m}_d(p_i)]$ is constructed by projecting the gradient magnitude $m(p_i)$ to the discrete orientations above. The projection is done by bilinear interpolation. Define $\alpha=\theta(p_i)-\phi_{i_1}$ and $\beta=\phi_{i_2}-\theta(p_i)$, then $\hat{m}_{i_1}(p_i)=\frac{\beta}{\alpha+\beta}m(p_i)$ and $\hat{m}_{i_2}(p_i)=\frac{\alpha}{\alpha+\beta}m(p_i)$ while all other coordinates are nullified.

To incorporate information from neighboring pixels, a local histogram of orientations over all the pixels within a local image patch is computed. At pixel p, the feature vector is $[\tilde{m}_1(p),\ldots,\tilde{m}_d(p)]$ where $\tilde{m}_i(p)=\sum_{p_j\in C}\hat{m}_i(p_j)$ and C refers to a patch (3×3) centered at the considered pixel.

The received histograms define d layers, where layer i refers to orientation ϕ_i and contains all $\tilde{m}_i(p)$ for all pixels in all the frames in the video. MIP is computed separately for each layer. In patchMIP, the distance between matching pixels in consecutive frames is computed as an SSD distance on $0-255$ gray-level intensities of local patches centered around the pixels. Instead, for each layer $i=1..d$ separately, we compute the distance between matching aggregated magnitude scalars from three consecutive frames. Let $\tilde{m}_i^{prev}(p)$, $\tilde{m}_i^{curr}(p)$ and $\tilde{m}_i^{next}(p)$ be the values matching pixel p for orientation ϕ_i. The distance $d_i(p)$ is $(\tilde{m}_i^{curr}(p)-\tilde{m}_i^{prev}(p))^2-(\tilde{m}_i^{next}(p)-\tilde{m}_i^{curr}(p))^2$.

Each comparison provides a trinary value as described in Eq. 1. We set the threshold to 2500, which is approximately the distance $d_i(p)$ between a pair of idential aggregated magnitude values, and a pair of patches with a constant magnitude gap of 5.5, translating into $\approx$ 50 difference between the aggregated matnitudes. Finally, the trinary values across the layers are concatenated.

DoGMIP In this variant, the texture information of each frame is extracted using Difference of Gaussians (DoG). DoG applies d Gaussian kernels with differing standard deviation σ on the image to achieve various levels of blurring. For every pair of subsequent blurred images sorted by their σ values, we subtract one blurred image from the other, resulting in $d-1$ subtracted images per frame. We use four Gaussian kernels with standard deviation 0.5, 1, 2 and 3, hence compute three layers of subtracted images per frame.

Each layer is encoded separately, where the scalar values of the DoG operators are used instead of the patches of the same locations, and simple scalar square distances replace the SSD operators. The trinary values are provided based on the compared distance values and a threshold, set in our experiments based on trial and error to 80, 40 and 7 for the three layers respectively. The final output from all three layers is concatenated to represent the video.

An example of both methods is shown in Figure 3. For each method, the gradient-based layers and the MIP motion descriptors are presented.

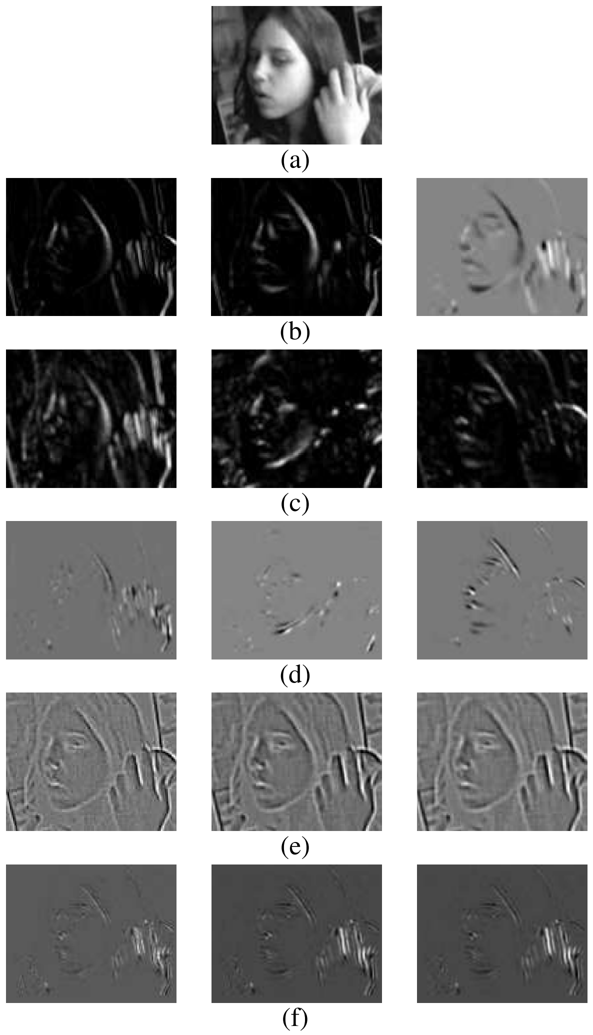

Figure 3. An example of the MIP varients encodings. (a) the original image, (b) patchMIP features. Left: SSD1, middle: SSD2, right :MIP encoding of the SSD differences, (c) histMIP features per layer (d) MIP encoding for each histMIP layer, (e) DoGMIP features per layer, (f) MIP encoding for each DoGMIP layer. The encoded motion extracted by the gradient-based variants is more accurate than the patchMIP encoding, possibly because patchMIP compares all image charectaristics over time, while histMIP and DoGMIP first extract image gradients and compare only this information over time, yielding a better localization of motion.

5. Experiments

We compare the performance of histMIP and DoGMIP to the performance of patchMIP with and without combining those to the STIP method [15], the dense trajectories method [29] and the Motion Boundary Histograms (MBH) method [3]. In all our experiments we use the MIP parameters as described in [11].

To evaluate the dense trajectories and the MBH we used the code published in [1] and employed it with the default parameters. For the HMDB51 dataset, our results are significantly lower than the results reported in [29].

We test on two challenging real-world Action Recognition benchmarks, ASLAN and HMDB51.

ASLAN. The Action Similarity Labeling (ASLAN) benchmark [12] is a large-scale benchmark containing thousands of video clips collected from YouTube and over 400 complex action classes. Following the Labeled Faces in the Wild face identification dataset [8], the authors supply a baseline test for the action pair-matching task ("same/not-same") using a cross-validation over 10 splits. The splits are mutually exclusive, and each contains 300 pairs of same action videos and 300 not-same pairs.

The results are averaged on the ten experiments. In each experiment, nine splits are used for training, and the tenth for testing. To ensure that the experiments are independent of each other, all intermediate models, such as the dictionary built in the BOW stage, the PCA matrices and the CSML transformations, are learned per experiment.

The comparison among all tested methods and combinations of methods with and without employing CSML transformation is presented in Table 1. For each method or combination of methods, we report the average accuracy $\pm$ standard error, and the aggregated Area Under the ROC Curve (AUC). Incorporating the STIP detection consistently pushes performance $1-3\%$ higher. Learning the CSML transformations further boosts performance by additional $1-2\%$. Combining the three MIP variants with MBH descriptors achieves the best recognition rates, with and without CSML transformations.

HMDB51. The Human Motion Database [14] contains 51 actions and at least 101 video clips per action, summing to a total of $6,766$ video clips. The data was collected from movies and public databases. The performance level of HOG and HOF is in the low twenties, which suggests that this dataset is very challenging. However, recently the dense trajectories method was reported to achieve a state-of-art performance of 46.6%.

The dataset was evaluated using the three splits benchmark, each containing a hundred clips per each action - 70 for training and 30 for testing. The splits were selected to display a representative mix of video quality and camera motion attributes. We did not use the stabilized HMDB51 and used the original video clips instead, as the MIP contains a motion compensation mechanism.

The results are depicted in Table 2. The patchMIP result is taken from [11]. We tested histMIP, DoGMIP and their combinations. Combining patchMIP, histMIP and DoGMIP achieves a significant improvement over each variant by its own, and when incorporating dense trajectories or MBH to this combination, the accuracy is further increased.

Table 2. Comparison of MIP variants, dense trajectories and MBH on the HMDB51 database, tested on the unstabilized HMDB51 data. Combininng the MIP variants with the other methods boosts the performance. The patchMIP results are taken from [11].

System	Accuracy
patchMIP	29.22%
histMIP	29.65%
dogMIP	22.5%
patchMIP + histMIP + dogMIP	34.77%
Traj (our own runs)	30.63%
MBH (our own runs)	29.13%
patchMIP + histMIP + dogMIP + Traj	**36.93%**
patchMIP + histMIP + dogMIP + MBH	36.71%

6. Conclusions

In countless competitive contributions in computer vision, including the original MIP work and the MBH work, the combination of multiple descriptors leads to a boost in performance. In action recognition, which still lags behind other computer vision domains with respect to performance, such a combination might be a necessity when considering the complexity of the tasks involved in real-world applications. We set to create new variants of the Motion Interchange Patterns framework. While not being able to present an increase in performance in comparison to the original MIP, combined with MIP, performance improves. We also present results, which currently lead the ASLAN benchmark, in which the MBH descriptor is incorporated into the set of descriptors employed. We are now working on directly combining the underlying encoding of MBG into the MIP framework, i.e., on creating a MIP variant which is based on optical flow and its derivatives. Hopefully, such a hybrid descriptor would capture the strengths of both methods.

References

[1] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *TPAMI*, 32(2), 2010. 2

[2] K. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *CVPR*, 2003. 1

[3] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *Computer Vision–ECCV 2006*, pages 428–441. Springer, 2006. 2, 5

[1] http://lear.inrialpes.fr/people/wang/dense_trajectories

Table 1. Performance comparison of various combinations of methods on the ASLAN benchmark. The average accuracy and standard error are given for a list of methods (see text for details). The patchMIP and patchMIP + STIP results are taken from [11].

System	No CSML		With CSML	
	Accuracy	AUC	Accuracy	AUC
patchMIP	$62.23 \pm 0.8\%$	67.5	$64.62 \pm 0.8\%$	70.4
patchMIP + STIP	$64.27 \pm 1\%$	69.2	$65.45 \pm 0.8\%$	71.92
histMIP	$61.13 \pm 0.9\%$	66.18	$64.3 \pm 1\%$	69.09
dogMIP	$60.13 \pm 1\%$	64.17	$60.82 \pm 1.1\%$	65.76
Traj (our own runs)	$59.9 \pm 0.7\%$	64.04	$62.02 \pm 1.1\%$	66.99
Traj + STIP	$62.1 \pm 0.9\%$	67.23	$64.5 \pm 1\%$	70.47
MBH (our own runs)	$62.43 \pm 0.8\%$	66.92	$64.25 \pm 0.9\%$	69.91
MBH + STIP	$63.9 \pm 0.9\%$	69.51	$65.55 \pm 0.9\%$	71.95
patchMIP + histMIP + dogMIP + STIP + Traj	$64.25 \pm 1\%$	69.91	$65.6 \pm 0.9\%$	72.06
patchMIP + histMIP + dogMIP + STIP + MBH	$\mathbf{65.63 \pm 1\%}$	**71.05**	$\mathbf{66.13 \pm 1\%}$	**73.23**

[4] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, 2003. 2

[5] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *CVPR*, 2008. 2

[6] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *TPAMI*, 29(12), 2007. 1

[7] T. Hassner, Y. Itcher, and O. Kliper-Gross. Violent flows: Real-time detection of violent crowd behavior. In *CVPRW*, 2012. 2

[8] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, UMASS, TR 07-49, 2007. 5

[9] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *ICCV*, 2005. 2

[10] V. Kellokumpu, G. Zhao, and M. Pietikäinen. Human activity recognition using a dynamic texture based method. In *BMVC*, 2008. 2

[11] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *Computer Vision–ECCV 2012*, pages 256–269. Springer, 2012. 1, 2, 5, 6

[12] O. Kliper-Gross, T. Hassner, and L. Wolf. The action similarity labeling challenge. *TPAMI*, 34(3), 2012. 1, 2, 5

[13] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR*, 2010. 1

[14] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011. 2, 5

[15] I. Laptev. On space-time interest points. *IJCV*, 2005. 1, 5

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1

[17] J. Liu, Y. Yang, I. Saleemi, and M. Shah. Learning semantic features for action recognition via diffusion maps. *CVIU*, 116(3), 2012. 1

[18] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999. 2

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[20] H. Nguyen and L. Bai. Cosine similarity metric learning for face verification. In *ACCV*, 2010. 3

[21] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, 24(7), 2002. 2

[22] R. Poppe. A survey on vision-based human action recognition. *IVC*, 28(6), 2010. 1

[23] S. Sadanand and J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012. 1

[24] K. Schindler and L. V. Gool. Action snippets: How many frames does human action recognition require? In *CVPR*, 2008. 2

[25] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *CSVT*, 18(11), 2008. 1

[26] N. Vu. Exploring patterns of gradient orientations and magnitudes for face recognition. 2013. 2

[27] N.-S. Vu and A. Caplier. Face recognition with patterns of oriented edge magnitudes. In *Computer Vision–ECCV 2010*, pages 313–326. Springer, 2010. 2

[28] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011. 1, 2

[29] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, pages 1–20, 2012. 2, 5

[30] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *CVPR*, 1992. 1

[31] L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *ICCV*, 2009. 2

[32] G. Zhao and M. Pietikäinen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *TPAMI*, 29(6), 2007. 2