Motion Binary Patterns for Action Recognition*

Florian Baumann[1], Jie Liao[2] , Arne Ehlers[1] and Bodo Rosenhahn[1]
[1]*Institut für Informationsverarbeitung, Leibniz Universität Hannover, Appelstraße 9a, 30167 Hannover*
[2]*Department of Electronic Science and Technology, USTC, Hefei, Anhui 230027,P.R. China*
{*baumann, ehlers, rosenhahn*}*@tnt.uni-hannover.de, ljtale@gmail.com*

Keywords: Human Action Recognition, Volume Local Binary Patterns, Random Forest, Machine Learning, IXMAS, KTH, Weizman

Abstract: In this paper, we propose a novel feature type to recognize human actions from video data. By combining the benefit of Volume Local Binary Patterns and Optical Flow, a simple and efficient descriptor is constructed. Motion Binary Patterns (MBP) are computed in spatio-temporal domain while static object appearances as well as motion information are gathered. Histograms are used to learn a Random Forest classifier which is applied to the task of human action recognition. The proposed framework is evaluated on the well-known, publicly available KTH dataset, Weizman dataset and on the IXMAS dataset for multi-view action recognition. The results demonstrate state-of-the-art accuracies in comparison to other methods.

1 Introduction

Human action recognition is a complex area of computer vision since static object characteristics, motion and time information have to be taken into account. Furthermore, actions are divided into human actions, human-human interactions, human-object interactions and group activities (Aggarwal and Ryoo, 2011). Due to environment variations such as moving backgrounds, different view points or occlusions the detection and classification of actions is even more difficult. Additionally, each actor has its own style of performing an action, leading to many variations in the subject's movement and a large intra-class variation (Aggarwal and Ryoo, 2011; Poppe, 2010).

Contribution In this work, we address the problem of recognizing actions performed by a single person, e.g. boxing, clapping, waving, walking, running, jogging. We suggest a simple and efficient novel feature type, namely Motion Binary Pattern (MBP), which combines static object appearances as well as motion information in the spatio-temporal space, in one descriptor. An MBP is computed from three frames followed by a histogram computation, leading to an invariance against different video lengths. Finally, the histogram is used to learn a Random Forest classifier. The proposed approach is evaluated on the single-view KTH dataset (Schuldt et al., 2004) and Weizman (Blank et al., 2005; Gorelick et al., 2007) dataset as well as on the IXMAS (Weinland et al., 2006; Weinland et al., 2010) dataset for multi-view action recognition. Figures 1, 2 and 3 show some example images of the used datasets.

Related Work The first Local Binary Pattern on Three Orthogonal Planes (LBP-TOP) was developed by Zhao et al. (Zhao and Pietikainen, 2007) to classify dynamic textures while Mattivi and Shao applied LBP-TOP (Mattivi and Shao, 2009; Shao and Mattivi, 2010) to the task of human action recognition. The authors reached 88.19% accuracy on the KTH dataset and by combining Extended Gradient LBP-TOP with Principal Component Analysis in (Mattivi and Shao, 2009) they reached 91.25%. Best results (92.69%) were achieved by combining Extended Gradient LBP-TOP with Dollar's detection method (Shao and Mattivi, 2010).

Liu et al. (Liu and Yuen, 2010) proposed a boosted EigenActions framework which calculates a spatio-temporal information saliency map (ISM) by estimating pixel density functions. It has only 81.5% accuracy for the KTH dataset but reaches accuracies up to 98.3% on the Weizman dataset.

Using only single descriptors, the best result is not more than 90% in the above mentioned papers. Better results are obtained by Yeffet et al. (Yeffet and Wolf, 2009), the average accuracy is a little more than 90%. And most recently, Kihl et al. (Kihl et al., 2013) reached 93.4% with a series of local polynomial ap-

*This work has been partially funded by the ERC within the starting grant Dynamic MinVIP.

proximation of Optical Flow (SoPAF).
Many approaches have drawbacks, for instance the amount of feature lead to ambiguities, cannot deal with different video lengths or have a large feature space. To overcome these issues, we suggest a new feature that combines the capabilities of describing static object appearances as well as motion information, in a single descriptor.

2 Method

In this Section, Volume Local Binary Patterns and Motion Binary Patterns are described. VLBPs have become famous for describing features in the spatio-temporal domain and are derived from simple Local Binary Patterns.
Our proposed MBP is computed in the X-Y-T space too and additionally takes the temporal step size into account.
Finally, Section 2.3 briefly describes the well-known machine learning approach Random Forest by Leo Breiman (Breiman, 2001).

2.1 Volume Local Binary Pattern

A Local Binary Pattern (LBP) was first described in (Ojala et al., 1994) for texture classification. The original LBP is computed in a 3×3 cell by comparing every gray value to the center one. If the neighbor values are larger than the center one, an *1* is assigned to the corresponding position, otherwise *0*. By computing a 3×3 - LBP the codeword length is 8 bit. This codeword is interpreted as a binary word and converted to a decimal number. Finally, a histogram of all occurring numbers is built.
Since LBP features describe static object appearances they are not suited for action recognition where motion and time information should be taken into account. A Volume Local Binary Pattern (VLBP) is introduced in (Zhao and Pietikainen, 2007). It is computed in the spatial and temporal domain and

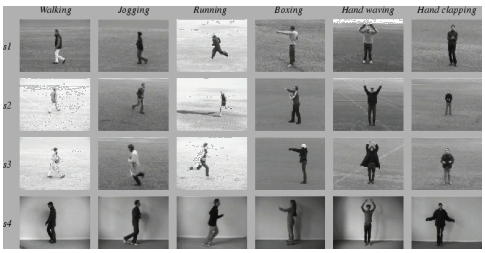

Figure 1: Example images of the single-view KTH dataset (Schuldt et al., 2004). The dataset contains six actions performed by 25 people under different conditions.

Figure 2: Example images of the multi-view IXMAS dataset (Weinland et al., 2006; Weinland et al., 2010). The dataset contains 12 actions. Each action is performed three times by 12 people.

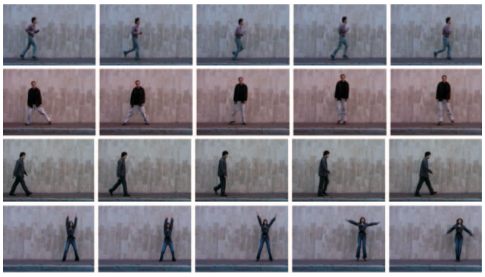

Figure 3: Example images of the single-view Weizman dataset (Blank et al., 2005; Gorelick et al., 2007). The dataset contains nine actions performed by nine people.

able to recognize dynamic textures. In (Zhao and Pietikainen, 2007), the authors define a radius around the center point within the space-time volume from three continuous frames to get neighboring pixels rather than using a 3×3 cell from one frame.
The computation of a VLBP is similar to the LBP: if the gray value of neighboring voxels within the space-time volume is larger than that of the voxel's center, the corresponding position is assigned to an *1*, otherwise *0*. By computing a VLBP the codeword length is 24 bit, leading to $2^{24} = 16777216$ different patterns. Similar to the LBP, a histogram of all occurring patterns is computed. Often, this huge feature pool leads to several ambiguities. To overcome this problem (Fehr, 2007; Topi et al., 2000) introduced an uniform LBP and demonstrate that the overall amount of LBPs can be reduced to a small subset. Experiments on object detection show that 90% of all possible patterns belong to this subset. For our application of action recognition uLBPs were unsuitable. Presumably, the final feature pool contains not enough information to find discriminative patterns in the spatio-temporal space.
Figure 4 illustrates how to compute a VLBP, the final result is 2039583. A VLBP is computed from three continuous frames followed by a histogram computation of all occurring patterns. The histogram is directly used to learn a Random Forest classifier.

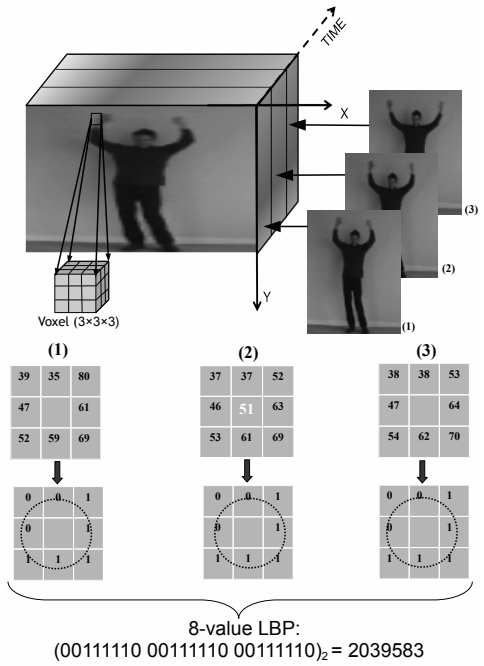

Figure 4: Procedure of computing a Volume Local Binary Pattern.

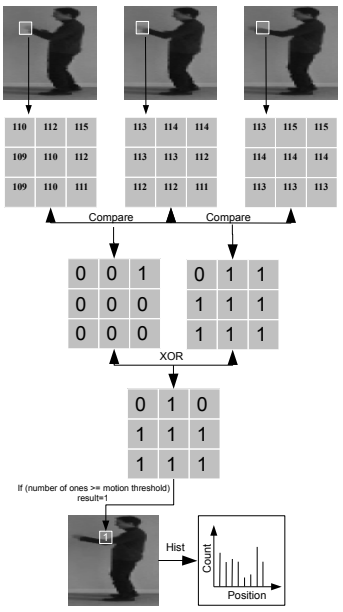

Figure 5: Procedure of computing our proposed MBP in three frames.

2.2 Motion Binary Pattern

In this Section, the characteristics and the computation of our proposed Motion Binary Pattern[2] is described. Assuming that motion can be detected by the change of pixel intensity values, MBPs are computed from three frames and measure the motion between them. Similar to the Optical Flow (Horn and Schunck, 1981), an MBP describes characteristics of motion. Figure 5 explains the computation of an MBP in three frames. The frames are divided into cells and for three cells at the same position, the corresponding values within one cell are compared. If the gray value within one cell of the first frame is larger than that in the second frame, an *1* is assigned, otherwise *0*. By using the same method, the third frame is compared to the second frame. The resulting two patterns are combined by using an exclusive OR (XOR), leading to the final motion pattern.

Regarding the computation of all MBPs in three frames, the number N of sampled patterns $C_n(x,y)$, $1 \leq n \leq N$ depends on the frame size, on the patterns's size and it's step size. Each pattern represents the motion between these frames while the binary values, especially the number of ones $\|C_n\|_1$, denoted by their entry-wise 1-norm, can be interpreted as the strength of motion. To distinguish between weak or strong motions, a motion vector $\vec{I} = (i_1, ..., i_N)^T$ is introduced. $\vec{I}$ is derived by a strength condition:

$$i_n = \begin{cases} 1, & \|C_n\|_1 \geq S, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Each element of the motion vector is set to *1* if the number of ones in the pattern $C_n(x,y)$ is higher than or equal to the motion strength threshold $S \in \{1 \dots 7\}$. Thus, $\vec{I}$ contains positions with a certain proportion of motion. It describes the spatial-dependent changing of intensity values from three frames and ideally only positions with large changes in the pixel intensity are assigned to *1*.

An MBP descriptor is created by computing a histogram from all motion vectors of a video. Thus, the histogram is directly used to construct a feature for learning a Random Forest classifier.

Temporal Variations In order to learn features from fast and slow motions, MBPs are not only computed from three continuous frames. Obviously only fast actions could be recognized by deriving features from continuous frames. In addition, four spatial scale steps are defined and MBPs are computed by incorporating these steps for shifting the pattern through the space-time volume. A time step of $t_s = 1,2,3,4$ was empirically chosen. For the case of $t_s = 1$, an MBP is computed from three continuous frames. Every second frame is collected for $t_s = 2$. Respectively,

[2]Source code available at http://www.tnt.uni-hannover.de/staff/baumann/

for $t_s = 3,4$ every third or fourth frame was chosen. Instead of creating a single histogram that can describe fast motions, four histograms are created to characterize different kind of motions. These histograms are concatenated and used to learn a Random Forest classifier.

2.3 Random Forest

In this part, a brief explanation of the theory behind Random Forest is given. Random Forests were developed by Leo Breiman (Breiman, 2001) and combine the idea of bagging (Breiman, 1996) with a random feature selection proposed by Ho (Ho, 1995; Ho, 1998) and Amit (Amit and Geman, 1997). A Random Forest consists of a collection of CART-like decision trees h_t, $1 \leq t \leq T$:

$$\{h(\vec{x}, \Theta_t)_{t=1,\ldots T}\}$$

where $\{\Theta_k\}$ is a bootstrap sample from the training data. Each tree casts a vote on a class for the input $\vec{x}$. The class probabilities are estimated by majority voting and used to calculate the sample's label $y(\vec{x})$ with respect to a given feature vector $\vec{x}$:

$$y(\vec{x}) = \underset{c}{\operatorname{argmax}} \left(\frac{1}{T} \sum_{t=1}^{T} F_{h_t(\vec{x})=c} \right) \tag{2}$$

The decision function $h_t(\vec{x})$ returns the result class c of one tree with the indicator function F:

$$F_{h_t(\vec{x})=c} = \begin{cases} 1, & h_t(\vec{x}) = c, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Random Forest has a high classification accuracy and can deal with large data sets for multiple classes with outstanding time efficiency.

2.3.1 Classification

Input descriptors are classified by passing them down each tree until a leaf node is reached. The result class is defined by each leaf node and the final decision is determined by taking the class having the most votes (majority vote), see Equation (2).

3 Experimental Results

KTH: VLBPs and MBPs are evaluated on the well-known and publicly available KTH dataset (Schuldt et al., 2004) consisting of six classes of actions. Each action is performed by 25 persons in four different scenarios. The KTH dataset consists of 599 videos. Similar to (O'Hara and Draper, 2012), a fixed position bounding box with a temporal window of 24 frames is selected, based on annotations by Lui (Lui et al., 2010). Presumably, a smaller number of frames is sufficient (Schindler and Van Gool, 2008). Furthermore, the original training/testing splits from (Schuldt et al., 2004) are used.

Weizman: In a second experiment we evaluate MBPs on the well-established Weizman action dataset (Blank et al., 2005; Gorelick et al., 2007). In our opinion, the Weizman dataset is already solved since many researchers report accuracies of 100%. However, in recent publications (Li et al., 2013; Tian et al., 2013; Yu et al., 2013) this dataset is still used to evaluate the corresponding methods. In order to allow a comparison to recent works and to show the benefit of our proposed method we evaluate Motion Binary Patterns on this dataset too. The Weizman dataset consists of nine actions while each action is performed by nine different persons. We manually labeled the dataset and used the bounding boxes for the classification. The bounding boxes are available for download at our homepage[3].

IXMAS: Additionally, we evaluated MBPs on the IXMAS dataset for multi-view action recognition (Weinland et al., 2006; Weinland et al., 2010). The IXMAS dataset contains 12 classes of actions. Each action is performed three times by 12 persons while the body position and orientation is freely chosen by the actor. The IXMAS dataset consists of 1800 videos. A 5-fold cross validation is used to get the results.

3.1 Evaluation for Volume Local Binary Patterns

Several strategies for computing VLBP values were tested. Two different neighborhoods (eight and four values) were compared, the influence of different histogram ranges as well as the difference between frame-by-frame learning and multi-frame learning has been evaluated. Best results were achieved by computing a 4-value VLBP with multi-frame learning (one histogram for all frames of a video is created) and a histogram range of 400 bins. Figure 6(a) shows the confusion matrix with an average accuracy of 89.81% for the KTH dataset.

3.2 Evaluation for Motion Binary Patterns

For computing an MBP, the motion strength threshold has to be adjusted. This parameter strongly influences

[3] http://www.tnt.uni-hannover.de/staff/baumann/

	box	walk	run	jog	wave	clap
box	0.97	0	0	0	0.03	0
walk	0	1	0	0	0	0
run	0	0	1	0	0	0
jog	0.03	0.03	0.03	0.91	0	0
wave	0.11	0	0	0	0.72	0.17
clap	0.08	0	0	0	0.14	0.78

(a)

	box	walk	run	jog	wave	clap
box	1	0	0	0	0	0
walk	0	0.97	0	0.03	0	0
run	0.03	0.03	0.80	0.03	0.03	0.08
jog	0	0.03	0	0.97	0	0
wave	0	0	0	0	0.83	0.17
clap	0	0	0	0	0.06	0.94

(b)

Figure 6: Confusion matrices for the KTH dataset, (a): reference method (VLBP) with 89.81% accuracy, (b) proposed method (MBP) with 91.83% accuracy. Most confusions occur in similar actions like *walking, running, jogging* and *boxing, waving, clapping*.

	chk	cro	scr	sit	get	tur	wal	wav	pun	kic	poi	pic
chk	1	0	0	0	0	0	0	0	0	0	0	0
cro	0	1	0	0	0	0	0	0	0	0	0	0
scr	0.17	0.17	0.49	0	0	0	0	0.17	0	0	0	0
sit	0	0	0	0.5	0.5	0	0	0	0	0	0	0
get	0	0	0	0.17	0.66	0.17	0	0	0	0	0	0
tur	0	0	0	0	0	1	0	0	0	0	0	0
wal	0	0	0	0	0	0	1	0	0	0	0	0
wav	0	0	0	0	0	0	0	0.66	0.34	0	0	0
pun	0	0	0	0	0	0	0	0	0.83	0	0.17	0
kic	0	0	0	0	0	0	0	0	0	1	0	0
poi	0	0.17	0	0	0	0	0	0	0.17	0	0.66	0
pic	0	0	0	0	0	0	0	0	0	0.17	0	0.83

Figure 7: Confusion matrix for applying Motion Binary Patterns to the IXMAS dataset. The average accuracy is 80.55%. Most confusions occur in similar actions like *waving, punching* and *sitting down, getting up*.

the performance of the MBP. Furthermore, MBPs are more sensitive to different image sizes. In this Section, we tested the influence of these parameters and compare the results to several recent approaches.

3.2.1 Influence of the Motion Strength Threshold

Section 2.2 gives a brief explanation about the motion strength threshold $S \in \{1 \dots 7\}$. Table 1 shows the recognition accuracy when the threshold varies for a frame size of 75×150. When the threshold is larger than seven, there will be fewer non-zero values in the MBP histogram. As listed in Table 1, recognizing accuracy increases when the threshold becomes larger. The highest accuracy 92.13% is achieved by using a threshold of six, leading to the assumption that this value is perfectly suited for the task of human action recognition.

	bend	jack	jump	pjump	run	side	walk	wave1	wave2
bend	1	0	0	0	0	0	0	0	0
jack	0	1	0	0	0	0	0	0	0
jump	0	0	1	0	0	0	0	0	0
pjump	0	0	0	1	0	0	0	0	0
run	0	0	0	0	1	0	0	0	0
side	0	0	0	0	0	1	0	0	0
walk	0	0	0	0	0	0	1	0	0
wave1	0	0	0	0	0	0	0	1	0
wave2	0	0	0	0	0	0	0	0	1

Figure 8: Confusion matrix for applying Motion Binary Patterns to the Weizman dataset. The accuracy is 100.00%.

Threshold	Average accuracy (%)
1	74.53
2	83.33
3	85.65
4	90.74
5	88.89
6	**92.13**
7	90.27

Table 1: Average accuracy for MBP on the KTH dataset with different motion thresholds. A threshold of 6 is leading to the best accuracy.

3.2.2 Influence of Different Frame Sizes

We report results on two approaches of changing frame sizes, using a motion strength threshold of six. In the first experiment, all frames are resized to a squared size. Table 2 shows that the average accuracy tends to increase too when increasing the side length of the square from 50 to 135 pixel, inspite of some fluctuations. However, when we raise the frame size from 135×135 to 150×150, the accuracy does not increase considerably.

In a second experiment all frames are resized to a rectangular size. Table 3 shows the accuracies for a proportionally changed width and height. By this means, we can achieve an accuracy of 92.13% when width and height of the rectangles are 75×150. Figure 6(b) presents the confusion matrix of this result. The accuracy for hand-waving and hand-clapping is higher than that by VLBP and the result for boxing can reach 100%. Unlike the VLBP, MBP does not do well in recognizing running. MBP is less sensitive to fast motions than VLBP and fast actions may puzzle the descriptor. But MBPs are sensitive to weak motions. From all our test results, wrong classifications mainly

Size	Average accuracy (%)
50×50	85.64
60×60	83.33
75×75	87.50
80×80	89.35
100×100	87.96
120×120	88.89
135×135	**90.74**

Table 2: Average accuracy for MBP on the KTH dataset by squared frame sizes. The best accuracy was achieved by taking a frame size of 135×135.

Size	Average accuracy(%)
64×128	89.35
128×64	87.96
75×150	**92.13**
150×75	87.50
80×160	89.35

Table 3: Average accuracy for MBP on the KTH dataset by different frame sizes. The best accuracy was achieved by taking a frame size of 75×150.

happen on similar actions like *walking, running, jogging* and *boxing, waving, clapping*, as showed in Figure 6(b).

3.2.3 Comparison to state-of-the-art methods

In this Section, we compare the proposed method to several state-of-the-art works and show that the MBPs are a very efficient descriptor for action recognition.
KTH: The MBP achieves an accuracy of 92.13% on the KTH dataset. Table 4 reports the accuracies of our proposed MBP in comparison to other methods. Motion Binary Patterns reach the highest accuracy for original training-/testing split and is only slightly lower than the best result with cross-validation.
IXMAS: Table 5 presents the results of MBP applied to the IXMAS dataset for multi-view action recognition in comparison to other state-of-the-art methods. Figure 7 shows the confusion matrix. The matrix also reveals that most confusions occur at similar actions like *waving, punching* and *sitting down, getting up*.
We used a motion threshold of six and window size of 75×150. For this experiment we compare MBPs to single- and multi feature approaches. Our result of 80.55%, see Table 5 is based on a 5-fold cross validation and is only slightly lower than the best result (Li and Zickler, 2012).
Weizman: Figure 8 shows the confusion matrix for applying MBPs to the Weizman dataset. The accuracy is 100.00%. Table 6 presents a comparison to single- and multi-feature methods. Several approaches report perfect recognition accuracies.

Name	Accuracy (%)
(Kihl et al., 2013)	93.4
(Kihl et al., 2013)	91.5
(Yeffet and Wolf, 2009)	90.1
(Laptev et al., 2008)	91.8
Proposed method	**92.1**
(Schindler and Van Gool, 2008)	92.7

Table 4: Comparison to recent approaches on the KTH dataset with a single descriptor.

Name	Accuracy (%)
(Wang et al., 2012)	76.50
(Wu et al., 2011)	78.02
Proposed method	**80.55**
(Li and Zickler, 2012)	81.22

Table 5: Average accuracy for MBPs on the IXMAS dataset in comparison to single- and multi-feature methods.

4 Conclusions and Future Work

In this paper a novel feature type, namely Motion Binary Patterns (MBP) are proposed. MBPs combine the advantages of Volume Local Binary Patterns to gather static object information and Optical Flow to obtain motion information. An MBP is computed from three frames with a temporal shifted sliding window. The resulting histograms are used to learn a Random Forest classifier.
The proposed feature is evaluated on the well-known, publicly available KTH dataset, Weizman dataset and on the IXMAS multi-view dataset. The results demonstrate state-of-the-art accuracies in comparison to Volume Local Binary Patterns and to other single- and multi feature methods. The source code for an MBP computation is available at http://www.tnt.uni-hannover.de/staff/baumann/.
Future Work Our plans for future work are to evaluate Motion Binary Patterns on more complex datasets like Hollywood (Laptev et al., 2008), Hollywood2 (Marszałek et al., 2009) or YouTube action dataset (Liu et al., 2009). Furthermore, we plan to eliminate the manual adjustment of the motion threshold by introducing an entropy function that chooses patterns with more discriminative power. Alternatively, we suggest to encode more information into the pattern. For instance, all temporal shifted patterns could be integrated into one final histogram. Additionally,

Name	Accuracy (%)
(Jhuang et al., 2007)	98.80
(Lin et al., 2009)	100.00
(Blank et al., 2005)	100.00
(Gorelick et al., 2007)	100.00
Proposed method	**100.00**
(Schindler and Van Gool, 2008)	100.00

Table 6: Average accuracy for MBPs on the Weizman dataset in comparison to single- and multi-feature methods.

more research is needed to choose the optimal cell size of a Motion Binary Pattern. In this paper we suggest to compute a MBP in a 3×3 cell but the results might be improved by taking other cell sizes like 5×5 or 7×7 .

REFERENCES

Aggarwal, J. and Ryoo, M. (2011). Human activity analysis: A review. *ACM Computing Surveys*, 43(3):16:1–16:43.

Amit, Y. and Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural computation*, 9(7):1545–1588.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. In *Computer Vision (ICCV), 10th International Conference on*, pages 1395–1402.

Breiman, L. (1996). Bagging predictors. In *Machine Learning*, volume 24, pages 123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Fehr, J. (2007). Rotational invariant uniform local binary patterns for full 3d volume texture analysis. In *Finnish signal processing symposium (FINSIG)*.

Gorelick, L., Blank, M., Shechtman, E., Irani, M., and Basri, R. (2007). Actions as space-time shapes. *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*, 29(12):2247–2253.

Ho, T. K. (1995). Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(8):832–844.

Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17.

Jhuang, H., Serre, T., Wolf, L., and Poggio, T. (2007). A biologically inspired system for action recognition. In *Computer Vision (ICCV), 11th International Conference on*, pages 1–8. IEEE.

Kihl, O., Picard, D., Gosselin, P.-H., et al. (2013). Local polynomial space-time descriptors for actions classification. In *International Conference on Machine Vision Applications*.

Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B. (2008). Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, (CVPR). IEEE Conference on*.

Li, R. and Zickler, T. (2012). Discriminative virtual views for cross-view action recognition. In *Computer Vision and Pattern Recognition, (CVPR). IEEE Conference on*.

Li, W., Yu, Q., Sawhney, H., and Vasconcelos, N. (2013). Recognizing activities via bag of words for attribute dynamics. In *Computer Vision and Pattern Recognition, (CVPR). IEEE Conference on*, pages 2587–2594.

Lin, Z., Jiang, Z., and Davis, L. S. (2009). Recognizing actions by shape-motion prototype trees. In *Computer Vision (ICCV), 12th International Conference on*, pages 444–451. IEEE.

Liu, C. and Yuen, P. C. (2010). Human action recognition using boosted eigenactions. *Image and vision computing*, 28(5):825–835.

Liu, J., Luo, J., and Shah, M. (2009). Recognizing realistic actions from videos "in the wild". In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1996–2003. IEEE.

Lui, Y. M., Beveridge, J., and Kirby, M. (2010). Action classification on product manifolds. In *Computer Vision and Pattern Recognition, (CVPR). IEEE Conference on*.

Marszałek, M., Laptev, I., and Schmid, C. (2009). Actions in context. In *Computer Vision and Pattern Recognition, (CVPR). IEEE Conference on*.

Mattivi, R. and Shao, L. (2009). Human action recognition using lbp-top as sparse spatio-temporal feature descriptor. In *Computer Analysis of Images and Patterns (CAIP)*.

O'Hara, S. and Draper, B. (2012). Scalable action recognition with a subspace forest. In *Computer Vision and Pattern Recognition, (CVPR). IEEE Conference on*.

Ojala, T., Pietikainen, M., and Harwood, D. (1994). Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Pattern Recognition. Proceedings of the 12th IAPR International Conference on*.

Poppe, R. (2010). A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976 – 990.

Schindler, K. and Van Gool, L. (2008). Action snippets: How many frames does human action recognition require? In *Computer Vision and Pattern Recognition, (CVPR). IEEE Conference on*.

Schuldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: a local svm approach. In *Pattern Recognition. (ICPR). Proceedings of the 17th International Conference on*.

Shao, L. and Mattivi, R. (2010). Feature detector and descriptor evaluation in human action recognition. In *Proceedings of the ACM International Conference on Image and Video Retrieval*.

Tian, Y., Sukthankar, R., and Shah, M. (2013). Spatiotemporal deformable part models for action detection. In *Computer Vision and Pattern Recognition, (CVPR). IEEE Conference on*.

Topi, M., Timo, O., Matti, P., and Maricor, S. (2000). Robust texture classification by subsets of local binary patterns. In *Pattern Recognition. (ICPR). Proceedings of the 15th International Conference on*.

Wang, Z., Wang, J., Xiao, J., Lin, K.-H., and Huang, T. (2012). Substructure and boundary modeling for continuous action recognition. In *Computer Vision and Pattern Recognition, (CVPR). IEEE Conference on*.

Weinland, D., Özuysal, M., and Fua, P. (2010). Making action recognition robust to occlusions and viewpoint changes,. In *European Conference on Computer Vision (ECCV)*.

Weinland, D., Ronfard, R., and Boyer, E. (2006). Free viewpoint action recognition using motion history volumes. In *Computer Vision and Image Understanding (CVIU)*.

Wu, X., Xu, D., Duan, L., and Luo, J. (2011). Action recognition using context and appearance distribution features. In *Computer Vision and Pattern Recognition, (CVPR). IEEE Conference on*.

Yeffet, L. and Wolf, L. (2009). Local trinary patterns for human action recognition. In *Computer Vision and Pattern Recognition, (CVPR). IEEE Conference on*.

Yu, T.-H., Kim, T.-K., and Cipolla, R. (2013). Unconstrained monocular 3d human pose estimation by action detection and cross-modality regression forest. In *Computer Vision and Pattern Recognition, (CVPR). IEEE Conference on*, pages 3642–3649.

Zhao, G. and Pietikainen, M. (2007). Dynamic texture recognition using local binary patterns with an application to facial expressions. *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*, 29(6):915–928.