# Computation Strategies for Volume Local Binary Patterns applied to Action Recognition[*]

F. Baumann, A. Ehlers, B. Rosenhahn
Institut für Informationsverarbeitung (TNT)
Leibniz Universität Hannover, Germany
lastname@tnt.uni-hannover.de

Jie Liao
Department of Electronic Science and Technology
Hefei, Anhui 230027,P.R. China
ljtale@gmail.com

## Abstract

*Volume Local Binary Patterns are a well-known feature type to describe object characteristics in the spatio-temporal domain. Apart from the computation of a binary pattern further steps are required to create a discriminative feature. In this paper we propose different computation methods for Volume Local Binary Patterns. These methods are evaluated in detail and the best strategy is shown. A Random Forest is used to find discriminative patterns. The proposed methods are applied to the well-known and publicly available KTH dataset and Weizman dataset for single-view action recognition and to the IXMAS dataset for multi-view action recognition. Furthermore, a comparison of the proposed framework to state-of-the-art methods is given.*

## 1. Introduction

A Volume Local Binary Pattern (VLBP) is famous for describing characteristics in the spatio-temporal domain and widely used in the computer vision community [14, 21, 29, 32]. VLBPs are easy to implement and efficient in their computation. But in order to create a discriminative feature for action recognition further steps are necessary.

In this work, different computation strategies for Volume Local Binary Patterns are introduced:

- Two different neighborhoods in comparison
- Frame-by-Frame vs. Multi-Frame classification
- Influence by different histogram ranges
- Step size of the pattern
- Temporal shifting

We compare a 4-value VLBP to an 8-value VLPB computation. A Frame-by-Frame classification strategy to a Multi-Frame classification and the influence by different histogram ranges is exploited. Finally, several step sizes are evaluated and in comparison to the computation of a VLBP in three continuous frames we propose to incorporate a temporal sliding window to describe fast and slow motions.

**Related Work:** Zhao et al. developed the first Local Binary Pattern on Three Orthogonal Planes (LBP-TOP) to classify dynamic textures [32]. LBP-TOP were applied to the task of human action recognition by Mattivi and Shao [14, 21]. The authors reached 88.19% accuracy on the KTH dataset. By combining LBP-TOP with Principal Component Analysis in [14] they reached 91.25%. Best results (92.69%) were achieved by combining Extended Gradient LBP-TOP with Dollar's detection method [21].

Yeffet et al. [29] developed Local Trinary Patterns and achieved an average accuracy 90.16%. And most recently, Kihl et al. [9] reached 93.4% with a series of local polynomial approximation of Optical Flow (SoPAF).

In comparison to these methods we demonstrate several computation strategies for binary patterns and show the most promising one. The results from this paper can be applied to all types of volumetric binary patterns.

## 2. Volume Local Binary Patterns

The Volume Local Binary Pattern (VLBP) was introduced in [32] and is able to recognize dynamic textures. In [32], 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 $3 \times 3$ cell from one frame. Figure 1 illustrates how to compute a VLBP, the final result is 3948094.

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 *1*, otherwise *0*. By computing a VLBP the codeword length is 24 bit, leading to

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

$2^{24} = 16777216$ different patterns. To overcome the problem of this huge feature pool Fehr et al. introduced an uniform LBP and demonstrate that the overall amount of LBPs can be reduced to a small subset [5, 23]. Experiments on images show that 90% of all possible patterns belong to this subset. In our opinion, for the task of action recognition the number of uLBPs is insufficient.

**Temporal Variations** In order to learn features from fast and slow motions, VLBPs 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 VLBPs 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 $t_s = 1$, a VLBP is computed from three continuous frames. Every second frame is collected for $t_s = 2$. Respectively, 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 directly used to learn a Random Forest classifier.

## 3. Computation Strategies

This Section explains several computation methods after extracting a VLBP in the spatio-temporal domain. We compare the difference between an 8-value and a 4-value computation, the difference between Frame-by-Frame learning and Multi-Frame learning, the influence of several histogram ranges as well as an overlapping vs. non-overlapping computation of a VLBP. For each comparison pair, all other conditions except the compared one were fixed. Finally, a general conclusion of all these methods is given.

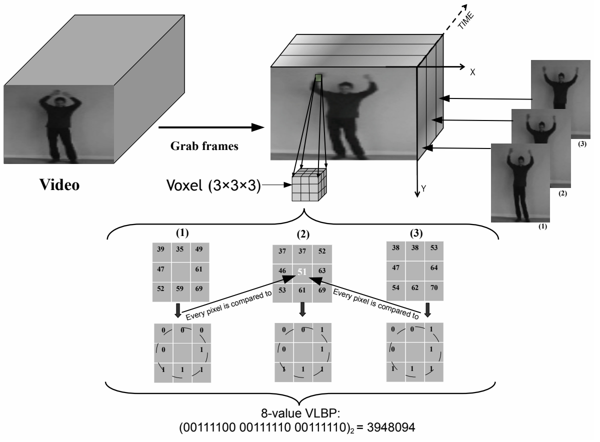


Figure 1. Procedure of computing an 8-value VLBP in three continuous frames.

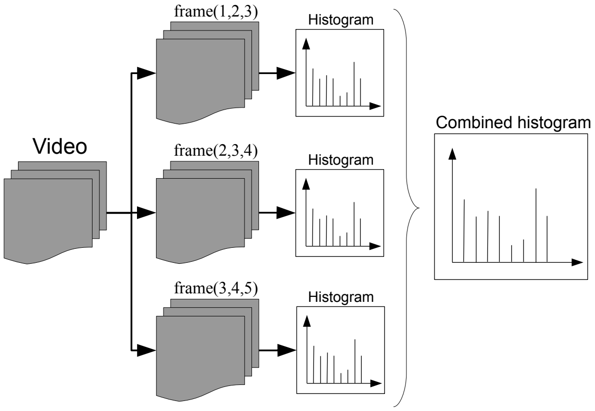


Figure 2. For the Frame-by-Frame learning approach a histogram is computed for three frames. Feature vectors are constructed by using each single histogram. All histograms of a video are combined into a single histogram for the Multi-Frame learning approach.

### 3.1. 8-value VLBP vs 4-value VLBP

The computation of a VLBP is explained in Figure 1. By computing a 4-value VLBP only four neighbors are taken into consideration. The 8-value VLBP computation uses all neighboring pixels to create the binary word.

### 3.2. Frame-by-Frame vs. Multi-Frame

Frame-by-Frame learning also called One-Shot Learning [15, 20] is a method that achieves a classification result by deriving features from one frame. A Multi-Frame classification takes several frames or the whole video into consideration for deriving a feature. Generally, a Multi-Frame classification is more discriminative leading to a global presentation.

**Multi-Frame:** To create a global feature, a VLBP is computed in three continuous frames. This process is done for the whole video until the last three frames are reached. Finally, a combined histogram that represents all motions of the video is created. This histogram is interpreted as a feature vector and directly used to learn a Random Forest classifier.

**Frame-by-Frame:** Similar to the Multi-Frame method a VLBP histogram is computed from three continuous frames until the last three frames are reached. In comparison to the Multi-Frame approach, each histogram is interpreted as a feature vector and directly used for learning a Random Forest classifier. Thus, the Random Forest is built from motions of single frames. But additionally for classification, the Frame-by-Frame learning approach requires a step of fusing the decisions of each frame to the final decision. Inspired by the majority voting of the Random Forest we are taking the class which gains the most votes.

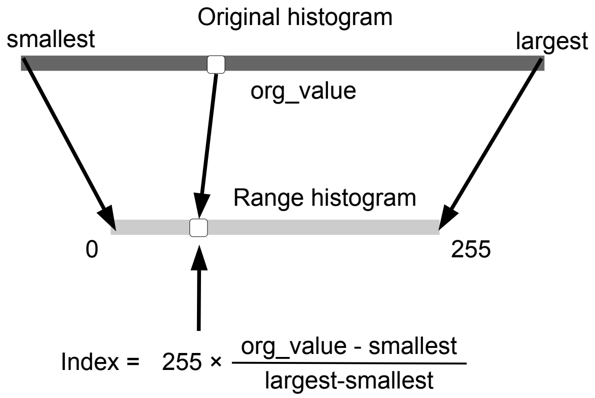


Figure 3. Example for mapping the original histogram to a range. In this example, the values are mapped to a range of 0 and 255.

Figure 2 presents both methods for a video with five frames. For the Multi-Frame method all histograms are combined to one histogram and for the Frame-by-Frame method several histograms are used to learn a Random Forest classifier.

### 3.3. Influence by different histogram ranges

As discussed in Section 2, the 8-value VLBP values can be as large as $2^{24} = 16777216$ and for the 4-value VLBP $2^{12} = 4096$ values, leading to many ambiguities and much useless information. To deal with this issue, we propose to map the histogram to a range. We map the histogram to several ranges between $[0, 127]$ and $[0, 8191]$. Figure 3 gives an explanation of how to map the original histogram to a range of $[0, 255]$.

### 3.4. Overlap vs. Non-overlap

The step size indicates the overlap of a VLBP in the spatio-temporal domain and specifies how accurate an action should be described. A step size of $x = 1, y = 1$ results in a pixel-wise shifting of the VLBP. A step size of $x = 9, y = 9$ is leading to the computation of non-overlapping VLBPs.

## 4. Random Forest

Random Forests were introduced by Leo Breiman [4] and bases on the idea of bagging [3] with a random feature selection proposed by Ho [7] and Amit [1]. A Random Forest consists of several CART-like decision trees $h_t$, $1 \leq t \leq T$:

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

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 $\vec{x}$:

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

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

$$F_{h_t(\vec{x})=c} = \begin{cases} 1, & h_t(\vec{x}) = c, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

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

**Classification** Images are classified by passing them down each tree until a leaf node is reached. The result class is assigned to each leaf node and the final decision is determined by taking the class having the most votes (majority vote), see Equation (1).

## 5. Experimental Results

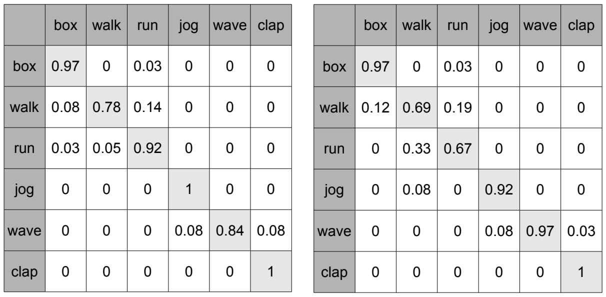
The results obtained in this paper base on the well-known KTH dataset [19], Weizman dataset [2, 6] and IXMAS for multi-view recognition [27, 26].

The **KTH** dataset contains six classes of actions: *boxing, walking, jogging, running, hand-waving, hand-clapping*, respectively each action is acted by *25* persons in *4* different scenarios: *outdoors, outdoors with scale variations, outdoors with different clothes and indoors*. There are totally 599 videos. Similar to [16], a fixed position bounding box with a temporal window of 32 frames is selected, based on annotations by Lui [13]. Presumably, a smaller number of frames is sufficient [18]. Furthermore, the original training/testing splits from [19] as well as a 5-fold cross validation strategy are used.

In a second experiment we evaluate our approach on the **Weizman** dataset [2, 6]. In our opinion, the Weizman dataset is already solved since many researchers report accuracies of 100%. However, in recent publications [11, 22, 30] 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 VLBPs 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[1].

Additionally, we evaluate the VLBPs on the **IXMAS** dataset for multi-view action recognition [27, 26]. The IXMAS dataset consists of 12 classes of actions. Each action is performed three times by 12 persons while the body position and orientation was freely chosen by the actor. The IXMAS dataset consists of 1800 videos.

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



| | box | walk | run | jog | wave | clap |
|---|---|---|---|---|---|---|
| box | 0.97 | 0 | 0.03 | 0 | 0 | 0 |
| walk | 0.08 | 0.78 | 0.14 | 0 | 0 | 0 |
| run | 0.03 | 0.05 | 0.92 | 0 | 0 | 0 |
| jog | 0 | 0 | 0 | 1 | 0 | 0 |
| wave | 0 | 0 | 0 | 0.08 | 0.84 | 0.08 |
| clap | 0 | 0 | 0 | 0 | 0 | 1 |

(a)

| | box | walk | run | jog | wave | clap |
|---|---|---|---|---|---|---|
| box | 0.97 | 0 | 0.03 | 0 | 0 | 0 |
| walk | 0.12 | 0.69 | 0.19 | 0 | 0 | 0 |
| run | 0 | 0.33 | 0.67 | 0 | 0 | 0 |
| jog | 0 | 0.08 | 0 | 0.92 | 0 | 0 |
| wave | 0 | 0 | 0 | 0.08 | 0.97 | 0.03 |
| clap | 0 | 0 | 0 | 0 | 0 | 1 |

(b)

Figure 4. (a): Confusion matrix of 4-value VLBP, with 91.83% average accuracy. (b) Confusion matrix of 8-value VLBP, with 87.00% average accuracy.

### 5.1. Influence of neighborhoods

In this experiment, we show results of taking a neighborhood of four and eight values into consideration. The histogram range is fixed to 512 bins while one histogram for all frames of a video is computed with a step size of $x = 1, y = 1$.

Figure 4(a) shows a confusion matrix for a 4-value VLBP computation with an average accuracy of 91.83% while Figure 4(b) shows an 8-value VLBP computation with an average accuracy of 87,00%. The result of 4-value VLBP is better than the 8-value VLBP. Presumably, the amount of $2^{24} = 16777216$ patterns for the 8-value computation results in ambiguities (every noisy pixel results in another pattern).

### 5.2. Frame-by-Frame vs. Multi-Frame

For the evaluation of Frame-by-Frame vs. Multi-Frame the range was fixed to 512 while a neighborhood of four and eight values was used with a step size of $x = 1, y = 1$. Table 1 presents the results of comparing the Frame-by-Frame learning approach to the Multi-Frame learning. Multi-Frame learning obtains better results. It can be observed that the 4-value VLBP outperforms the 8-value VLBP.

### 5.3. Influence by different histogram ranges

Different ranges cause different influence on the final result. In this experiment we show results how the accuracy is influenced by different histogram ranges $[128, 256, 512, 1024, 2048, 4096, 8192]$. A neighborhood of four and eight values for the VLBP computation is taken into consideration. One histogram for all frames of a video is computed with a step size of $x = 1, y = 1$. A 5-fold cross-validation was used to get the results.

| Type | 4-value VLBP | 8-value VLBP |
|---|---|---|
| Frame-by-Frame | 86.29% | 82.96% |
| Multi-Frame | **90.55%** | 88.24% |

Table 1. Comparison of the Frame-by-Frame learning approach to the Multi-Frame learning. The Multi-Frame learning outperforms the Frame-by-Frame learning.

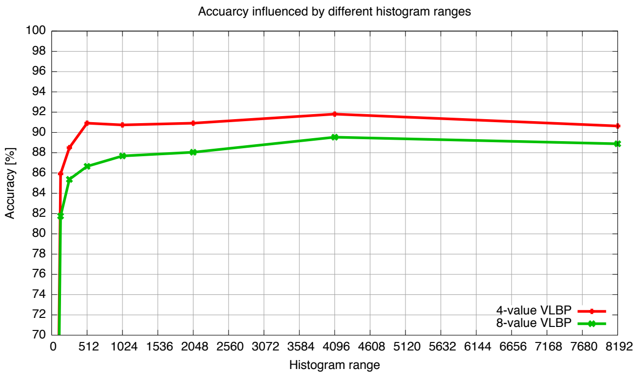


Figure 5. Influence of different ranges to the accuracy of 4-value and 8-value VLBP. Increasing the range is leading to better accuracies. It is also evident that the 4-value VLBP clearly outperforms the 8-value VLBP.

As showed in Figure 5, the accuracies of 4-value and 8-value VLBP increase with rising the histogram range. It also shows that the 4-value VLBP clearly outperforms the 8-value VLBP for all ranges.

### 5.4. Step size of shifting VLBPs

In this section we evaluate the influence of the step size to the accuracy. For this experiment one histogram for all frames of a video is computed. Also, a 5-fold cross validation was used to get the results. Table 2 presents the results of changing the step size from 1 to 9. The results demonstrate that a step size of $x = 1, y = 1$ clearly outperforms. The amount of information is higher leading to better accuracies for the 4-value and 8-value VLBP.

### 5.5. Conclusion of our proposed methods

We evaluated computation strategies for the 8-value VLBP and for the 4-value VLBP. The results show that

| Step size | 4-value VLBP | 8-value VLBP |
|---|---|---|
| $x = 1, y = 1$ | **90.92%** | 87.87% |
| $x = 2, y = 2$ | 90.46% | 86.46% |
| $x = 4, y = 4$ | 90.18% | 86.64% |
| $x = 8, y = 8$ | 79.62% | 81.75% |
| $x = 9, y = 9$ | 78.98% | 79.07% |

Table 2. Different step sizes $x, y$ for sliding the VLBP through the $x$- and $y$- dimension of the space-time volume. A step size of $x = 1, y = 1$ clearly outperforms.

| Name | Accuracy |
|---|---|
| Jhuang et al. [8] | 98.80% |
| Lin et al. [12] | 100.00% |
| Blank et al. [2] | 100.00% |
| Gorelick et al. [6] | 100.00% |
| **Proposed method** | **100.00%** |
| Schindler and Gool [18] | 100.00% |

Table 3. Average accuracy for VLBPs on the Weizman dataset in comparison to single- and multi-feature methods.

the 4-value VLPB clearly outperforms the 8-value VLBP in all experiments. The comparison between Frame-by-Frame and Multi-Frame classifications reveals that the Multi-Frame approach creates a more discriminative classifier, leading to better accuracies. The influence of different histogram ranges shows that a range between $0, 512$ and $0, 4096$ is leading to highest accuracies. Finally, the computation of VLBPs with a step size of $x = 1$ and $y = 1$ indicates the most promising step size.

### 5.6. Comparison to state-of-the-art methods

In this section we compare a 4-value VLBP with a range of $512$ bins, Multi-Frame classification and a step size of $x = 1$ and $y = 1$ to state-of-the-art approaches.

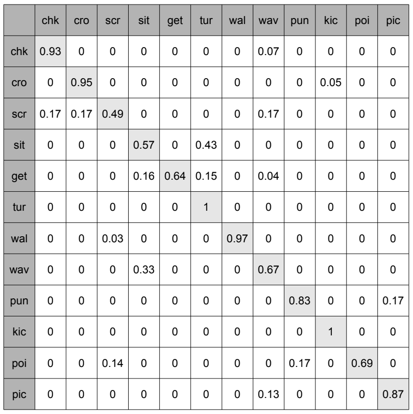
For the **KTH** dataset we implemented a 5-fold cross validation and used original training/testing split for the comparison. Table 4 presents the results in comparison to recent approaches. For both validation strategies the VLBPs achieve competing result.

Table 3 presents a comparison to single- and multi-feature methods on the **Weizman** dataset. Several approaches report perfect recognition accuracies.

For the **IXMAS** dataset we used a 5-fold cross validation. Figure 6 shows the confusion matrix and Table 5 presents results in comparison with other approaches. The results show competing accuracies in comparison to more complex state-of-the-art methods.

| Method | Validation | Accuracy |
|---|---|---|
| Schindler and Gool [18] | 5-fold | 87,98% |
| **Proposed method** | **5-fold** | **91,94%** |
| Zhang et al. [31] | 5-fold | 94,60% |
| Laptev et al. [19] | Original split | 91,80% |
| Zhang et al. [31] | Original split | 94,00% |
| Wang et al. [24] | Original split | 94,20% |
| **Proposed method** | **Original split** | **95,16%** |
| O'Hara and Draper [16] | Original split | 97,90% |
| Sadanand and Corso [17] | Original split | 98,20% |

Table 4. In comparison to state-of-the-art methods our proposed method achieves competing accuracies. Results were conducted on the KTH dataset [19].



| | chk | cro | scr | sit | get | tur | wal | wav | pun | kic | poi | pic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| chk | 0.93 | 0 | 0 | 0 | 0 | 0 | 0 | 0.07 | 0 | 0 | 0 | 0 |
| cro | 0 | 0.95 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.05 | 0 | 0 |
| scr | 0.17 | 0.17 | 0.49 | 0 | 0 | 0 | 0 | 0.17 | 0 | 0 | 0 | 0 |
| sit | 0 | 0 | 0 | 0.57 | 0 | 0.43 | 0 | 0 | 0 | 0 | 0 | 0 |
| get | 0 | 0 | 0 | 0.16 | 0.64 | 0.15 | 0 | 0.04 | 0 | 0 | 0 | 0 |
| tur | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| wal | 0 | 0 | 0.03 | 0 | 0 | 0 | 0.97 | 0 | 0 | 0 | 0 | 0 |
| wav | 0 | 0 | 0 | 0.33 | 0 | 0 | 0 | 0.67 | 0 | 0 | 0 | 0 |
| pun | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.83 | 0 | 0 | 0.17 |
| kic | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| poi | 0 | 0 | 0.14 | 0 | 0 | 0 | 0 | 0 | 0.17 | 0 | 0.69 | 0 |
| pic | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.13 | 0 | 0 | 0 | 0.87 |

Figure 6. Confusion matrix for the VLBP on the IXMAS dataset [27, 26]. A range of 512 bins was used, as well as the Multi-Frame approach with a step size of $x = 1$, $y = 1$. The average accuracy is 80.08%.

| Name | Average accuracy |
|---|---|
| STM+DBM [25] | 76.50% |
| AFMKL [28] | 78.02% |
| **Proposed method** | **80.08%** |
| Cross-View [10] | 81.22% |

Table 5. Average accuracy for VLBPs on the IXMAS dataset [27, 26] in comparison to other single- and multi-feature approaches our proposed method achieves competing results.

## 6. Conclusions

In this paper several post-processing strategies for Volume Local Binary Patterns (VLBP) are proposed. The experiments reveal that a computation of a 4-value outperforms the 8-value VLBP. Furthermore, it is more convenient to compute a histogram for a whole video than using a Frame-by-Frame learning approach. The optimal histogram range is 512 bins leading to highest accuracies. Our proposed method was applied to the publicly available and well-known KTH dataset, Weizman dataset and IXMAS dataset for action recognition. We used a 5-fold cross validation as well as the original training/testing split to compare our proposed methods to state-of-the-art results. Furthermore, the results of this paper can be applied to all spatio-temporal features to enhance the accuracies.

## References

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

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

[3] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

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

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

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

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

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

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

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

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

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

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

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

[15] T. Mauthner, P. M. Roth, and H. Bischof. Instant action recognition. In *Proceedings of the 16th Scandinavian Conference on Image Analysis, (SCIA)*, 2009.

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

[17] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *Computer Vision and Pattern Recognition, (CVPR). IEEE Conference on*, 2012.

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

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

[20] H. J. Seo and P. Milanfar. Action recognition from one example. *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*, May.

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

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

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

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

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

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

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

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

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

[30] T.-H. Yu, T.-K. Kim, and R. Cipolla. 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, 2013.

[31] Y. Zhang, X. Liu, M.-C. Chang, W. Ge, and T. Chen. Spatio-temporal phrases for activity recognition. In *In Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[32] G. Zhao and M. Pietikainen. 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, 2007.