RECOGNIZING HUMAN ACTIONS BY BP-ADABOOST ALGORITHM UNDER A HIERARCHICAL RECOGNITION FRAMEWORK

Nijun Li, Xu Cheng, Suofei Zhang, Zhenyang Wu

Key Laboratory of Underwater Acoustic Signal Processing of Ministry of Education
Southeast University, Nanjing, 210096, P. R. China
{lnjleo, xcheng, zhangsuofei, zhenyang}@seu.edu.cn

ABSTRACT

This paper explores the performance of Neural Network (NN) for human action recognition and proposes a novel hierarchical and boosting-based action recognition system. Specifically, the main contributions of our work are three-fold: (1) A boosted NN based scheme is applied to the human action recognition task for the first time, during which we extend the standard binary AdaBoost algorithm to a multiclass version; (2) A novel hierarchical recognition framework with pre-decision and post-decision modules is proposed, which can significantly enhance the training efficiency as well as the frame-based recognition accuracy; (3) Numerous modified features (both motion and shape features) are utilized and combined in this paper. Experiments on the Weizmann dataset show promising results of our approach in comparison with other state-of-the-art methods.

Index Terms — action recognition, feature extraction, BP-AdaBoost, neural network, pre/post-decision

1. INTRODUCTION

Human action recognition, which is one of the most challenging tasks to the computer vision community and the machine learning community, has attracted much attention in the past decades [1,2] due to its profound applications. Although there is no generic solution, researchers have proposed many algorithms for "feature extraction" and "machine learning" with various mechanisms and structures, achieving considerable progress.

(a) Feature extraction. Optical flow, which is a representative of motion features, has always been widely used for its high discriminative ability [3-5]. Besides, three types of shape features are also very popular: Histograms of Gradients (HoG) [6], silhouettes [7] and Spatial-Temporal Volumes (STV) [8,9], where the former two neglect the time dimension and just extract frame-based features. Histograms of Optical Flow (HoF) and those shape features can all be considered as global features**,** which usually have more discriminative power than local features**,** but at the same time are more affected by imaging conditions. Differently, there is no need to perform human detection, tracking or contour extraction to extract local features such as Space-Time Interest Points (STIP) [10], which can be further fitted into the Bag-of-Features (BoF) mechanism [11,12]. Local features are also more robust to imaging condition changes and variations among individuals, but sometimes they are too sparse.

(b) Machine learning. Template matching, discriminative methods and generative methods can represent the most widely used machine learning methods for action recognition. "Templates" can be acquired through various ways [4,11,13] and the k-nearest-neighbor method is often adopted to cast the classification directly. SVM (Support Vector Machine) and NN (Neural Network) are typical examples of discriminative methods, which achieve a reasonable division of points in the feature space. Numerous works [6,12,14,15] choose SVM as the classifier, but surprisingly there are few papers using NN. Discriminative methods, which are driven by data within a bottom-up mechanism, are easy to implement because they are nonparametric. On the other hand, generative methods use semi-structured graphs such as HMM (Hidden Markov Model) [16,17] to model the transition of different states of an action. Since generative methods take the contextual constraints into account, usually they can achieve better results but meanwhile are more computational expensive.

2. SYSTEM OVERVIEW

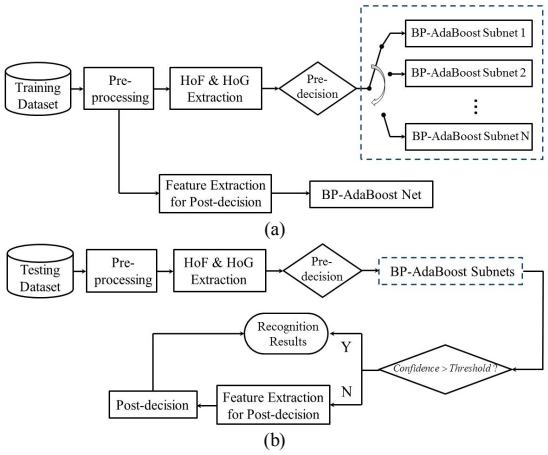


Fig. 1. Flow charts of our action recognition system. (a) The training phase. (b)The testing phase (the blue dashed rectangle means just the same as in (a)).

Fig. 1 (a) and (b) depict the flow charts of the training and testing phase of our system, respectively. The pre-processing procedure contains two steps — alignment and morphological processing. Then numerous features are extracted from each frame. To reduce the complexity and enhance accuracy in the training phase, we break down the whole training task into several subtasks through a data-mining procedure called pre-decision. However, the training for post-decision does not require such a procedure. Under a supervised learning framework, each training or sub-training task is accomplished by the BP (Back Propagation)-AdaBoost algorithm. The testing phase has much in common with the training phase. But pay attention that not all frames need post-decision — only those with low confidences do. Our benchmark dataset is the Weizmann dataset [9].

The remaining of this paper is organized as follows. The features for action recognition will be briefly introduced in Section 3. Then we elaborate on our machine learning approach in Section 4. The experimental results will be displayed and discussed in Section 5, followed by our conclusion in Section 6.

3. FEATURE EXTRACTION

Four kinds of frame-based features are extracted in this paper: Histograms of Optical Flow and Gradients (**HoF** and **HoG**), Hu's Moments (**HM**) and Block-Silhouettes (**BS**), where the latter two are only used for the optional post-decision procedure, considering the fact that accurate binary silhouettes are not always easy to obtain (especially under conditions of cluttered backgrounds or occlusions). Since the Weizmann dataset has already given the alignment masks and raw binary silhouettes, we do not bother to perform human detection and silhouettes extraction again.

To reduce the computational cost and improve the appearance of the raw binary silhouettes, the pre-processing procedure (i.e. alignment and morphological processing) is desirable before feature extraction (Fig. 2).

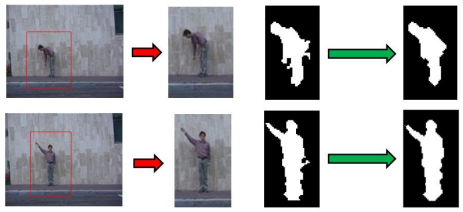


Fig. 2. Examples of pre-processing. (a) Alignment using given masks. (b) Morphological processing using "open" and "close" operations .

Inspired by [4], we divide the observation window into 4 sub-regions, and calculate HoF and HoG from each sub-region as well as the whole region (Fig. 3(a)). Such approach can preserve some locality of the histograms, which can further serve for the pre-decision procedure. To strike a balance between complexity and efficiency, we won't divide the observation window into too trivial sub-regions. Before generating the histograms, the original images and the calculated flow fields are blurred with a 5×5 Gaussian filter with variance 0.8. The perigon is evenly divided into 18 bins (20° for each bin), thus the final normalized HoF or HoG vector for one frame is $18\times5=90$ dimensional (Fig. 3(b) and 3(c)).

The modified binary silhouettes are used to generate HM and BS features. Hu's 7 moments are calculated from a series of symmetric expressions of the normalized centralized moments of order 2 and 3 [13]. To expect better discriminative power, we extract the HM both from the binary silhouettes and their skeletons (Fig. 4(a)), thus the HM vector is 14 dimensional. The BS feature is an approximation of the pixel-based silhouette [18], which uses a grid-based method and can largely diminish the storage requirement. The 28-dimensional BS vector comes from the normalized mean gray intensities in the sub-blocks by a column-scanned manner (Fig. 4(b)).

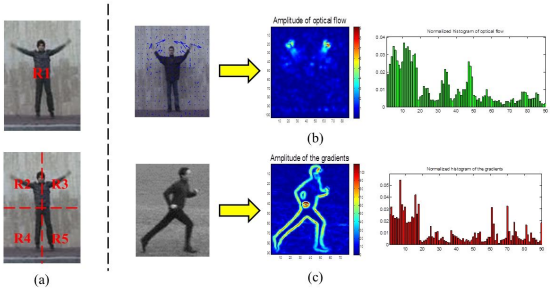


Fig. 3. (a) Our division of the observation window. (b) An example of HoF. (c) An example of HoG.

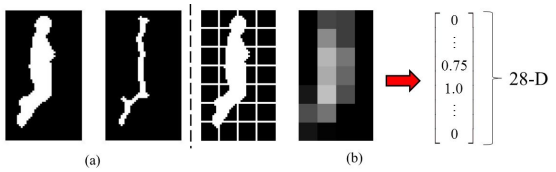


Fig. 4. (a) A binary silhouette and its skeleton. (b) Extraction of the BS feature.

4. BP-ADABOOST ALGORITHM UNDER A HIERARCHICAL RECOGNITION FRAMEWORK

4.1. Pre-decision

Unlike the traditional routine [4,14,15] that directly uses HoF vectors to train the classifiers, we propose an innovative data-mining method to detect Motion Salient Regions (MSR) from HoF before training (Fig. 5). By analyzing the distribution of the sub-histograms, we can know whether a region contains MSR through the comparison between the entropy of a sub-histogram and a pre-defined threshold, and can roughly pre-classify the video clips into several subcategories such as directional or non-directional actions, hands-related or feet-related actions (Table 1). Taking advantage of the pre-decision procedure, we can break the whole training task into several sub-training tasks, thus reducing the training complexity and confusion.

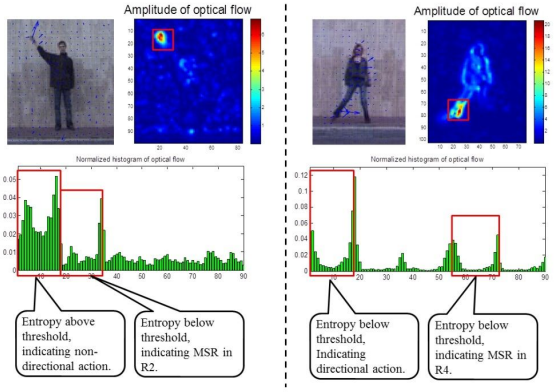


Fig. 5. Two examples of data-mining from the HoF vectors (red rectangles in the amplitude figures denote Motion Salient Regions (MSR)).

Table 1. Subcategories of the Pre-decision

Subcategory Number	Actions in the Subcategory	Directional Action	MSRs in R2 or R3	MSRs in R4 or R5
1	bend, jack, pjump	×	√	√
2	wave1, wave2	×	√	×
3	jump, run, side, skip walk	√	(not care)	√

4.2. BP-AdaBoost Algorithm

Although SVM and NN are both typical discriminative machine learning methods, the latter has the natural structure to cope with multiclass classification problems. We choose the most prevalent two-layer NN with sigmoid units as our basic classifier. In our experiments, the number of hidden units is 12 for single NN and 10 for boosted NN.

In general, BP-AdaBoost algorithm can be seen as an instance of applying the AdaBoost mechanism to the boosting of NN using BP training algorithm, which has much in common with standard AdaBoost except for a few details. We set the ideal

multiclass label in the form of $[0,...,0,1,0,...0]^T$ (the corresponding element indicating the class number is one and the rest elements are zeros). To measure the distance between the vector output $\mathbf{g}$ of NN and standard vector label $\mathbf{y}$, we can define a *confidence-distance* (Eq. (1)), which will later be used to update the sample weights.

$$conf\_dist(\mathbf{y},\mathbf{g}) = \begin{cases} \dfrac{\max(\mathbf{g}) - \max 2(\mathbf{g})}{\max(\mathbf{g}) + \max 2(\mathbf{g})}, & \text{if } \max\dim(\mathbf{g}) = \max\dim(\mathbf{y}) \\ \dfrac{true(\mathbf{g}) - \max(\mathbf{g})}{true(\mathbf{g}) + \max(\mathbf{g})}, & \text{if } \max\dim(\mathbf{g}) \neq \max\dim(\mathbf{y}) \end{cases} \quad (1)$$

In Eq. (1), $\max\dim(\bullet)$ returns the index of its parameter vector's largest element, $\max(\bullet)$ and $\max 2(\bullet)$ return the largest and second largest element of the parameter vector respectively, and $true(\bullet)$ returns the element in the correct index of its parameter vector. Similar with standard AdaBoost in the way that the weights of wrongly classified samples will grow, the sample weights can be updated by

$$D_t(i) = \frac{D_{t-1}(i)}{Z_t} \exp(-\alpha_t \cdot constrain(conf\_dist(\mathbf{y}_i, \mathbf{g}_t(\mathbf{x}_i)))), i = 1,...,N \quad (2)$$

where $D_t(i)$ is the $i-$th sample's weight in the $t-$th iteration; Z_t is a normalization factor; α_t is the weight of the $t-$th weak classifier; the function $constrain(\bullet)$ limits its parameter between -1 and +1 for fear that the weights of some difficult samples grow too fast.

Another issue should be pointed out is that the key to the boosting is the independence among the weak classifiers. It is easy to see that if all the classifiers are highly dependent, the AdaBoost method will fail because it is just a linear combination of similar classifiers, so the "strong classifier" won't outperform the best weak classifier. On the contrary, if all the classifiers are totally independent, the strong classifier could achieve arbitrarily good performance as long as the independent classifiers are enough. The actual situation of AdaBoost is between those two extreme cases: on the one hand, the weak classifiers can't be totally independent for they share the same whole training dataset; on the other hand, the weak classifiers can't be totally dependent for the training of each weak classifier is based on different sample weights. Therefore, though the AdaBoost method cannot achieve arbitrary accuracy, it can still considerably enhance the weak classifiers' performance.

In our approach, the training set of each weak classifier is just a subset of the whole training set chosen by *bootstrap method* [19], which treats the latest normalized sample weights as a priori probability mass function and performs *put-back sampling* N times (N is the size of training set). The bootstrap method brings in two benefits: (1) the training complexity is reduced since the training sets are just parts of the whole set; (2) loose independence of weak classifiers is guaranteed for they share different training subsets with variable sample weights.

Now that all the key points of BP-AdaBoost are discussed, readers can fully understand our algorithm listed in Algorithm 1.

Algorithm 1. BP-AdaBoost

Input: Training set $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ ($\mathbf{y}_i = [0,...,0,\underset{(c)}{1},0,...,0]^T$, for class c); weak classifier NN with known structure; the iteration number T.

Output: Strong classifier $\mathbf{H}(\mathbf{x})$.

1) Initialize the weights of all the training samples: $D_0(i) = 1/N, i = 1,...,N$;

2) ***for*** $t = 1$ to T ***do***

a) Use $\{D_{t-1}(i)\}_{i=1}^N$ as a priori probability mass function and choose a training subset of size N (with repeated samples) through bootstrap method;

b) Use the training subset to train the $t-$th NN $\mathbf{g}_t$ by standard BP algorithm;

c) Calculate the weighted error of the $t-$th NN over the whole training set:

$$e_t = \sum_{\max\dim(\mathbf{g}_t(\mathbf{x}_i)) \neq \max\dim(\mathbf{y}_i)} D_{t-1}(i), i = 1,...,N;$$

d) Calculate the weight of the $t-$th NN: $\alpha_t = \frac{1}{2}\ln\frac{1-e_t}{e_t}$;

e) Update the samples' weights:

$$D_t(i) = \frac{D_{t-1}(i)}{Z_t} \exp(-\alpha_t \cdot constrain(conf\_dist(\mathbf{y}_i, \mathbf{g}_t(\mathbf{x}_i)))), i = 1,...,N,$$

where Z_t is a normalization factor to ensure $\sum_{i=1}^N D_t(i) = 1$;

end for

3) ***if*** $\alpha_t < 0$ ***then*** $\alpha_t = 0$;

4) Obtain the strong classifier $\mathbf{H}(\mathbf{x}) = \mathbf{f}[\sum_{t=1}^T \alpha_t \mathbf{f}(\mathbf{g}_t(\mathbf{x}))]$, where $\mathbf{f}$ is an activation function that sets the input's largest dimension 1 and the rest dimensions 0.

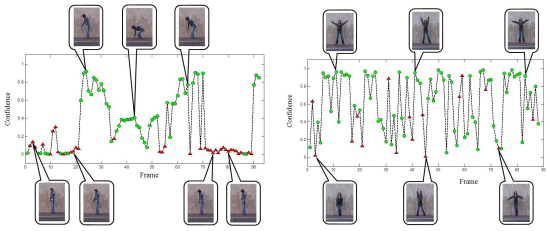
4.3. Post-decision

This module, whose complexity depends, is optional but strongly recommended if high performance is the primary objective. Here HM and BS features are used for post-decision. To maintain consistency, this module also adopts boosted NN as the classifier. Note that not all video sequences need post-decision — only those whose final confidences are below 0.35 do. In that case, the results of post-decision are also considered to make the final decision.

5. EXPERIMENTAL RESULTS

5.1. Action Recognition Based on Single Feature without Boosting

In this section, we test the discriminative power of single feature based on SVM and NN without boosting, expecting to show the merits of NN. Since both SVM (1-vs-all) and NN can provide confidence information, we exploit it to discard the low-confidence frame-based results to improve the recognition accuracy. As shown by Fig. 6, the frames containing key poses usually have high confidences, whereas the frames containing not very discriminative poses usually have low confidences and may very well be wrongly classified. In our experiments, frames with confidences below 0.3 are discarded.



(a) (b)

Fig. 6. Confidence of each frame in sample video clips. The horizontal coordinate is the frame number. The correctly classified frames are denoted by green circles whereas the wrongly classified ones are denoted by red triangles. (a) Sample video clip "bend"; (b) Sample video clip "jack".

The average recognition accuracy and the training time of SVM and NN are displayed in Fig. 7(a) and 7(b), respectively. The results clearly show that NN outperforms SVM both in accuracy and training time using whatever feature. We can see that the BS feature, which is not always easy to obtain, performs best. The HoF and HoG features have comparable performance,

both outperforming HM feature using SVM. However, it is very interesting that the improvement of NN+HM over SVM+HM is the most significant (average accuracy increases by 5.6%), which again demonstrates the advantages of NN over SVM.

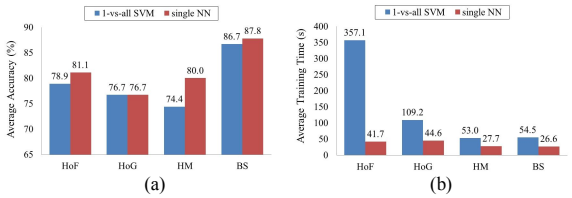


Fig. 7. Recognition results based on different combinations of feature and classifier. (a) Average accuracy; (b) Average training time.

5.2. Action Recognition Based on BP-AdaBoost

This section compares the performance of single NN and boosted NN (Table 2). Here we just use the HoF and HoG features, whereas the features for post-decision will be used in the next section. As one can imagine, the training time of boosted NN (12 weak classifiers) is longer than single NN (still tolerable), but the former achieves higher accuracy. From the last line of Table 2 we can see that the boosted NN using both HoF and HoG feature achieves best performance (corresponding confusion matrix is shown in Fig. 10(a)), which implies the complementarity of the motion feature and the shape feature. The details of the updating of the sample weights are depicted in Fig. 8.

Table 2. Comparison of Single NN and Boosted NN

		Average Accuracy (%)	Average Training Time (s)
HoF	NN	81.1	41.7
	boosted NN	85.6	188.8
HoG	NN	76.7	44.6
	boosted NN	84.4	205.7
HoF+HoG	boosted NN	**87.8**	≤188.8+205.7

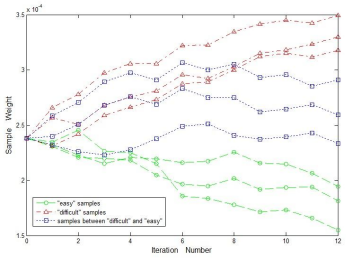


Fig. 8. Updating of some typical sample weights: (1) "easy" samples whose weights decrease most of the time; (2) "difficult" samples whose weights increase most of the time; (3) samples between "easy" and "difficult" whose weights fluctuate during the whole training procedure.

5.3. Action Recognition Based on BP-AdaBoost under a Hierarchical Recognition Framework

We will add the pre-decision and post-decision modules to the system in this section. It can be seen from Fig. 9(a) that the training time is considerably reduced when the whole training task is broken into several subtasks, and from Fig. 9(b) that better performance can be achieved using a hierarchical recognition framework. Note that the sub-training tasks are independent and can be carried out parallelly if conditions permit, thus the training time would be no more than the longest subtask.

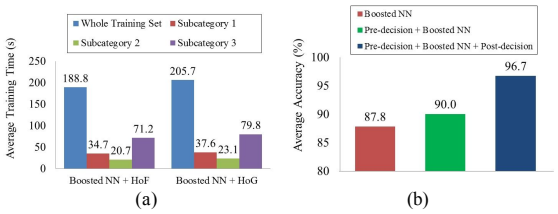


Fig. 9. (a) Average training time of boosted NN without and with Pre-decision (subcategorie numbers are consistant with Table 1); (b) Average accuracy of boosted NN using different system structure.

The highest accuracy (corresponding confusion matrix is shown in Fig. 10(b)) is achieved by combining BP-AdaBoost with the complete hierarchical recognition framework, which lives up to our expectations. A comparison of our approach and other state-of-the-art methods is listed in Table 3.

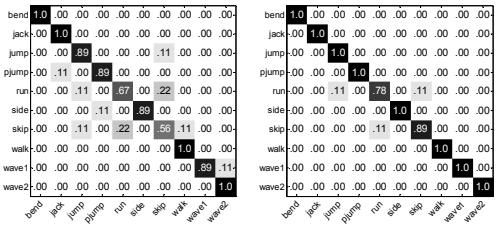


Fig. 10. Confusion matrices. (a) Boosted NN; (b) Boosted NN under complete hierarchical recognition framework.

Table 3. Action Recognition Accuracy of Various Methods

Method	Lertniphonphan *et al.* [4]	Li X. [16]	Kai G. *et al.* [5]	Grundmann *et al.* [7]	Junejo I.N. *et al.* [20]	**Proposed Method**
Average Accuracy (%)	79.2	91.1	94.4	94.6	94.9	**96.7**

6. CONCLUSION & FUTURE WORK

In this paper, we explore the performance of NN for human action recognition and design a novel hierarchical and boosting-based action recognition system. Our work is based on prior studies but makes new contributions as well. We exploit multiple prevalent motion and shape features (i.e. HoF [4], HoG [6], HM [13] and BS [18]) and modify them to enhance their efficiency. Although most researchers use SVM [6,12,14,15] as the discriminative classifier, we adopt the boosted NN, during which we extend the standard binary AdaBoost algorithm to a multiclass version. In addition, we propose a innovative hierarchical recognition framework with pre-decision and post-decision, which breaks down the whole training task into several subtasks by data mining, and proves to be superior to the conventional routine that directly uses the HoF feature to train the global classifier [4,14,15].

Experiments on the Weizmann dataset show promising results of our approach in comparison with other state-of-the-art methods. We believe that our BP-AdaBoost algorithm and the hierarchical recognition framework can enrich the machine learning methodology, and can be conveniently transplanted to other recognition tasks. However, our dataset is relatively simple, so exploring how to effectively recognize human actions in cluttered backgrounds is our future work.

Acknowledgement: This work is supported by National Natural Science Foundation of China (NSFC) under Grant No. 60971098 and Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institutions.

7. REFERENCES

[1] Ronald Poppe, "A survey on vision-based human action recognition", *Image and Vision Computing*, Vol. 28, pp. 976-990, 2010.

[2] Turaga P. and Chellappa R., "Machine Recognition of Human Activities: A Survey", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 18 (11), pp. 1473-1488, 2008.

[3] Efros A.A., Berg, A.C., Mori, G., Malik, J., "Recognizing Action at a Distance", 9th *IEEE International Conference on Computer Vision*, Vol. 2, pp. 726-733, 2003.

[4] Lertniphonphan K., Aramvith S., Chalidabhongse T.H., "Human Action Recognition using Direction Histograms of Optical Flow", 11th *International Symposium on Communications and Information Technologies (ISCIT)*, pp. 574-579, 2011.

[5] Kai G., Ishwar P., Konrad J., "Action Recognition Using Sparse Representation on Covariance Manifolds of Optical Flow", 7th *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 188-195, 2010.

[6] Laptev I., Marszalek M., Schmid C., Rozenfeld B., "Learning realistic human actions from movies", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[7] Grundmann M., Meier F., Essa I., "3D Shape Context and Distance Transform for Action Recognition", *International Conference on Pattern Recognition*, pp. 1-4, 2008.

[8] Yilmaz A. and Shah M., "Actions Sketch: A Novel Action Representation", *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 984-989, 2005.

[9] Blank M., Gorelick L., Shechtman E., Irani M., Basri R., "Actions as Space-Time Shapes", *IEEE International Conference on Computer Vision*, Vol. 2, pp. 1395-1402, 2005.

[10] Laptiv I., "On Space-Time Interest Points", *International Journal of Computer Vision*, Vol. 64(2/3), pp. 107-123, 2005.

[11] Dollar P., Rabaud V., Cottrell G., Belongie S., "Behavior Recognition via Sparse Spatio-Temporal Features", *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 65-72, 2005.

[12] Schuldt C., Laptev I., Caputo B., "Recognizing Human Actions: A Local SVM Approach", *International Conference on Pattern Recognition*, Vol.3, pp. 32-36, 2004.

[13] Bobick A.F. and Davis, J.W., "The Recognition of Human Movement Using Temporal Templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23(3), pp.257-267, 2001.

[14] Mahbub U., Imtiaz H., Ahad M.A.R., "An Optical Flow Based Approach for Action Recognition", *International Conference on Computer and Information Technology (ICCIT)*, pp. 646-651, 2011.

[15] Imtiaz H., Mahbub U., Ahad M.A.R., "Action Recognition Algorithm Based on Optical Flow and RANSAC in Frequency Domain", *SICE Annual Conference*, pp. 1627-1631, 2011.

[16] Li X., "HMM based action recognition using oriented histograms of optical flow field", *Electronics Letters*, Vol. 43 (10), pp. 560-561, 2007.

[17] Ahmad M. and Seong-Whan L., "HMM-based Human Action Recognition Using Multiview Image Sequences", 18th *International Conference on Pattern Recognition*, Vol. 1, pp. 263-266, 2006.

[18] Liang W. and Suter D. "Recognizing Human Activities from Silhouettes: Motion Subspace and Factorial Discriminative Graphical Model", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

[19] Jiawei H. and Kamber M., "Data Mining: Concepts and Techniques" (second edition), copyright by *Elsevier (Singapore) Pte Ltd.*, 2007.

[20] Junejo I.N., Dexter E., Laptev, I., Pérez P., "View-Independent Action Recognition from Temporal Self-Similarities", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33 (1), pp. 172-185, 2011.