Faster Human Activity Recognition with SVM

K. G. Manosha Chathuramali[#1], Ranga Rodrigo[#2]
[#1]*Engineering Research Unit, University of Moratuwa, Sri Lanka.*
mashi.gamage@gmail.com
[#2]*Department of Electronic and Telecommunication Engineering, University of Moratuwa, Sri Lanka.*
ranga@ent.mrt.ac.lk

***Abstract*—Human activity recognition finds many applications in areas such as surveillance, and sports. Such a system classifies a spatio-temporal feature descriptor of a human figure in a video, based on training examples. However many classifiers face the constraints of the long training time, and the large size of the feature vector. Our method, due to the use of an Support Vector Machine (SVM) classifier, on an existing spatio-temporal feature descriptor resolves these problems in human activity recognition. Comparison of our system with existing classifiers using two standard datasets shows that our system is much superior in terms of the computational time, and either it surpasses or is on par with the existing recognition rates. It performs on par or marginally inferior to existing systems, when the number of training examples are a few due to the imbalance, although consistently better in terms of computation time.**

Keywords: Silhouette, normalized bounding box, optic flow, SVM, label activities, activity recognition

I. Introduction

Human activity detection is a challenging, unsolved problem [1], [2], even though great efforts have been made. Human motion analysis in computer vision involves detecting, tracking and recognition of human activities [2]. This has a wide range of promising applications. Some examples are security surveillance, human machine interaction, sports, video annotations, medical diagnostics and entry, exit control. However it remains a challenging task to detect human activities, because of their variable appearance and wide range of poses that they can adopt [1].

An activity can be represented as a set of features. Optic flow is a pattern of visible motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene [3]. At low spatial resolution when limbs cannot be identified, flow fields are discriminative for a range of motions [4]. At higher spatial resolutions, can be recovered body configuration [5]. It is shown that 3D configuration can be inferred from 2D images [6] which propose building appearance features for body configuration. There are many such feature extraction methods that have shown to be successful in activity detection. They are: characterize spatio-temporal volumes [7]; spatio-temporal interest points [8]; and silhouette histogram of oriented features [9]. The features are usually synthesized into a descriptor. These descriptors: Histogram of Oriented Gradient (HOG) descriptors [10], SIFT descriptors [11] and shape contexts [12] are most popular in this area. Generally these descriptors encode what the body looks like and some context of motion. We identified that background subtraction methods are commonly used and effective in feature extraction process and as well as flow fields are discriminative for a range of motions.

Among several methods of feature extraction, we used a frame descriptor which is a histogram of silhouette and the optic flow inside a normalized bounding box [1]. Combining these histograms gives a very rich feature vector confirming that the combination of background subtraction with flow fields are discriminative. We will verify it further, later in the paper. This descriptor has been used by Du Tran *et al.* [1] and achieved accurate results using metric learning.

An activity recognition method should mostly have the following properties to evaluate with a reasonable accuracy and time. *Robustness* [1]: features should be relatively straightforward to obtain from image sequence with acceptable accuracy and should demonstrate good noise behavior even in cluttered backgrounds under difficult illumination. *Discriminative nature*: methods must focus on what is important about the relationship between body configurations and activities. In human activity recognition, discriminative methods have been achieved success [1], [10] *Reliability*: the results achieved by the activity recognition method should be accurate. *Cost Effectiveness*: the algorithm must not be computationally expensive even though a high level algorithms may be used to detect features. These requirements are highly demanding. However there is evidence that we can meet them because of special properties of activity data. One of them is focusing on the key poses, which can capture the essence of an action class, even if there is variance in execution styles of the same action [8]. Second, labelling motion capture data with activity labels is straightforward and accurate [13]. Third, categorizing of human motion using hybrid of spatial-temporal and static features [14]. All these clearly suggest that appropriate motion data can be classified, because different activities tend to seem strongly different in the feature descriptor space. Therefore, Support Vector Machine (SVM) algorithm should be able to learn to classify activities, in a discriminative feature space.

In this paper we propose a SVM classifier for activity recognition within a high dimensional feature space. We consider about two major objectives: first, labelling activities and second, learning with few examples. We use multi class SVM classifier since our datasets include multiple activities done by different actors. We choose leave-one-out cross validation technique for our classification with a variety of protocols to

978-1-4673-5530-8/12/$31.00 ©2012 IEEE

label the activities. We show that our method performs well in labelling activities with a much lower computational cost, in comparison with metric learning. When attempted to learn with a few examples, our system preforms poorly, given the nature of SVM.

This paper is organized as follows: section 2 gives overview of our method, in which we describe about the feature extraction method. Classification methods and SVM classifier are briefly described in section 3. Descriptions about the datasets that we used and evaluation methodology are given in section 4. Experimental results are shown in section 5 and we show comparison results with other reported methods. Finally, we gives a brief discussion about our approach and future directions in section 6.

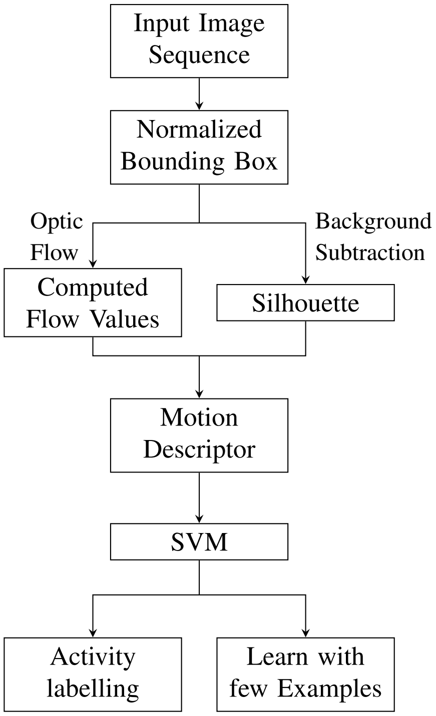

Fig. 1. An Overview of the feature extraction method and activity recognition chain. The feature extraction window first scales the normalized bounding box values from the input image sequence. Using these scaled images, the generated silhouette of each image by background subtraction method and optic flow measurements are split into horizontal and vertical channels. These combined vectors from the motion context feature descriptor. Then the SVM classifier is used for the activity recognition process.

II. Overview of the Method

We represent the feature extraction and activity recognition chain in Figure 1. First step is extracting the image sequence from the video and calculating the normalized bounding box coordinates. Then we extract the silhouettes of characters (actors) using background subtraction. Generate the optic flow values by using the algorithms Lucas-Kanade [15]. Then we concatenate both optic flow values and histogrammed silhouette to produce the motion descriptor. Finally, we classify the activities by using SVM classifier.

Feature extraction can be categorized into two stages: First, local features extracted from each frame. Second, global features are found through activity sequence, comprising of several frames. Computing such motion descriptors centered at each frame will enable us to compare frame sequences from different sequences based on local motion characteristics.

A. Local Features

A local feature is a histogram of the silhouette of the actor and of the optic flow inside the normalized bounding box, as used by Du Tran *et al.* [1]. The bounding box is scaled in to an $N \times N$ square box and is placed at the center bottom of the box. See Figure 2. This box is used for silhouette extraction and to resample the flow vectors. The optical flow vector field F is first split into two scalar fields corresponding to the horizontal and vertical components of the flow, Fx and Fy. To compute the optic flow values we use Lucas-Kanade algorithm [15]. Each channel is smoothed by median filter to reduce the effect of noise.

Two real-valued channels Fx and Fy and binary channel silhouette are the three channels which constitute the histogram. Each of these channels is histogrammed using the following technique: First, we divide the normalized bounding box into 2 × 2 sub-windows and then each sub-window is divided into 18 pie slices covering 20 degrees each. These pie slices do not overlap and the center of the pie is in the center of the sub-window. The values of each channel are integrated over the domain of every slice. The result is a 72 $(2 \times 2 \times 18)$-dimensional histogram. By concatenating the histograms of all 3 channels we get a 216-dimensional frame descriptor [1]. Please refer to Du Tran *et al.* [1], for more details.

B. Motion Descriptor

The most important question is what are the appropriate features to be put in to the motion descriptor. This feature extraction method can be adopted to capture very rich representations by incorporating static and dynamic features. That means, it can capture local appearance and local motions of a person.

Following Du Tran *et al.* [1], we too use 15 frames and split them into 3 blocks of 5 frames named as past, current and future. The frame descriptors of each block are stacked together into a 1080-dimensional vector. This block descriptor is then projected onto the first N principal components using Principal Component Analysis (PCA) [16]. The resulting 70-dimensional context descriptor is appended to the current frame descriptor to form the final 286-dimensional motion context descriptor [1].

III. Action Classification Methods

Classification is a task of assigning a label to a new instance from a group of known instances called the class. Our instances, as described in the previous section, are activities characterized by the silhouette and optic-flow histograms of an $N \times N$ box over a sequence of 15-frames. We use several classifiers to carry out activity recognition, including nearest

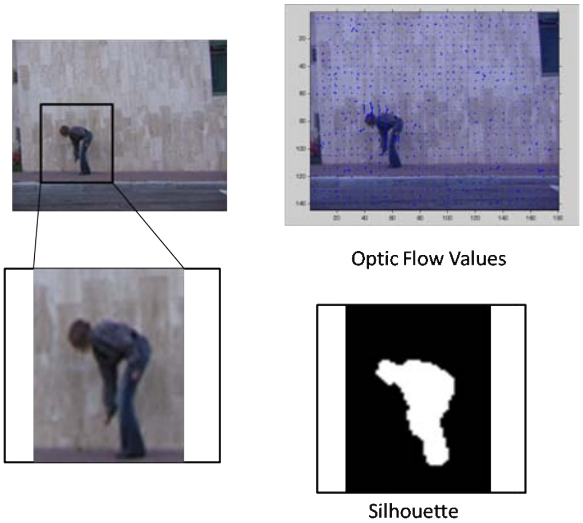

Fig. 2. A graphical representation of the feature extraction. The silhouette and optic flow values are extracted from the normalized bounding box image.

neighbor (1-NN), and SVM. We briefly describe each classifier in the following text. Among several discriminative classifiers we will give brief description about selected classifiers below.

A. Nearest Neighbor

Nearest neighbor classifier is one of the simplest methods for a feature vector. The distance is calculated between the new feature vector and every vector of the training set. Any distance measure can be used for this purpose. Here 1-NN classifier assigns a label to every query frame by finding the closet neighbor among training action descriptors. Every frame (actually, the feature descriptor that corresponds to the frame) of the query sequence votes for the label of the sequence and the label is determined by the majority. We calculate the Euclidean Distance for 1-NN classification.

B. 1-Nearest Neighbor with Metric Learning

Nearest neighbor classification method depends crucially on the distance metric used to identify nearest neighbors. Most k-NN classifiers use simple Euclidean distances to measure the dissimilarities between a new feature vector and every vector in the training set. Euclidean distance metrics do not capitalize on statistical regularities in the large training set of labelled examples. The metric learning algorithm [17] addresses the above problem and comes up with Large Margin Nearest Neighbors (LMNN). This method is especially designed for k-NN classifiers. We give brief explanation to LMNN below,:

LMNN learns a Mahalanobis distance D between vectors or points x_i and x_j, :

$$D(x_i, x_j) = (x_i - x_j)^T M (x_i - x_j) = ||L(x_i - x_j)||^2 \quad (1)$$

where the matrix $M = L^T L$, Mahalanobis distance metric induced by the linear transformation L. That means LMNN maximizes the distances between examples with different labels and minimizes the distance between closet example with the same label.

Minimize:

$$\sum_{ij} \eta_{ij} (x_i - x_j)^T M (x_i - x_j) + c \sum_{ijl} \eta_{ij} (1 - y_{il}) \xi_{ijl}$$

Subject to:

$$(x_i - x_l)^T M (x_i - x_l) - (x_i - x_j)^T M (x_i - x_j) \geq (1 - \xi_{ijl})$$

$$\xi_{ijl} \geq 0$$

$$M \succeq 0$$

where y_{ij} is a binary value indicating whether points x_i and x_j are in the same class. η_{ij} is a binary value indicating whether the selected x_j is the closet neighbor of x_i with the same class. And ξ_{ijl} are slack variables for all pairs of differently labeled inputs. The last constraint $M \succeq 0$ represented that the matrix M is required to be positive semidefinite.

C. Support Vector Machines (SVM)

SVM is very competitive within the existing classification methods in many areas and relatively easy to use [18]. SVM performs classification using linear decision hyperplanes in the feature space. During training, the hyperplanes are calculated to separate the training data with different labels. Using kernel function SVM has many extensions, regression, density estimation and kernel PCA. If the training data are not linearly separable, a kernel function is used to transform the data into a new space. The data have to linearly separable in the new vector space. SVMs scale well for very large training sets and perform well with accurate results cost effectively. The complexity for training increases with the number of training samples; however, the classification is independent of it.

Separating hyperplanes for linear classification can be represented simply as follow:

$$y = \mathrm{sgn}((w \cdot x) + b) \quad (2)$$

This can be upper bounded in terms of margin. For separating hyperplane classifier, the condition for classification without training error is

$$y((w \cdot x) + b) \geq 1, i = 1, \ldots, n \quad (3)$$

The goal of this learning method is to formulate the optimal hyperplane.

Minimize:

$$\frac{1}{2} ||w||^2 \quad (4)$$

Subject to:

$$y((w \cdot \phi(x_i)) + b) \geq 1, i = 1, \ldots, n \quad (5)$$

where $\phi(x_i)$ substitutes for each training example x_i , since the linear functions are not good enough for some problems [18].

1) Multi-Class SVM: Although support vector machines were originally developed for binary classification, it can be effectively extended for multi class classification [19]. Basically there are two types of techniques for multi-class SVM. First type is constructing and combining binary classifiers in a certain manner to form a multi-class classifier. The second method is directly considering all the classes of data in one optimization formulation. We experienced that it is computationally more expensive to solve a multi class problem than a binary problem with the same number of data. *One-against-rest method* needs N SVM classifiers for an N-class classification problem [19]. Training process takes much time. One-against-one method needs $\frac{N(N-1)}{2}$ classifiers, each of which is trained on samples from the two corresponding classes. Compared with the one-against-rest method, the classification accuracy is improved and computationally less expensive [19]. We used LIBSVM [20] of which the classifier prediction is made by a collection of one-against-one SVM classifiers for our experiment. We compared our results with other classification methods.

IV. Dataset and Evaluation Methodology

A. Description of Datasets

TABLE I
THE VARIATIONS IN THE ACTIVITY DATASET

Dataset	Actors	Actions	Sequences	views
Weizmann	9	10	93	1
UIUC1	8	14	532	1

For our experiment we used two standard datasets summarized in the Table I. Weizmann and UIUC1 datasets represent the multiple actors, actions and number of sequences. In Weizmann dataset there is only one instance of activity per actor and UIUC1 has extensive repetitions, same activity by the same actor.

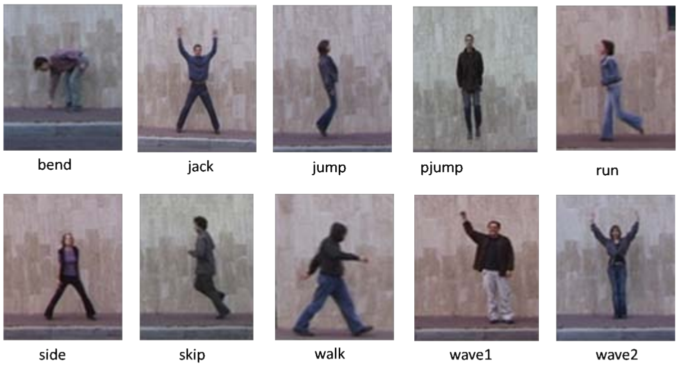

Fig. 3. Sample frames from each action in weizmann dataset [7].

1) Weizmann Dataset: The Weizmann human action dataset [7] contains 10 types of human actions performed by 9 different people. Each actor performs every action, giving 10×9 classes. There are isolated 93 different sequences with three extra sequences. The snapshots of action categories are shown in Figure 3. This dataset is a low-resolution (80px) dataset.

2) UIUC1 Dataset: UIUC1 is a high resolution (300px) dataset collected by Du Tran *et al.* [1]. It contains 8 different actors and 14 human actions. There are 532 sequences due to repeating the activities by the actors.

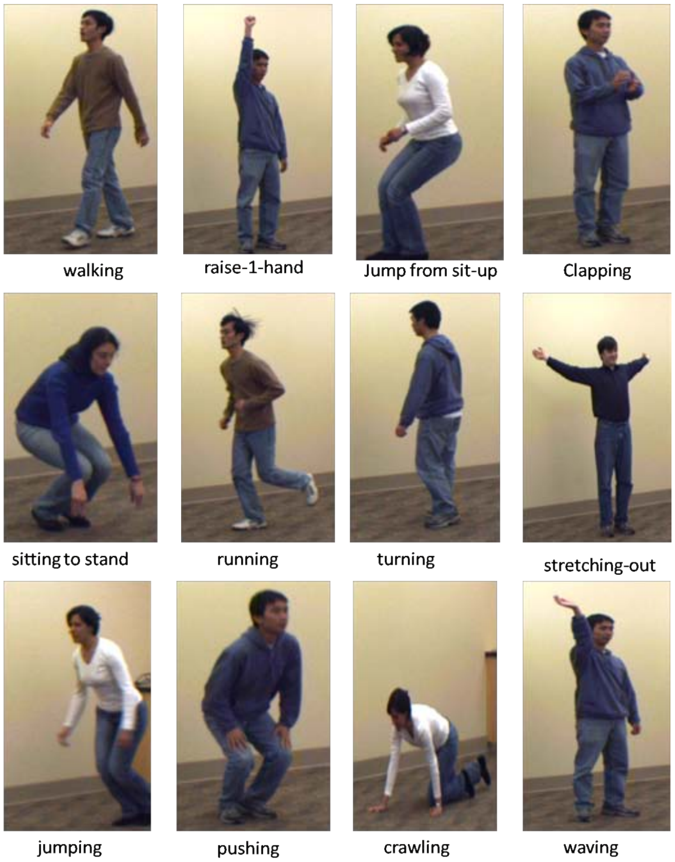

Fig. 4. Sample frames from some actions in UIUC1 dataset [1].

B. Evaluation Methodology

To perform action classification using the aforementioned motion descriptor, we train a multi-class SVM classifier with labeled action descriptors. In the training phase, each binary SVM classifier leads to an inequality constrained quadratic optimization problem. Because of the nonlinear relation between action classes and histogram features in the descriptor, we choose radial basis function (RBF) kernel for our SVM classifier [21].

To estimate the best classifier for our datasets, we carry out a grid search in the space of parameter C and γ. Here, C is the weight of error penalty and γ determines the width of the RBF kernel. The appropriate SVM classifier is selected by the set of (C,γ) which maximizes the cross-validation rate in the space of search, which, in turn, increases the accuracy of the results [21].

We evaluate the accuracy of the activity label prediction for a query sequence. We consider every sequence in a dataset as a query sequence: We use the technique of leave-one-out cross validation, where one single action sequence is selected for a testing at a time, using the rest as training examples. We use different protocols based on the leave-one-out method used by Du Tran *et al.* [1], which enables us to compare our results with theirs. The used protocols are as follows: Leave One Actor Out (L1AO) removes all sequences of the same actor from the training set and measures prediction accuracy. Leave One Actor-Action Out (L1AAO) removes all the sequences of the query activity performed by the query actor from the training set and measures prediction accuracy. Leave One Sequence Out (L1SO) removes only the query sequence from the training set. This protocol is equivalent to L1AAO, if an actor performs every actions once.

The few examples (FE-K) protocol is slightly different from the previous protocols. It allows K examples of the activity of the query sequence to appear in the training set. The actors of the query sequences are required to be different from the training examples. We report the accuracies at $K = 1, 2, 4, 8$.

V. Results

A. Experimental Results

We compared our activity classifier with 1-NN and 1-NN with metric learning using two standard datasets. Our algorithm outperforms or on-par with existing results, except for the case of training with a few examples.

TABLE II
Percent recognition rates for training with regular training sets: Our Algorithm outperforms or is on par with the exciting results.

Dataset	Algorithm	Discriminative task		
		L1SO	L1AAO	L1AO
Weizmann	1-NN	95.7	95.7	96.77
	1-NN (Metric)	100	100	100
	SVM	100	100	100
UIUC1	1-NN	98.87	97.74	98.12
	1-NN (Metric)	99.06	97.74	98.31
	SVM	99.04	98.04	98.84

Table II represents that our approach achieves state-of-the-art discriminative performance compared to metric learning and 1-NN. The time taken to UIUC1 dataset, that includes 42800 training exapmle and 200 testing examples, is 5 to 6 minutes. Killian *et al.* [17] indicate that, metric learning takes 4 hours, for a dataset including 60000 training examples and 10000 testing examples 4 hours. For these results we used complete feature space because it improves the recognition accuracy. Du Tran *et al.* [1] used reduced dimensionality for LMNN because it is computationally very expensive to use the complete feature vector. In this context, our method is superior due to the ability to use a high dimensional feature vector and generating more accurate results in much less time in the order of minutes.

Table III gives the results of learning with few examples. It is a significantly more difficult [1] task and our results are wither comparable or slightly poor in performance. Since we use the whole training set, it is quite out of balance when FE-$K = 1, 2, 4$. SVM classifiers generally perform poorly on imbalanced datasets because they are designed to generalize from sample data and output the simplest hypothesis that best fits the data [22]. We verify the same through our results as shown in Table III.

TABLE III
Percent recognition rates for training with a few training examples: Our Algorithm is slightly poor compared to the exciting results.

Dataset	Algorithm	Few Examples			
		FE-1	FE-2	FE-4	FE-8
Weizmann	1-NN	53.00	73.00	89.00	96.00
	1-NN (Metric)	72.31	81.77	92.97	100
	SVM	48.81	66.67	70.24	100
UIUC1	1-NN	58.70	76.20	90.10	95.00
	1-NN (Metric)	88.80	94.84	95.63	98.86
	SVM	40.74	45.56	80.65	97.45

TABLE IV
Percent recognition rates using the Weizmann Dataset

Method	Recognition rate
Our Method	100
Du Tran *et al.* [1]	100
Chen *et al.* [21]	100
Blank *et al.* [7]	98.8
Hutan and Duygulu [23]	92.0

Table IV compares the results of weizmann dataset with different feature extraction method and different classification methods used. Chen *et al.* [21] used HOG descriptors with SVM classifier.

VI. Conclusion

In this paper, we presented the SVM classification for human activity recognition. Using SVM is quite computationally cost effective in comparison with metric learning. It gives better results for activity recognition and for high dimensional vector space within five to ten minutes. The computational time increases when the number of training examples increase.

In our experiments we experienced that SVM classifier performs poorly on imbalance training sets. Our system performs poorly, in terms of recognition rate, when the number of training examples is a few. Increasing the number of examples in the training set increase the accuracy of the results. Using a few examples, we showed that the imbalance of the training set gives rise to poor recognition results. We verified that, in the case of activity recognition with few examples, the SVM classifier performs marginally inferior to the existing results. However, our system is consistently superior in regard to computational time.

When actions are being observed, available visual cues form human figures are usually sparse and vague. Therefore, action recognition algorithms require an exact description of human shapes and performed motion descriptors. Our selected feature extraction is able to tolerate these conditions to a reasonable degree. The feature extraction method seems to be tolerant

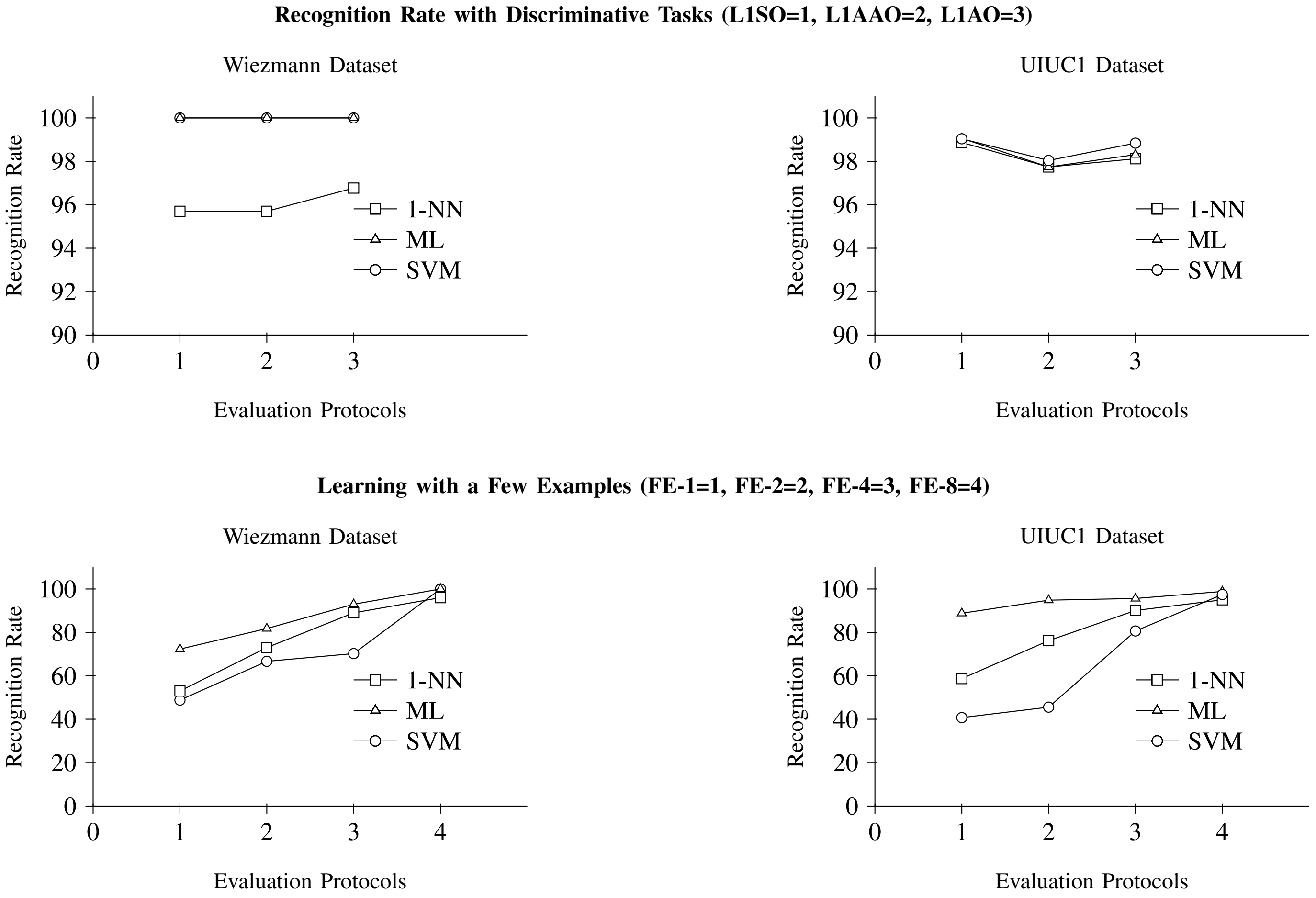

Fig. 5. Comparison results of three classifiers for Weizmann and UIUC1 datasets are summarized. Recognition rate is higher in SVM for labelling activity with three discriminative tasks. However, learning with few examples shows performance poorly than other classifiers as shown in the nature of SVM.

to some level of noise. Based on the nature of datasets used, there is evidence that our method works well in both low and high resolution images.

Our human activity recognition system is limited to a few action categories. However, the number of primitive activities that people can name and learn is not limited to a few. There are plethora of sports, dance and special activities that can be categorized. Each of these has dozens of distinct specialized motions known to experts. The background may not be the same for every activity that people can adopt. Building methods that can cope successfully with activities that have not been seen before is the key to making application of activity recognition feasible in real scenarios. Taking into consideration different backgrounds for image sequences and how to work with them to recognize human activities too need much work.

References

[1] D. Tran, A. Sorokin, and D. Forsyth, "Human activity recognition with metric learning," in *Proceedings of the European Conference on Computer Vision*, ser. LNCS 5302, vol. Part I. Marseille, France: Springer-Verlag Berlin Heidelberg, 2008, pp. 549–562.

[2] J. K. Aggarwal and Q.Cai, "Human motion analysis: A review," *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 428–440, March 1999.

[3] D. Warren and E. R. Strelow, *Electronic Spatial Sensing for the Blind: Contributions from Perception, Rehabilitation, and Computer Vision*. Springer, 1985.

[4] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in *Proceedings of the IEEE International Conference on Computer Vision*, Nice, France, October 2003, pp. 726–733.

[5] D. Ramanan and D. A. Forsyth, "Automatic annotation of everyday movements," in *Proceedings of the Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2003.

[6] N. R. Howe, M. E. Leventon, and W. T. Freeman, "Bayesian reconstruction of 3d human motion from single-camera video," in *Proceedings of the Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2000, pp. 820–826.

[7] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Proceedings of the IEEE International Conference on Computer Vision*, Beijing, China, October 2005, pp. 1395–1402.

[8] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and viterbi path searching," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, USA, June 2007, pp. 1–8.

[9] Y. Sheikh, O. Javed, and T. Kanade, "Background subtraction for freely moving cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, Kyoto, Japan, September-October 2009, pp. 1219–1225.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on*

Computer Vision and Pattern Recognition, vol. 1, INRIA Rhone-Alps, Montbonnot, France, June 2005, pp. 886–893.

[11] S. Belongie, J. Malik, and J. Puzicha, "Matching shapes," in *Proceedings of the IEEE International Conference on Computer Vision*, Vancouver , Canada, July 2001, pp. 454–461.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[13] O. Arikan, D. A. Forsyth, and J. F. O'Brien, "Motion synthesis from annotations," in *Proceedings of the Special Interest Group on Graphics and Interactive Techniques*, San Diego, California, USA, July 2003, pp. 402–408.

[14] J. C. Niebles and L. Fei-Fei, "A hierarchical model of shape and appearance for human action classification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, USA, June 2007, pp. 1–8.

[15] B. D. Lucas and T. Kanade, "An iterative image registration technique with an applica- tion to stero vision," in *Proceedings of the International Joint Conferences on Artificial Intelligence*, 1981, pp. 674–679.

[16] C. M. Bishop, *Pattern Recognition and Machine Learning*, J. K. Michael Jordan and B. Scholkopf, Eds. LLC, 233 Spring Street, New York, NY 10013, USA: Springer, 2006.

[17] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Proceedings of the Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2006.

[18] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Netwroks*, vol. 12, no. 2, pp. 181–201, March 2001.

[19] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Netwroks*, vol. 13, no. 2, pp. 415–425, March 2002.

[20] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[21] C.-C. Chen and J. K. Aggarwal, "Recognizing human action from a far field of view," *IEEE Workshop on Motion and Video Computing*, December 2009.

[22] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Proceedings of the IEEE European Conference on Machine Learning*, Pisa, Italy, September 2004, pp. 39–50.

[23] K. Hutan and P. Daygulu, "Pose sentences: A new representation for action recognition using sequence of pose words," in *Proceedings of the IEEE International Conference on Pattern Recognition*, Tampa, Florida, USA, December 2008, pp. 1–4.