HUMAN ACTION RECOGNITION USING LOCAL SPACE TIME FEATURES AND ADABOOST SVM

M. Shillin Bella[1], R.Bhanumathi[2], G. R. Suresh[3]

[1]*PG Student, Dept. of CSE, Apollo Priyadharshanam Institute of Technology, Chennai, Tamilnadu, India*
[2]*Assistant professor, Dept. of CSE, Apollo Priyadharshanam Institute of Technology, Chennai, Tamilnadu, India*
[3]*Department of ECE, Easwari Engineering College. Chennai, Tamilnadu, India*

Abstract

Human action recognition has a wide range of promising applications like video surveillance, intelligent interface, and video retrieval. The objective of this project is to recognize and annotate different human activities present in digital videos taken from both constrained and unconstrained environment. This work extended the use of the techniques existing in object recognition in 2D images to video sequences by determining the Spatio-temporal Interest points using the extension of Harris operator in the time dimension. Features descriptors are computed on the cuboids around these interest points and further they are clustered and bag of features is built .SVM is used to classify the different classes of action present in the video. The recognition rate is further improved by using Adaboost SVM wherein number of weak classifier is weighted to form a strong classifier. The result shows that the proposed method using adaboost SVM classifier, the mean accuracy rate of recognition of KTH dataset is 89.13%.

Keywords: *Space time Interest Point, Bag of Words, Support Vector machine, Multiple Instance Learning, AdaBoost SVM*

----------***----------

1. INTRODUCTION

Human Activity Recognition is an active research area in computer vision with wide range of applications in video surveillance, motion analysis, virtual reality interfaces, robot navigation and recognition, video indexing, browsing, HCI, choreography, sports video analysis etc. However, it remains a challenging problem because of factors like background cluttering, camera motion, occlusion and other geometric and photometric variances. The major steps involved in are analyzing and extracting the characteristic features of various human actions and classifying them. Many previous works can be roughly categorized into model-based methods and appearance-based approaches. Model-based methods [1], [2] usually rely on human body tracking or pose estimation in order to model the dynamics of individual body parts for action recognition. However, it is still a non-trivial task to accurately detect and track the body parts in unrestricted scenarios. Appearance-based approaches mainly employ appearance features for action recognition. For example, global space-time shape templates from image sequences are used in [3]–[5] to describe an action. However, in these methods, highly detailed silhouettes need to be extracted, which may be very difficult in a realistic video. Recently, approaches [6]–[8] based on local spatiotemporal interest points have shown much success in action recognition. Compared to the space-time shape and tracking based approaches, these methods do not require foreground segmentation or body parts tracking, so they are more robust to camera movement and low resolution. Many of the problems in categorization of human actions in video sequences are surprisingly well handled with bag of words [9] representation combined with machine learning techniques like Support Vector Machine (SVM). In this work, the local spatiotemporal interest points in a video are identified by collecting cuboid prototypes around the space time interest points, and then the Spatiotemporal Bag of features (BoF) was extracted by applying transformation on the detected cuboids. To classify various classes of actions present in videos SVM classifier is trained for different human actions. The accuracy of the classifier is further improved by Adaboost SVM wherein number of weak classifier is weighted to form a strong classifier. The videos for training and testing are taken from the standard dataset KTH [11] and Weizmann [10] dataset supplied for this purpose are well-explored in various studies.

2. SPACE TIME INTEREST POINT

In this section, the different steps used to extract the local space time interest point (STIP) are explained. The steps are as follows

2.1 Smoothing Out the Videos

Before detecting STIPs, the entire video sequence needs to be smoothen out by appropriate smoothening filter. The reason being are the lighting, rate of actions or the speed at which they are taking place may vary from one video to another. Let's consider a video sequence as a 3-D matrix with each frame spread out in x-y plane and the 3rd dimension corresponding to temporal domain. We convolve this matrix with a Gaussian kernel with independent spatial variance and temporal variance

.In order to detect the correct STIP select spatial and temporal variances of the gaussian kernel optimally. The spatial variance is fixed by the average variance of the background and the foreground frames. The background and the fore ground images are obtained by converting the images into frames followed by the simple background subtraction. But adapting these variances according to the sequence takes significant computation time.

2.2 Detection of Space-Time Interest Points

Space-Time interest points either correspond to corners in a particular frame or instances of sudden variation in time occurring in the video. It relies heavily on both the Harris measure [2] and a Gaussian scale-space representation. The algorithm basically tries to extend the principle of Harris Detector to detect STIPs in a video clip.The algorithm relies on a central principle that, at a corner, the image intensity will change largely in multiple directions. The useful information required for distinguishing between various actions lie across these STIPs. The aim is to obtain robust, stable and well-defined image features for object tracking and recognition. Corner detection was being used for many years in the past for detecting interest points in images. Harris affine detector is one of the Corner detection used in this work.

3. BUILDING BAG OF FEATURES

In this section, the different steps used to build the Bag of Feature (BoF) is explained. The steps are as follows

3.1 Cuboid Prototypes

Once STIPs of a video is available, it is necessary to extract useful information from them. The data around the interest points are grouped to form small chunks, called the cuboid prototypes. These cuboids contain key features of that particular frame. The size of each chunk $(\Delta_x, \Delta_y, \Delta_t)$ is related to the detection scales and the size of the feature vector used for representing an interest point. Then, each volume is further divided into grids of cuboids based on the similarities between the actions. The cuboid prototypes are preprocessed by applying transformations on them to make them invariant to difference in lighting, appearance, motion etc. yet retaining their discriminative power. These might include normalization of pixel values, brightness gradient, windowed optical flow etc. Once the transformations have been performed, a feature vector is generated by stretching out the pixel values of the cuboid into a vector sensitive to perturbations or histogramming the values.

3.2 HoG Descriptor

In this paper, for each cuboid a histogram of oriented gradient (HoG) is computed. The idea behind the Histogram of Oriented Gradient descriptors is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. This involves Gradient computation and Orientation binning. For the gradient computation the cuboids are filtered with the kernels [-1, 0, 1] oriented along 3 different dimensions. In the second phase, each pixel in the cell casts a weighted vote for an orientation-based histogram channel based on the values found in the gradient computation. Specifying the orientation of a vector in 3-D requires 2 angles as opposed to 1 angle in a 2-D case. So, we will be quantizing the orientations of the "projection" of the gradient vector onto 2 orthogonal planes (x-y and y-t) .The orientation of the gradient is quantized into finite number of bins in the range of 0o-180o.The magnitude of the gradient at a pixel is used as the weight of the vote that particular pixel casts. In the similar fashion it is extended to get feature descriptors for space-time volumes.

3.3 Bag of Features

From the HOG for each cuboid a Spatio-temporal bag of features (BoF) is build this requires construction of visual vocabulary. For this purpose, several such features are clustered using K-means clustering algorithm to find types of features in human behavior. Each behavior might have many such features and each feature might be present in many behaviors. But, the distribution of these features varies significantly for behaviors belonging to different classes. The BoF representation assigns each feature to the nearest cluster and computes the histogram of visual word occurrences over space-time volume to obtain distribution of the features in a video.

4. MIL FOR ACTION RECOGNITION

In this paper, we employ the Support Vector Machine (SVM) classifiers to classify various classes of videos. Specifically for each class, an independent SVM classifier is trained using the generated visual Bag of Word feature. The SVM uses Radial Basis function (RBF) as component classifier SVM represents the training sample as points in space, mapped so that the data of the different classes are divided by a clear gap that is as wide as possible. The test data is then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. SVM essentially tries to find a hyper plane in higher dimension that maximizes the separation between examples of various classes. Adaboost is the adaptive boosting algorithm and is a machine learning algorithm. The goal of boosting is to improve the accuracy of any given learning algorithm [14]. This classifier aims to employ RBFSVM as component classifier in Adaboost. Problems have been encountered when applying a single σ to all RBFSVM component classifiers. Generally, maintaining large value of σ leads to weak RBFSVM component classifier and its classification accuracy is often less than 50% which cannot meet the requirement on a component classifier given. On the other hand, a smaller σ often makes the RBFSVM component classifier stronger and boosting them may become inefficient because the errors of these component classifiers are highly correlated. Furthermore, too small a value of σ can even make RBFSVM over fit the training samples. Hence, finding a

suitable σ for these SVM component classifiers in Adaboost becomes a problem. By using model selection techniques such as k-fold or leave one – out, cross-validation, a single best σ may be found for these component classifiers. Initially, a large value is set to σ, corresponding to a RBFSVM classifier with very weak learning ability. Then, RBFSVM with this σ value is trained as many cycles as possible as long as more than half accuracy can be obtained. Otherwise, this σ value is decreased slightly to increase the learning capability of RBFSVM, to help it achieve more than half accuracy. By decreasing the σ value slightly prevents the new RBFSVM from being too strong for the current weighted training samples, and thus moderately accurate RBFSVM component classifiers are obtained.

5. RESULTS AND DISCUSSION

The entire work was implemented in MATLAB. The input videos for the experiment are taken from the standard KTH and Weizmann Dataset. Firstly, the videos from the dataset are read as 3-D matrices and are converted to gray scale. Smoothing action is performed on these matrices with various scales of spatial and temporal variances. The spatial variance σ and the temporal variance τ are fixed empirically σ = 2 and τ = 2.5 respectively. The interest points are extracted from the smoothened image is with the same spatial and temporal scale parameters σ and τ. Then cuboid prototypes were extracted around the these STIPs. The size of the cuboid is empirically fixed as 9×9×5 and 1000 interest points are extracted from each video. Then using 2-D HOG descriptor [12] feature descriptors for each volume are extracted. The orientation (unsigned) of the gradient is quantized into 9 bins in the range of 0o-180o .These descriptors are clustered using fast k-means algorithm [13] to obtain feature vector for each video. The extracted features are used to train "LibSVM" [14] to learn the features of videos belonging to different classes and labeling is done in this step. For each class 100 frames having different styles of the same action performed by different person are taken for training. The radial basis kernel is used for training the SVM. Then the adaptive boosting Adaboost SVM is used to increase the accuracy of the recognition is [15] used. The boosted or strong classifier which we used is a combination of weights and weak classifiers. We have selected arbitrarily 5 number of strong classifiers depend on Adaboost-SVM training procedure. After, these strong classifiers are organized into cascade-classifier structure. On every iteration the weights of each incorrectly classified example are increased, and the weights of each correctly classified example are decreased, so the new classifier focuses on the examples which have so far eluded correct classification. The trained SVM is tested with a video samples taken from videos in the standard dataset KTH and Weizmann. The performance is reported in Table I. It can be seen that performance using our proposed approach exceeds other methods on the KTH Dataset.

Table 1 Comparison of Different Methods About Mean Accuracy on KTH Dataset

Methods	Mean Accuracy(%)	Feature
Niebles and Li [18]	81.50	Spatio-temporal interest points
Saad and Mubarak [19]	87.90	Optical Flow
Our Method	89.13	Spatio-temporal interest points

The confusion matrix obtained for four different actions namely "Run" "Walk", "Jump" and "Side Run" using SVM classifier is shown in figure 1 (a) .The fig 1(a), shows that the recognition rate for the walk action is 50%. In order to increase the recognition rate Adaboost SVM classifier is used, where in number of weak classifier is weighted to form a strong classifier. The confusion matrix obtained for four different actions namely "Run" "Walk", "Jump" and "Side Run" using Adaboost SVM classifier is shown in figure 2. In the fig 1(b), it shows that the recognition rate for the walk action increased to 88% compared to the recognition rate using Support vector machine and the recognition rate of run and side run actions reached to 98 %.

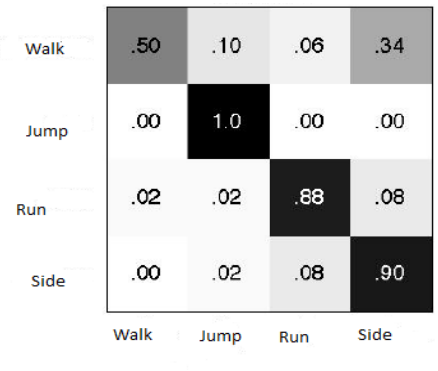

Fig. 1 (a) Confusion matrix obtained for four different

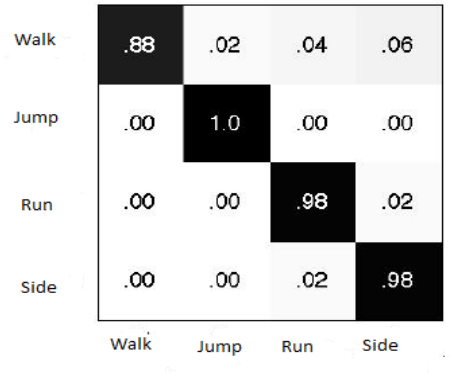

Fig. 1(b) Confusion matrix obtained for four different

action using SVM classifier

action using AdaBoost SVM classifier

6. CONCLUSIONS

The project uses space time interest point approach for action recognition. Each videos centered on the human figure was described using simple local space time features. The local space time features are clustered to form the BoF for each action. Based on the features an SVM classifier was learned to discriminate each action, and then AdaBoost-SVM classifier was employed to select the most discriminative features to form a strong classifier to discriminate closely related actions. Experimental results illustrated the effectiveness and efficiency of the proposed method for video action recognition and annotation.

REFERENCES

[1] C. Fanti, L. Zelnik-manor, and P. Perona, "Hybrid models for human motion recognition," in *Proc. IEEE CVPR*, pp. 1166–1173 Jun. 2005

[2] A. Yilmaz, "Recognizing human actions in videos acquired by uncalibrated moving cameras," in *Proc. IEEE ICCV*, pp. 150–157 Oct. 2005.

[3] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.

[4] A. Yilmaz and M. Shah, "Actions sketch: A novel action representation," in *Proc. IEEE CVPR*, pp. 984–989, Jun. 2005.

[5] X. Wu, Y. Jia, and W. Liang, "Incremental discriminant-analysis of canonical correlations for action recognition," *Pattern Recognit.*, vol. 43, no. 12, pp. 4190–4197, Dec. 2010.

[6] P. Doll´ar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. IEEE Int. Workshop PETS*, pp. 65–72 Jun. 2005.

[7] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. ICPR*, pp. 32–36, Aug. 2004.

[8] J.C. Niebles, H. Wang, and L. Fei-fei, "Unsupervised learning of human action categories using spatial-temporal words," *Int. J. Comput. Vision*, vol. 79, no. 3, pp. 299–318, Sep. 2008

[9] Ivan Laptev, MarcinMarszałek, Cordelia Schmid, Benjamin Rozenfeld. Learning realistic human actions from movies.IEEE Conference on Computer Vision & Pattern Recognition, 2008

[10] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Proc. ICCV*, 2005, pp. 1395–1402

[11] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. ICPR*, 2004, pp. 32–36

[12] I. Laptev. On space-time interest points. IJCV, 64(2/3):107–123, 2005.

[13] Alexander Klaser, MarcinMarszałek, CordeliaSchmid. A Spatio-Temporal Descriptor Based on 3D-Gradients.In BMVC, 2008

[14] O. Ludwig, D. Delgado, V. Goncalves, and U. Nunes, 'Trainable Classifier-Fusion Schemes: An Application To Pedestrian Detection,' In: 12th International IEEE Conference On Intelligent Transportation Systems, 2009.

[15] Mark Everingham's code on k-means implementation in C,2003.

[16] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.

[17] Xuchun Li_, Lei Wang, Eric Sung, "*AdaBoost with SVM-based component classifiers",* Engineering Applications of Artificial Intelligence 21 (2008) 785–795

[18] J. Niebles and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," in *Proc. BMVC*, 2006, pp 299–318

[19] S.Ali and M. Shah, "Human action recognition in videos using kinematic features and multiple instance learning," *IEEE Trans. Patt Anal. Mach. Intell.*, vol. 32, no. 2, pp. 288–303, Feb. 2010