Human Activity Recognition in Video Based on Histogram of Oriented Gradients and K-Nearest Neighbor

Khet Khet Khaing Oo
Faculty of Computer Systems and Technologies, University of Computer Studies, Myitkyina
khetkhetkhaingoo.edu@gmail.com

Yan Naung Soe
Faculty of Computer Systems and Technologies, University of Computer Studies, Myitkyina
ynaungsoe2007@gmail.com

Abstract

The human activity recognition system is automatically identified as human activity from the input video stream. It is a vital task in computer perception because it has many application areas such as healthcare, security, entertainment, and tactical scenarios. The system provides a way to automatically recognize human activity in the input video stream by distinguishing the Oriented Gradient Features (HOG) and K Nearest Neighbor (KNN) types. Functional features can be extracted from the input video frames with the HOG feature and can be organized to form an activity pattern. The empirical results and its accuracy indicate that the proposed system applies to recognition of human activity in real life. This system has been completed using the MATLAB programming software and the evaluation results for the system quality are measured by the confusion metric that is precision, recall and fitness measures.

Keywords: Histogram of Oriented Gradient features (HOG), K-Nearest Neighbor (KNN), and Leave One out Cross Validation (LOOCV).

1. Introduction

Today, people record their daily activities using digital cameras and this enriches the video resources on the Internet; and also raises the issue of how to categorize existing video resources and how do they classify new input videos according to their cast. The automation of actions that humans perform when communicating with one another may be of particular interest to the communication and may be an automatic expression of choice. [1]. Recognizing human acts from a distance is a challenge in the eyes of computers. It is automatically monitored, video analysis from Hawaii; a significant benefit for many applications, such as sports video comments and search. Most of these features can be safely extracted from advanced graphic programs.[2] .The system aims to provide a reliable solution for recognizing basic human actions including basketball, biking, billiards , clean and jerk, golf swing, playing guitar, playing piano, playing violin, punch, skiing and walking. In this paper Histogram of Oriented Gradients (HOG) features are used to represent human. HOG was originally proposed to be discovered by humans. HOG has successfully applied for the certification of its performance and material recognition due to its fitness. To create an action descriptor, the HOG function is extracted from the timeline of the overlay activity frame.

2. Related Work

Human activity is very complex and very different in nature because a movement can act in different ways for different reasons, depending on different circumstances. The majority of existing frameworks for action recognition consist of three main steps: feature extraction, feature vector construction for a video based on the extracted features, and finally classification of the video using the representation. In the first step, a set of features are extracted from a given video. HOG has been able to successfully exploit the problem of performance recognition due to its robustness and object recognition [2]. Assigning a video to a similar class of people by classifying a video that interrogates the learned models [8].

For this purpose, the proposed activity recognition system initiates HOG activities over a period of time to build the activity model. Then the HOG feature vector are used to classify the human activity using K-Nearest-Neighbor (K-NN) classifier. Stable video is provided by the actors' tracks. The purpose of our pre-implementation phase is to obtain a route-centric action plan. This step is very important. Because of the low resolution of small frameworks in low-resolution video frames, the loss of body parts or large backgrounds is limited. The input video is converted into frames as in Figure 1. Each frame is eroded with morphological structuring element. The eroded frame is subtracted from the actual input frame to get the boundary extracted frame Figure 2. Black and white images with only two colors. They are used to obtain a representation or description of an object or region (boundaries, bones, and body structures) in a pre- or post- modification (filtering, slapping, and pruning).

Figure 1. Input Image

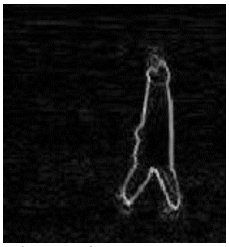

Figure 2. Boundary Extracted Image

2.1. Histogram of Oriented Gradients (HOG) Feature

Oriented Gradients' Histogram is one of the most popular photo descriptions used by humans to find images. The HOG algorithm is presented by Navneet etal. Navneet's work has four types of HOG: rectangular HOG (R-HOG), circular HOG (C-HOG), bar HOG and center-surround HOG. The HOG is a robust feature descriptor for the shape of the object. This feature is extracted from the image based on two factors. Number of bins (B) for overlapping windows (N x N) and gradient angles in the image. The first image is divided into windows that overlap N x N. The gradients of intensities for each window's pixels are computed by using a horizontal kernel [-1, 0, 1] and a vertical kernel [-1, 0, 1]-1. Next, the angles and magnitude are computed for each pixel in the window. Subsequently, the angles are divided into B groups based on the number of bins. The total amount for each group is obtained. Next, each of these operations is performed. Finally, when all windows are completed, N x N x B numbers become normal. These numbers are the HOG feature for the image [3]. The HOG algorithm consists of 3 parts: Gradient computation, Orientation Binning and Block normalization. The last 2 parts of the algorithm consist of variable that is defined by the user. This system defines the cell in the Orientation Binning to have 8x8 pixels and the block in the Block normalization with 2x2 cells. The following are detailed explanation on how HOG is calculated [7]:

- Gradient Computation
- Orientation Binning
- Block Normalization

2.2. K-Nearest Neighbor (KNN) Classifier

The KNN has three points. The first point sets the number of voters for K. The second distance metric is defined as Euclidean, city block (completely different), cosine matrix, etc. The third point specifies the rule for selecting the approximate class for the test sample, defined randomly by the nearest neighbor, etc. The KNN classifier is calculated using the distances between Activity recognition is formulated as a classification problem [4]. The KNN is a machine learning algorithm and the simplest method used for classification, clustering, and regression. The testing is divided by a majority vote of its K nearest neighbors. Given a test sample, the distance between the test sample and all the training samples in the training set is calculated based on some distance metric.

The testing video and each training sample, as provided by the following Equation:

$$d(x, m_j) = \arg\ min_j \{d\ (x, m_j)\} \qquad (1)$$

Where d is a distance metric, x is a testing sample, m is training samples, j = [1, 2… N], and N is a number of training samples. The distance d is arguably the minimum distance (nearest neighbor) among distances between x and each training sample. In all KNN experiments, the Leave-One-Video-Out (LOVO) cross validation technique is employed. Thus, all videos in the dataset are used for training except one video that is used for testing [6]. A K-Nearest Neighbor (K-NN) rule is one of the simplest and the most important methods in pattern recognition.

3. System Design and Implementation

The system flow diagram of the human action recognition in video is illustrated in Figure 3. Firstly, the system accepts video frame and subtract each frame with static background. The result image will contain motion blobs and the system have to remove noise to extract the accurately. First, the image converted into binary image and dilation and erosion operation is done to remove noise. Then, motion blob from the frame image is extracted and it is ready to extract gradient features. The motion blob image is resized into 128 x 128 pixels image. Then, the image is divided into 4 x 4 pixels size cells. Then, the system regroups the cells 2x 2 cells into block. After then, the system computes horizontal gradient, vertical gradient, gradient strength, and orientation of gradient using sobel 1-d operator. Finally, feature vector is extracted from each block by using 9-bin histogram. Feature vectors of all blocks are concatenated and the whole feature vector must be classified by the K-Nearest Neighbor classifier. The human action recognition system consists of two modes: training mode and testing mode. The training model is the first step toward recognition of human movement. The test mode is the second step in the human rights recognition system.

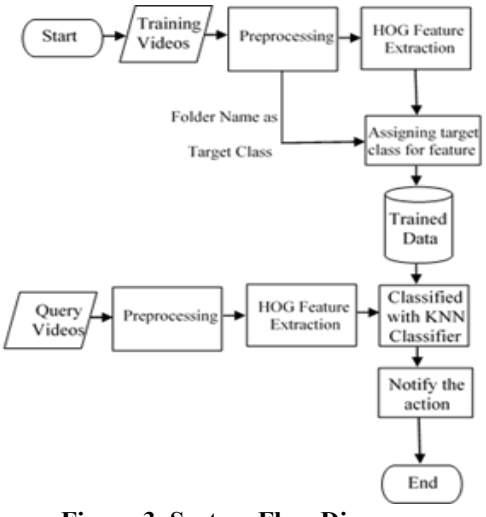

Figure 3. System Flow Diagram

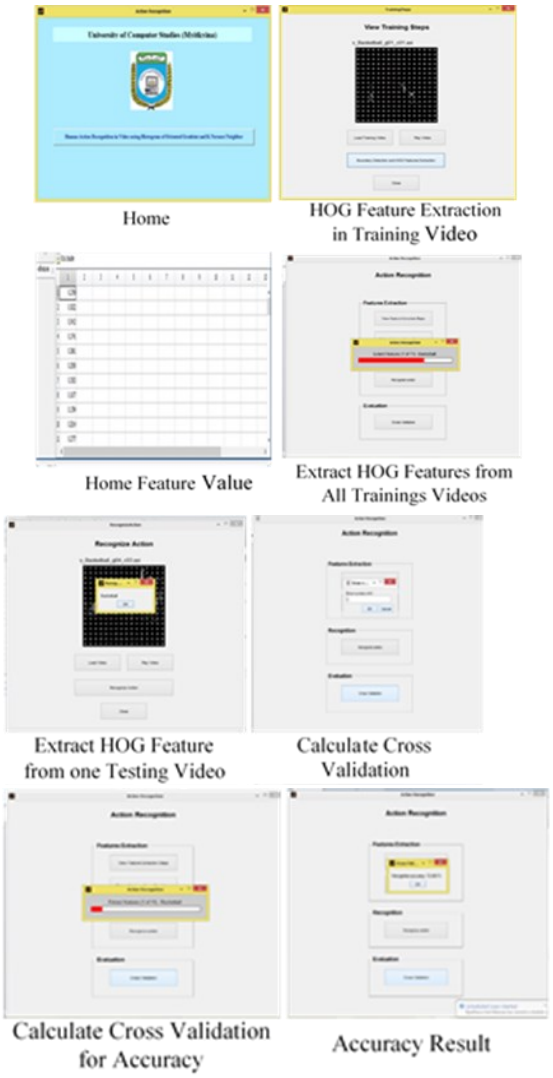

Figure 4 Graphical User Interface of Proposed System

In Figure 4, there are eight main utilities in this system. Home page of the proposed system is one button. When user clicks this button the HOG Feature Extraction in Training Video are four buttons. The first button is used for boundary detection and HOG feature extraction in one training video as shown in HOG Feature Extraction in Training Video. HOG Feature Value shows HOG feature value. The second button is used for extract features from all trainings videos as shown in Extract HOG Features from All Trainings Videos. The third button is used for extract HOG feature and human action recognition in testing video as shown in Extract HOG Feature from one Testing Video. The fourth button is used to calculate cross validation for accuracy as shown in Calculate Cross Validation and Calculate Cross Validation for Accuracy. Accuracy Result shows accuracy result for action recognition.

4. Experimental Results

In UCF dataset, the main focus is to provide the computer vision community with an action recognition dataset includes of realistic videos which are taken from YouTube. It is very challenging due to large variations in camera movement, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions [5]. The dataset includes 11 actions: basketball, biking, billiards, playing violin, playing guitar, playing piano, clean and jerk, and golf swing. Figure 5 shows a sample frame of all eleven actions. Datasets are available to the public with notes and a limited view of the population.

Figure 5. UCF Dataset

To measure the quality of human action recognition, Leave One out Cross Validation is used. The visual quality of human action recognition is evaluated by the Leave One out Cross Validation method.

❖ Leave One Out Cross Validation

Leave One out Cross Validation is the only validation study in the original sample, and the rest of the findings were used as training data. The LOOCV strategy is adopted to evaluate the human action recognition problem. In LOOCV, each action instance is chosen as the test sample, and the rest action instances are chosen as the test set.

❖ K-fold Cross Validation

Cross-validation is a re-sampling plan used to appraise machine learning models on a restricted information sample. There is a single parameter called k that refers to the number of groups to be separated into a given sample of data. As such, the procedure is often called k-yard cross-validation. When you select a specific value for k, k = 5 is used to refer to the 5-fold cross-validation model. The goal of cross-validation is to test the model's ability to appropriate new data that was not used in forecasting it, in order to flag problems like over fitting or provide an insight into how generally the bias and choice of model and independent datasets. Because it's easy to understand, it's generally a popular method because it's generally less biased or less optimistic than the others methods, such as a simple train/test split. The general procedure is as follows:

- Change datasets randomly
- Divide datasets into k groups
- For each unusual group:
  - Take a group or hold a bunch of data
  - Take the rest of the team as a set of training data
  - Set an example in the training package and evaluate the inspection plan
  - Retain the evaluation score and discard the model
- Summarize the skills of the model using the example of standardized assessment scores

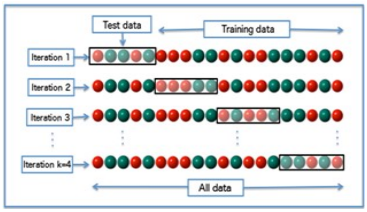

Figure 6.Cross Validation

❖ Confusion Matrix

A complexity table is a table used to describe the performance of specific models on a set of test data for known complexity values. This system contains information about predictions made by categorization. The performance of such systems is usually evaluated using data in the matrix. The process for calculating a confusion matrix as follows:

- First, complexity requires a set of tests that are related to a test or to the expected value.
- It then makes predictions for each class in the system test set.
- Lastly, the user can count the number of correct and incorrect predictions for each class from each of the estimated numbers and estimates.

These numbers are arranged into a table or matrix as follows:

- Expected down the side: Each row of the matrix is associated with a row.
- Predicted across the top: Each matrix corresponds to a predefined row of columns.

The right and wrong numbers are then added to the table. The correct estimate for a class goes to the expected column for that class and the estimated column for that class. Similarly, the total number of incorrect estimates for a row goes to the expected column for that value and the estimated column for that value. The System is used 220 videos for training and 110 videos for testing in UCF Dataset with eleven actions. The following Table 4.1 shows by the value of k=3. In recognition of eleven actions, 11 videos are tested for each action. When testing for biking action, 8 videos are correct and the remaining 2 videos are wrong with skiing videos. The reason is the feature vectors of biking action are much similar with feature vectors of skiing action. The following Figure 7 and 8 shows by the Feature Vectors of Training Skiing Video and Testing Biking Video.

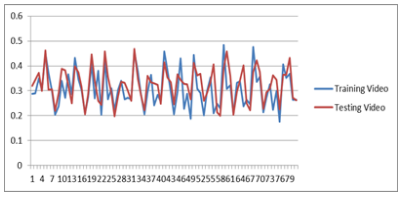

Figure 7. Feature Vectors of Training Skiing Video and Testing Biking Video 1

In Figure 7, the training skiing video feature vectors and the testing biking video1 feature vectors are nearly the same distance at 0.54.

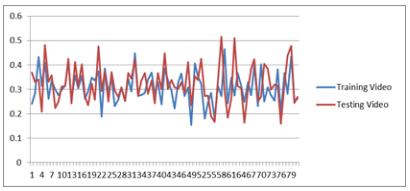

Figure 8. Feature Vectors of Training Skiing Video and Testing Biking Video 2

In Figure 8, the training skiing video feature vectors and the testing biking video2 feature vectors are nearly the same distance at 0.67.

Table 4.1 Results of Confusion Matrix for UCF Dataset Using HOG and KNN Classifier at K=3

Human Actions	Basketball	Biking	Billiards	Clean And	Golf Swing	Playing	Playing Piano	Playing	Punch	Skiing	Walking With Dog	Recall
Basketball	10											1
Biking		8								2		0.8
Billiards			10									1
Clean and Jerk				9					1			0.9
Golf Swing			1		9							0.9
Playing Guitar						8		2				0.8
Playing Piano							10					1
Playing Violin						1		7	2			0.7
Punch				3					7			0.7
Skiing										10		1
Walking With Dog											10	1
Precision	1	1	0.91	0.75	1	0.80	1	0.78	0.7	0.83	1	0.89

The following Table 4.2 shows by the value of k=11.The accuracy of all videos in UCF dataset with eleven action is 68%.\

Table 4.2 Results of Confusion Matrix for UCF Dataset Using HOG and KNN Classifier at K=11

Human Actions	Basketball	Biking	Billiards	Clean And	Golf Swing	Playing	Playing	Playing	Punch	Skiing	Walking	Recall
Basketball	9				1							0.9
Biking		6		1					1	2		0.6
Billiards			5	5								0.5
Clean And Jerk				9				1				0.9
Golf Swing			1		6		1	1	1			0.6
Playing Guitar						6		3	1			0.6
Playing Piano							10					1
Playing Violin						1		8			1	0.8
Punch				7				1	2			0.2
Skiing									4	6		0.6
Walking With Dog								2			8	0.8
Precision	1	1	1	0.39	0.86	0.86	0.91	0.5	0.29	0.75	0.89	0.68

The following Table 4.3 shows by the values of k=3,k=5,k=7,k=9 and k=11.The accuracy of all videos in UCF dataset with eleven actions is 70%,86%,78% ,72 and 68%. Like the Recall columns computed by Table 4.2 (k = 11 @ 68%), Table 4.3 shows the results for k = 3 @ 70%, k=5@86%,k=7@78%,k=9@72%.

Table 4.3 Results of Confusion Matrix for UCF Dataset Using HOG and KNN Classifier

Human Actions	The Value of K, the accuracy of all video in UCF dataset with eleven action percentage (%)					
	K=3,@70%	K=5,@86%	K=7,@78%	K=9,@72%	K=11,@68%	Recall
Basketball	0.67	0.9	0.8	0.8	0.9	
Biking	0.59	0.8	0.8	0.8	0.6	
Billiard	0.73	0.8	0.7	0.6	0.5	
Clean /Jerk	0.64	0.9	0.9	0.9	0.9	
Golf Swing	0.61	0.7	0.6	0.5	0.6	
Playing Guitar	0.63	0.9	0.6	0.6	0.6	
Playing Piano	0.57	1	1	1	1	
Playing Violin	0.9	0.9	0.9	0.8	0.8	
Punch	0.78	0.6	0.5	0.3	0.2	
Skiing	0.71	1	0.8	0.6	0.6	
Walking With Dog	0.83	1	1	1	0.8	
Precision	0.70	0.86	0.78	0.72	0.68	

When the value of k increases, the accuracy of all videos in UCF dataset with eleven action slightly decrease (i.e., basketball, biking, billiards, clean and jerk, golf swing, playing guitar, playing piano, playing violin, punch, skiing, and walking with dog). The UCF dataset contains 1478 videos. The System is used 1185 videos for training and 293 videos for testing. The following table 4.4 shows the accuracy of all videos in UCF dataset with eleven actions.

The following table 4.4 shows by the value of k=3.The accuracy of all videos in UCF dataset with eleven action is 70%.

5. Conclusion

This system presents a new method using KNN for action recognition and HOG for feature extraction. The result is hopeful enough to discover real-life applications of action recognition in fields like surveillance systems, healthcare systems and entertainment environments system. Human action–recognition approaches involve two important blocks. The feature extraction is first of the important blocks, which extract a set of key parameters that best describe the particular set of human action so that the parameters can be used to distinguish among other actions. It is the second major blockade that plays a vital role in the recognition of human rights discrimination in the extractive system. The empirical results and its accuracy indicate that the proposed system recognizes human activity.

Table 4.4 Results of Confusion Matrix for UCF Dataset with all Videos Using HOG and KNN Classifier

	Predicted											
Human Actions	Basketball	Biking	Billiards	Clean And Jerk	Golf Swing	Playing Guitar	Playing Piano	Playing Violin	Punch	Skiing	Walking With Dog	Recall
Basketball	18	1		1	1	2	1			3		0.67
Biking		17				1	2	5		2	3	0.59
Billiards	1		22			1	5		1			0.73
Clean and Jerk	4	2	1	14			1					0.64
Golf Swing	2	3	1		17	2			2	1		0.61
Playing Guitar		3	1			20		2	1	3	2	0.63
Playing Piano	1				2	1	12	3		2		0.57
Playing Violin		2						18				0.9
Punch	1	2	3				1		25			0.78
Skiing	1	4		2			1			20		0.71
Walking With Dog									2	2	20	0.83
Precision	0.64	0.5	0.79	0.82	0.85	0.74	0.55	0.63	0.81	0.61	0.8	0.70

References

[1]. Jagdish L.Raheja, A.Singhal and Sadab, “Android based Portable Hand Sign Recognition System”, Machine Vision Lab, Ankit Chaudhary Department of Computer Science Truman State University USA, March, 2015.

[2]. Chia-Chih Chen and J.K.Aggarwal, “Recognizing Human Action from a Far Field of View”, Computer & Vision Research Center at Department of ECE, University of Texas at Austin ,January, 2010.

[3]. Jagdish L.Raheja, A.Singhal and Sadab, “Android based Portable Hand Sign Recognition System”, Machine Vision Lab, Ankit Chaudhary Department of Computer Science Truman State University USA, March, 2015.

[4]. Zhe Zhang, “Vision-based Human Action Recognition: A Sparse Representation Perspective”, Electrical Engineering Theses and Dissertations, April 2012.

[5]. https://www.crcv.ucf.edu/data/UCF50.php

[6]. H.S. Hippert, C.E. Perdreira and R.C. Souza, “Neural Network for Short Term Load Forecasting: A Review and Evaluation,” IEEE Transactions on Power Systems vol. 16, pp. 44-55, February 2001.

[7]. S. Rahman and R. Bhatnagar, “An Expert System Based Algorithm for Short Term Load Forecast,” IEEE Transactions on Power Systems vol. 3, pp. 392-399, May 1988.

[8]. Khurram Soomro and Amir R. Zamir, “Action Recognition in Realistic Sports Video”, Center for Research in Computer Vision, University of Central Florida, Orlando USA and Gates Computer Science, Stanford University, Stanford, USA, 2014.