Machine Learning for Continuous Human Action Recognition

Tian Tang
Department of Electrical Engineering, Stanford University
tangtian@stanford.edu

***Abstract*—In this term project, we consider the problem of automatic recognition of continuous human activity. Our source data are short videos with RGB and depth information of seven predefined categories of human action as well as long videos that contain a series of continuous actions. By extracting frame-level features that represent each action and are invariant to small environmental noise, we train the model with part of our data and test it with the remaining data and then compare the precision of SVM and Order Representation model with different choice of features.**

I. Introduction

This project falls into the domain of computer vision and artificial intelligence, where continuous human motion analysis and recognition is one of the most attractive topics. There are tremendous amount of work on this field during the past decade based on static images and 2D videos. With the development of depth sensors, we can make better cognition algorithms utilizing depth maps. To achieve this, previous work proposed many ideas on feature choice and extraction, such as sparse interest points, dense trajectory and skeleton information, etc.

However, challenges lie in several aspects. The difficulties in video and image processing still exist. Variations in the environments where the actions take place are an important source of uncertainties and failures in recognition tasks. Feature selection affects both accuracy and computational expenses of the algorithm, which makes it crucial to the project.

The exact problem we address in this paper can be described as follows.

To start with, seven basic action categories, drinking, eating, using laptop, reading cellphone, making phone call, reading book and using remote, are defined and videotaped. These source videos are provided in RGB-D by Gang Yu et.al online [1]. By modeling each motion, we aim to achieve the following tasks:

(1) Same-environment action recognition:

Training videos and test videos are taken in same environment. Each action is performed by different people in their own ways which suggests large intra-class variations. The algorithm should be robust to recognize each action precisely.

(2) Cross-environment action recognition:

Training videos and test videos are taken in different environment. We should be able to eliminate the interference of environmental factors and recognize the action correctly.

(3) Continuous action recognition:

Test videos contain a series of continuous actions performed by one person, where we don't have prior knowledge about the commencement and the termination of each action. We are expected to segment a continuous human activity into separate basic actions as defined and correctly identify each one of them.

After preprocessing the original video episodes, we choose and extract frame-level features that can effectively represent each action in consideration of computational complexity as well as model precision.

In this paper, we adopt the frame-level skeleton method proposed in [2] that extracts four types of features from RGB skeleton joints as well as depth maps. The feature selection method will be further elaborated in part 3.

In part 4, I will explain the order representation model used. Test results and discussion will be given in part 5 and 6 correspondingly while conclusion and future research interests will be shown in part 7 and 8.

II. DATA AND IMPLEMENTATIONS

The first table below shows the data sets used in this paper. There are 37 performers who perform 7 different actions in two environment settings in total. Each person demonstrates each action several times with slight differences, which implies varieties. Note that S0, S1 and S2 are in the same background setting, and S3, S4 are in a different setting.

The second table shows the training and testing set(s) used for each task.

TABLE I. DATA SET INFORMATION

Dataset	Episodes	Performers	Action(s)	Time
Set 0 (S0)	14	NO. 0-2	1 (“Sit”)	N/A
Set 1 (S1)	112	NO. 0-7	7	2
Set 2 (S2)	112	NO. 8-16	7	2
Set 3 (S3)	112	NO. 17-24	7	2
Set 4 (S4)	36	NO. 25-36	Continuous	3

TABLE II. IMPLEMENTATION

Task Number	Training set(s)	Test set(s)
Task 1	S1	S2
Task 2	S1, S2	S3
Task 3	S0, S1 and S2	S4

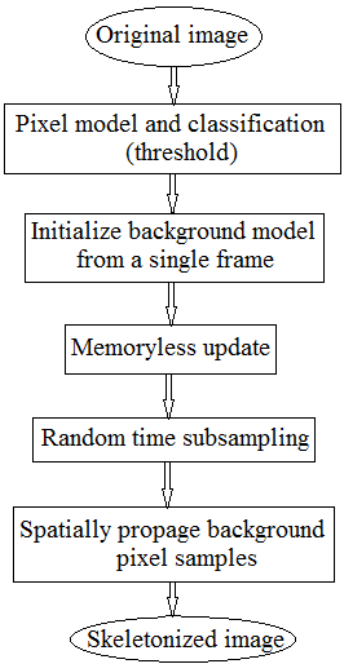
III. FEATURE SELECTION

In this paper, we adopt the frame-level skeleton joints methodology proposed in [2] that selects the special configuration of 20 skeleton joints and represent each action with four types of features derived by calculating the relative distance of the chosen joints. First, we need to preprocess the video and segment the foreground in each frame of the sequence and acquire its skeleton.

(1) Preprocessing and Skeletonization

Skeletonization has been widely used in human motion analysis to generate skeletonized image of human subject in foreground. Thus the accuracy of preprocessing and segmentation of the subject in each frame of the video sequence may affect the overall precision of the algorithm, which makes it crucial.

Traditional skeletonization methods contain the following steps: threshold segmentation which yields a binary image, morphological operations (dilation, erosion, connected component labeling etc.) which give the foreground of each frame. Here, we use the universal background subtraction algorithm proposed in literature [4] called ViBe. The overall idea of this algorithm can be described as below:



Picture I. Flowchart of preprocessing method

(2) Feature extraction

In this paper, we use expressions and characters in consistent with previous papers.

Suppose the training dataset has N_R videos: $R = \{(v_i, y_i), i = 1,2,\cdots N_R\}$, and $y_i \in \{0,1\}$ refers to the label of the video. $V = [I_1, I_2, \cdots I_T]$, where I_t, $t = 1,2,\cdots T$ refers to a frame of the RGB-D video sequence. For I_t, the skeleton is denoted as $S^t = \{s_1^t, s_2^t, \cdots, s_{N_s}^t\}$, where $s_i^t = (x_i^t, y_i^t, z_i^t)$ is the coordinate of joint on the t-th frame and $N_s = 20$ is the total number of joints.

Here, we have 4 types of features, three of which are extracted from human skeleton (RGB video) and one is extracted from the depth video to get the object shape and position.

(i) Pairwise joint distance:

$$\lambda^{(1)} = \left\| s_i^t - s_j^t \right\| \quad (1)$$

(ii) Spatial coordinate of a joint:

$$\lambda^{(2)} = x_i^t \text{ or } y_i^t \text{ or } z_i^t \quad (2)$$

(iii) Temporal variation of a joint location:

$$\lambda^{(3)} = \left\| s_i^t - s_i^{t-\tau} \right\| \quad (3)$$

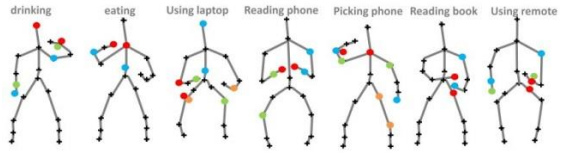
where τ is a time duration.

(iv) Object feature:

It is crucial to gain object information (shape and position) in this human-object interaction motion recognition problem. By using Local Occupancy Pattern (LOP) [6], we can obtain the potential object positions. Here, we give the extracted object feature directly.

$$\lambda^{(4)} = \|d(i) - d(j)\| = \left\| l_i - l_j \right\|, \quad 1 \le i, j, \le N_b, i \ne j, \quad (4)$$

where $N_b = N_x \times N_y \times N_z$, $d = [l_1, l_2, \cdots l_{N_b}]$, $l = \frac{1}{1+exp(-\beta\gamma)}$, β is a parameter, γ is the number of cloud points of each cell for each potential object position.



Picture II. Examples of extracted joint features

The major merit of this feature selection method is that it successfully transforms a large amount of RGB-D information into low-level joint features and their Euclidian distances. Moreover, this feature extraction method is comparatively robust and insusceptible to background noises. Moreover, this feature extraction method is comparatively robust and insusceptible to background noises, which is very useful in cross-environment action recognition tasks.

IV. MODELS

As we mentioned before in introduction section, there are large intra-class variations in human actions as same actions are performed differently in speed, style and environment. This makes automatic action recognition even more challenging.

To deal with this problem, we use order representation model based on our selected features. For each action, there are certain joints that play more important roles than other joints in defining the action itself. Take "making phone call" as an example. The hand-ear distance should be small for every demonstration no matter what the human subject's leg positions are. That is to say, the hand-ear joint pair should be of the smallest norm for action "making phone call".

Moreover, object features are also important in helping recognize an action. For example, even with the information of "the object is in human's hand", we still need more information about the object itself to decide whether the person is eating, drinking or using cell phone.

Given a video episode with a certain label of action type, we can get a series of video frame corresponding to time. Within each frame, we extract four types of features from the RGB-D video. Denote $\Lambda^{(f)}$ as the complete feature set of type f. Then we have $\left|\Lambda^{(1)}\right| = N_s \times (N_s - 1)/2$, $\left|\Lambda^{(2)}\right| = N_s \times 3$, $\left|\Lambda^{(3)}\right| = N_s \times N_\Delta$, $\left|\Lambda^{(4)}\right| = N_b \times (N_b - 1)/2$.

Here, we define size-n order indicator p as in [2] as:

$$p = (O_p, k) \quad (5)$$

where $O_p = [\lambda_{i_1}^{(f)}, \lambda_{i_2}^{(f)}, \cdots, \lambda_{i_n}^{(f)}]$ is a subset of $\Lambda^{(f)}$, k is the index of element of O_p with the minimum value λ.

The response of p on video frame I_t is defined as:

$$v_p(I_t) = \begin{cases} 1, & \lambda_{i_k}^{(f)} \le \lambda_{i_j}^{(f)} \text{ for all } \lambda_{i_j}^{(f)} \in O_p \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

Thus, the representation of video V with T frames on p can be written as:

$$V_p(V) = \sum_{t=1}^{T} v_p(I_t) \quad (7)$$

This value of $V_p(V)$ should be high for relevant videos and low for less relevant videos.

Given a size-2 ($n = 2$) order indicator p and a threshold θ_p, we define the classification function $F_{p,\theta_p}(V)$ as:

$$F_{p,\theta_p}(V) = 1(V_p(V) > \theta_p) \quad (8)$$

Then, to address the problem of different action demonstration speed, we need to add a normalization weight to each frame to get videos of same durations.

We define classification error as:

$$\epsilon_p = \frac{1}{N_R} \sum_{i=1}^{N_R} 1(F_{p,\theta_p}(V) \neq y_i) \quad (9)$$

where $y_i = \{0,1\}$ is the label of the video.

The optimal threshold value can be gained by minimizing classification error $\theta_p = \min_{\theta_p} \epsilon_p$. For simplification, write F_{p,θ_p} with optimal θ_p as F_p.

Based on the above classification error, we can sort all size-2 order indicators Γ_2 and remove the redundant ones to form a compact set by using threshold μ. If $\epsilon_p < \mu$ then we keep p and move on until we finish with all indicators of size-2 and get the updated size-2 order indicator set Γ_2'.

Similarly, we can update size-2 order indicators to size-L order indicators with smaller classification errors. Finally we get an order indicator pool $P^{(f)}$ as output.

After obtaining distinct pools for the four types of features, we need to combine them to get the final order indicator pool $P = P^{(1)} \cup P^{(2)} \cup P^{(3)} \cup P^{(4)}$. Upon doing so, weights should be determined for each $P^{(f)}$ for each action category.

Here, we use the boosting method in literature [7], called Weak Learn. The weak learners are trained to find a weak hypothesis while trying to maintain a distribution over the training set.

After applying Weak Learn algorithm to classification function $F_{p,\theta_p}(V)$ defined in equation (8), we get the revised classifier $g(V)$:

$$g(V) = 1(\sum_{m=1}^{M} \alpha_m 1(F_m(V) = 1) > \sum_{m=1}^{M} \alpha_m 1(F_m(V) = 0)) \quad (10)$$

where M is the number of weak learners, F_m is the m-th weak classifier and α_m is its weight.

For multi-class recognition problem we have here, the number of categories is 7, i.e. $C = 7$.

For each video V, the classifier gives 0 and 1 label on action category and the video belongs to the action category c^* with largest weighted sum (one-to-one mapping):

$$c^* = \arg\max_c \sum_{m=1}^{M} \alpha_m^c (1(F_m^c(V) = 1) - 1(F_m^c(V) = 0)) \quad (11)$$

Algorithm Order Indicator Pool

Input: Initial order indicator set Γ_2, indicator size L

Output: Order Indicator Pool $P^{(f)}$

1: $P^{(f)} \coloneqq \emptyset$
2: **for** $l \coloneqq 2 \to L$ **do**
3: $\Gamma_l' \coloneqq \{p_j | p_j \in \Gamma_l, \ \epsilon_{p_j} < \mu\}$
4: $P^{(f)} \coloneqq P^{(f)} \cup \Gamma_l'$
5: **if** $l < L$ **then**
6: $\Gamma_{l+1} \coloneqq \emptyset$
7: **for** $\lambda_{i_{l+1}}^{(f)} \in \Lambda^{(f)} - O_{p_j}$, where $p_j \in \Gamma_l'$ **do**
8: **for** $k := 1 \to l + 1$ **do**
9: $p^* \coloneqq \left(\left[O_{p_j}, \lambda_{i_{l+1}}^{(f)}\right], k\right)$
10: $\Gamma_{l+1} \coloneqq \Gamma_{l+1} \cup p^*$
11: **end for**
12: **end for**
13: **end if**
14: **end for**

V. Results

Apply the order representation model introduced in section 4 on our dataset, and compare the result with SVM (full feature sets). We get the following table with training and test errors for each model and task.

TABLE III. TABLE OF RESULTS

Task 1: Training set: S1 (112 samples); Test set: S2 (112 samples).		
Models	**Training error**	**Test error**
Pairwise joint distance features only	0.205	0.366
Spatial coordinate features only	0.348	0.446
Temporal variation features only	0.411	0.536
Object features only	0.383	0.473
Weak learn algorithm boosted Order Representation	0.098	0.232
SVM	0.089	0.268
Task 2: Training set: S1 and S2 (224 samples); Test set: S3 (112 samples).		
Order Representation	0.237	0.295
SVM	0.263	0.321
Task 3: Training set: S0, S1 and S2 (238 samples); Test set: S4 (36 samples).		
Order Representation	-	0.417
SVM	-	0.472

VI. DISCUSSION

The results gained using order representation model are consistent with our common senses. The overall precision of same-environment action recognition is the highest of all three tasks, followed by cross-environment action recognition and then continuous action recognition. Weak learn boosting algorithm gives better precision than SVM with same feature choice while there is still room to improve them both. This high bias problem can be fixed by using larger set of features or by using other feature choices. This also makes sense as the size of feature pool we use to identify each action is much smaller than the originally defined feature sets.

According to leave-one-out cross-validation results, pairwise joint distance feature contributes most to our model accuracy, while weak learn boosting contributes least.

VII. CONCLUSION

In this paper, we applied order representation model as well as SVM to same-environment, cross-environment and continuous human action recognition tasks. The results of the first two tasks are acceptable, and the comparatively low precision of continuous action recognition indicates the big challenges in computer vision where more effort can be paid.

VIII. FUTURE

We can further improve the algorithm to reach higher precision and shorter running time (for example, for the action drinking, we are able to tell whether the tester is drinking water or soda and we can make the recognition process real-time). Both training error and test error are high for task 2 and 3, we should try to use more features or try other features. Hidden Markov Model (HMM) also gives a promising future in dealing with the temporal relationships inherent in human actions.

REFERENCES

[1] Data source: http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/

[2] Gang Yu, Zicheng Liu, Junsong Yuan. Discriminative Orderlet Mining for Real-time Recognition of Human-Object Interaction, *ACCV* 2014

[3] Anjum Ali, J.K. Aggarwal. Segmentation and recognition of continuous human activity, *Detection and Recognition of Events in Video*, 2001.

[4] Olivier Barnich, Marc Van Droogenbroeck. ViBe: A Universal Background Subtraction Algorithm for Video Sequences, *Image Processing, IEEE Transactions*, 2010 (Volume 20, Issue 6).

[5] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kip- man, and A. Blake. Real-time human pose recognition in parts from single depth images, CVPR, 2011.

[6] J. Wang, Z. Liu, Y. Wu, J. Yuan. Mining Actionlet Ensemble for Action Recognition with Depth Cameras, *CVPR*, 2012.

[7] Robert ESchapire. A Brief Introduction to Boosting, *Sixteenth International Joint Conference on Artificial Intelligence*, 1999.

[8] J. Wang, Z. Liu, J. Chorowski, Z. Chen, Y. Wu. Robust 3D Action Recognition with Random Occupancy Patterns, *ECCV*, 2012