Human Action Prediction with Hierarchical Movemes

Tsung-Chuan Chen, Pei-Chun Chen, Stanford University

Abstract— **We propose a hierarchical movemes structure for the problem of human action prediction. The features of human actions are calculated by HOG descriptors and exemplar-SVM models. The finer-grained descriptions (movemes) of the human action are calculated from these features using dynamic time warping based segmentation. The dataset we use is collected from Youtube and consists of clips from 20 different TV shows. Different numbers of layers and different classifiers are used to predict the future action. We achieve the best prediction accuracy 46.7% using SVM on the proposed hierarchical moveme model with 3 layers.**

Keywords— **Human Action Prediction, Hierarchical Movemes, Finer Grained Actions, Support Vector Machines, Unsupervised Learning Methods, Dynamic Time Warping segmentation.**

I. INTRODUCTION

Human action recognition is one of the most classic problems in the area of computer vision and there are many useful related applications. For example, automatic human action recognition in a video scene of a surveillance system helps abnormal event detection, person counting in a dense crowd, person identification, gender classification, etc. Motion sensing device like Kinect enables users to control and interact with their console/computer without the need for a game controller, but through a natural user interface using gestures and spoken commands. In this project, we focus on human action prediction. How and when is the future motion going to happen? One possible application of this is to detect fall or other dangerous motions of elderly people. According to World Health Organization, the world population is rapidly ageing. Between 2000 and 2050, the proportion of the world's population over 60 years will double from about 11% to 22% [1]. Other possible applications include automated robots, which can now respond faster and more accurate to human actions according to the prediction.

Due to the promising applications of human action prediction, we aim to build systems and models that can estimate the states of human action with time order. In this way, after deciding the current state of the human action, we can predict how the action will proceed. In the future, successful models for human action prediction can be incorporated into light portable or mobile devices and make protecting elderly people and toddlers from falls, etc. an automatic and easy process without human intervention.

In this project, we focus on predicting future human actions given their current behaviors as shown in Fig. 1. Specifically, given the current image of the human object, we want to answer what actions he/she will execute in the near future, e.g. next few seconds.

Fig.1. The predicted human actions given their current behaviours

However, most of the existing datasets lack finer annotations of human actions. For example, a hug might consist of opening arms, body approach/contact, closing arms, etc. In the current dataset, all different states of hug will be annotated with action 'hug' without differentiation. Nevertheless, these fine-grained action states are helpful while predicting future action. Thus, we introduce the concept of hierarchical movemes, which decompose the human actions into multiple layers with different levels of granularities to make our model closer to how people interpret human actions in reality.

The learning problem with hierarchical movemes is more complicated since we have to consider the relation between movemes across different layers in addition to the relation between features and the action labels. We also have to predict a set of labels, i.e. labels for different layers, at the same time. Different classification methods are explored to solve the optimization problem with the proposed hierarchical model. The results show that using multiple layers of movemes achieves better prediction accuracy, which validates the presentation of hierarchical movemes and the claim that fine-grained action states help improve the prediction.

II. DATA SETS

The dataset in this study is obtained from the Computational Vision and Geometry Lab, Stanford. The dataset was collected from Youtube and consists of video clips from 20 different TV shows. There are five action categories in the dataset, which are "High-five", "Hug", "Handshake", "Kiss" and "None". The total number of videos is 200 with 50 videos in each of the action category except for "None". The frame numbers of the videos range from approximately 70 to 200. Additionally, the bounding boxes and the orientations of the five kinds of human activities are

TABLE I
DATASET

Video Source	Clips from 20 TV shows on YouTube
Total Number of Videos	200
Frames per video	70 – 200 frames
Frame Rate	20 Frames/s
Action classes	High-five Hug Handshake Kiss None
Orientation classes	Facing Left Facing Right Facing Toward the camera Back against the camera
Bounding Boxes	Human with actions

annotated. And all the information of the dataset are summarized in Table 1.

III. Previous Work

Recent research of human action prediction includes the work of recognizing detection of specific actions [2]-[4]. However, the proposed methods in these works require a relatively long sequence prior to the exact execution of the actions, such as the work in [2] and [3]. Specifically, they would need at least 3 seconds (or approximate 90 frames) before the actions really happens [2]. But this may not be a reasonable condition since that the time takes for one action to happen usually takes less than 1 to 2 seconds. If we not only aim to predict human action but require fast response to the action, we could only use shorter video clips or even single frame of the image in order to predict the action.

The other similar application of predicting human behaviors in the literature is to predict the trajectory of pedestrians for either flow control or for traffic control in dense-populated regions [5] such as subway or department stores. However in this study, we are more focused on predicting the real human action instead of only predicting the statistics of actions of a group of people.

Furthermore, previous works rely only on the annotated labels, which may not be accurate or informative enough as described in section I. The annotations in most of the dataset are only approximate to what the people in the frame are actually doing. As can be seen in Fig 2, the annotation of action is coarse and cannot accurately reflect the finer details of the action. Although all the actions in the frames are labeled as hug, these actions are actually more diverse and complicated. For example, in order to hug, we start from raising and opening arms, etc. If these actions can be described by finer-grained representations as shown on the bottom of Fig. 2, then the predictions can be made based on these representations. However, to the best of our knowledge, none of the datasets have this kind of finer-grained annotations of actions. Thus, in this project, we make a first attempt to automatically discover these finer-grained actions based on unsupervised methods.

There are also works of predicting the actions without a learning scheme. And one of the most popular methods is siftflow [6]. The algorithm using siftflow would predict human actions by comparing the similarity between the features of query and candidate images calculated by the proposed siftflow descriptors. However, the performance of siftflow may not be competitive enough, since the information provided by the annotations of the dataset is not utilized. This kind of method is better suited for applications like image retrieval. In order to prove our viewpoints, we still try using siftflow to predict human action and the result is shown in Section VII.

IV. Contribution

In this work, we proposed a method to automatically find the finer grained descriptions of the actions, which we call hierarchical movemes. Specifically, we use unsupervised method to learn the finer grained annotations of the action states.

We also propose a multilayer hierarchical model to represent the structure of the optimization problem, which uses both the annotations in the dataset and the learnt hierarchical movemes.

Finally, we use different kinds of classifiers to train and test our proposed model, which provides insights into which kind of classifier is best suited for action prediction with hierarchical movemes.

V. Hierarchical Movemes

We propose a new representation called hierarchical

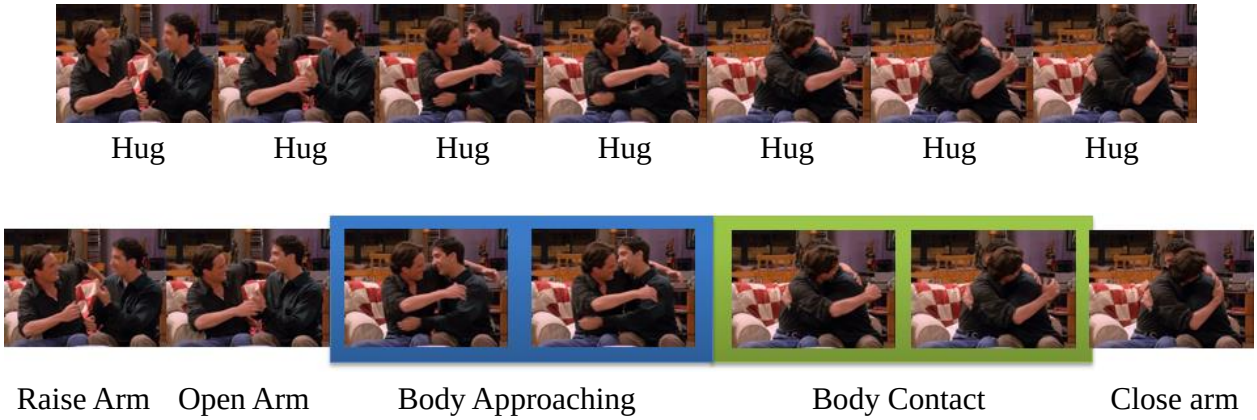

Fig.2. Up: The original coarse annotation of the actions; Bottom: The finer-grain representation (movemes) of the human actions

Fig.3 The model for the hierarchical movemes

movemes for future action prediction. The hierarchy depicts human movements at multiple levels of granularities from coarse to fine. Traditional action recognition methods focus on recognizing the higher level action classes. However, in action prediction, critical clues are usually hidden in finer grained motions. For example, an open arm usually implies hugging, but "open arm" is not necessarily an important class for action recognition, as shown in Fig. 2.

In this section, the hierarchical movemes model and the algorithm used for finding the fine-grained movemes will be described. The hierarchical moveme model constructed in this project has three layers, where the three layers consist of movemes with different levels of granularities as shown in Fig 3. In all three layers, the low level representations of the movemes are described by the HOG descriptors, which will serve as the input for our proposed algorithm for calculating the fine-grained movemes. The first layer of the hierarchical model consists of examples with the same action labels, ex. high five. The second layer further distinguishes between different orientations (viewpoints) of the person in the frame. These two layers utilize the annotations in the datasets so the classification is straightforward and supervised. Given "mid-level movemes" that correspond to movements of people with consistent viewpoints, our goal is to partition the examples in each mid-level moveme into multiple "fine-grained movemes" each corresponding to a specific human pose type (e.g. raise hand, reach, etc.). Thus, movemes at the bottom level capture viewpoint-specific and pose-specific characteristics of the future action. However, the labels for the bottom-layer moves are not given and have to be discovered from the training set. We describe the details in the following paragraphs.

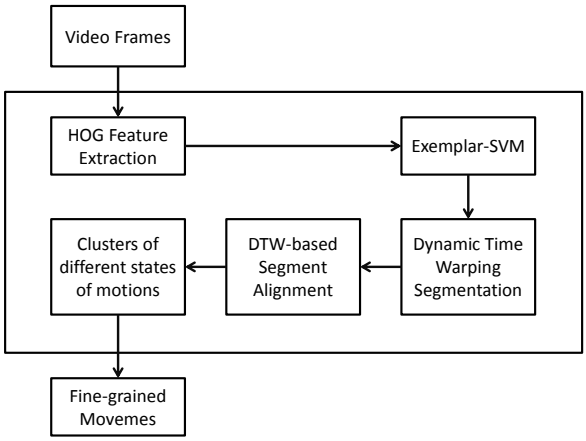

Fig.4 The block diagram of hierarchical movemes generations

A. System Block Diagram

The system block diagram for calculating the hierarchical movemes is shown in Fig. 4. After calculating the finer grained movemes for each action and orientation class, we can build our proposed hierarchical model as shown in Fig. 3 based on these movemes.

B. Ensemble of Exemplar-SVM models

Given the high dimensional HOG descriptors of each image, we would need to calculate the similarities matrix between each image for clustering in subsequent steps. The similarity matrix would be a K by K matrix, where the (i,j) entry in the matrix correspond to the scores of running the *i*th detector on the *j*th image. And the detector that we used here is the exemplar-SVM classifier.

The exemplar-SVM models [4] are based on training a separate linear SVM classifier for every exemplar in the training set. Each exemplar-SVM is defined by a single positive instance and many other negative samples. The negative examples are selected from the images not containing the same human activities. Each exemplar defines its own HOG dimensions respecting the aspect ratio of its bounding box. After calculating the exemplar-SVM models, each frame in an action category will be tested with the exemplar-SVM models in that same category. The output score from each model in that category can represent the input frame in a vector form. Specifically, if there are 200 positive examples in the "High-five" category, then each sampled frame in that category will be represented as a vector with 200 dimensions, where the elements of the vector are outputs from the 200 exemplar-SVM models trained from those 200 frames that fall in the High-five category.

The advantage of using the scores from the exemplar-SVM detectors is to separate frames corresponding to different subcategories of actions as far as possible, since these models are trained with negative examples from other

Fig.5 The alignments of the segmented sequence correspond to three states of the actions

subcategories. In order to predict human actions, the inputs for these models are the frames that are prior to the target human actions. To reduce the computation required, we do sampling by taking one frame out of every five frames.

C. Dynamic Time Warping (DTW) based Sequence Segmentation and Alignment

Once we have the similarity matrix, we cluster the frames of the person using a recently proposed temporal clustering algorithm [7]. We use a dynamic time warping (DTW) kernel to achieve the invariance of temporal order. The dynamic time warping method is used for both the segmentation of the video clips and the segment alignment. After representing each sampled frame as a vector using exemplar SVM models, dynamic time warping is used to segment the video frames in time, where the most similar frames will stay within the same segment.

After getting these segments, dynamic time warping will be used again for aligning two temporal sequences with different speeds. Based on the similarity matrix calculated in V.B and applying the dynamic programming algorithm, the optimal alignment of the segments in different video clips can be calculated. The reason we apply it here is to take the different speeds of action in different videos into consideration. And after segmentation and alignment, the number of alignments would be the number of finer-grained action states, i.e. atomic motion segments corresponding to the same pose type are merged into a fine-grained moveme as shown in Fig. 5.

D. Hierarchical Moveme Generation

After successfully clustering atomic motion segments corresponding to the same pose type across different video clips, we can train models based on the HOG descriptors for the fine-grained movemes at the bottom layer. The difference of the movemes at the bottom layer with other two layers is that it has the finest grained details of the action without the need of annotations in the dataset. The fine grained movemes can also be interpreted as the latent variables of the provided annotations. Visualization of the clustered images corresponding to one specific action state across different video clips is shown in Fig. 6. The blurs on the left column of the figure are due to input training samples from different video clips.

VI. Approaches

We learn a classifier for each moveme in the hierarchical structure given a hierarchy of movemes. Our goal in this

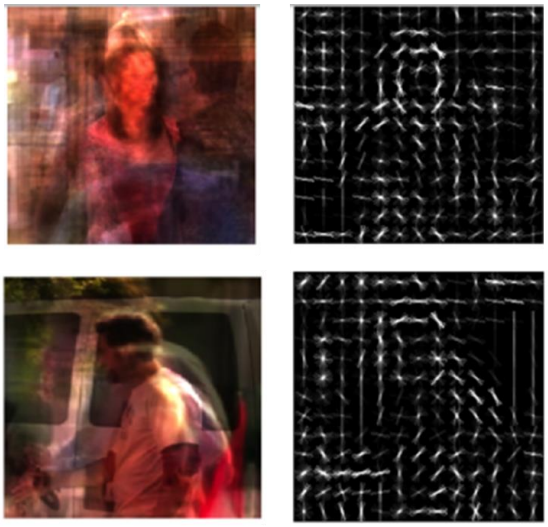

Fig.6 Visualization of the learnt fine-grained moveme templates based on the HOG descriptors. The upper row is the visualization of a "Hug" and the lower row is the visualization result of shaking hands

project is to predict future actions based on a single frame. For each moveme, we learn a classifier based on the appearance features, i.e. the HOG descriptors. Multiple classifiers were explored, including simple CART, naïve Bayes, random forest, and SVM. In addition, we also explored using only the first layer, the first two layers, and all three layers in order to justify our claim that using finer-grained movemes (the 3rd layer) helps improve the prediction accuracy. Our scoring function for labeling an example X with movemes Y is written as:

$$\Phi(X,Y) = \sum_{i=1}^{L} \alpha_{yi}^{T} \emptyset(\mathrm{X}, \mathrm{yi}) + \sum_{i=1}^{L-1} \beta_{yi,y_{i+1}}^{T} \varphi(y_i, y_{i+1}) \quad (1)$$

where X is the feature vector for the person in a frame, which is associated with labels corresponding to one branch of the movemes hierarchy: Y = $\{y_1, y_2, \ldots y_L\}$. L is the number of hierarchies, i.e. L=3 in our case. y_1 corresponds to the future action label, y_2 corresponds to the label of a future action with a particular viewpoint, and y_3 corresponds to the fine-grained moveme label that is automatically discovered by our clustering algorithm.

A. Scoring Function

The scoring function can be separated into two parts: the unary model part and the pairwise model part.

(1) Unary model $\alpha_{y_i}^{T} \emptyset(\mathrm{X}, \mathrm{y}_i)$:

This potential function captures the compatibility between the feature X and the moveme y_i. $\emptyset(X, yi)$ denotes the response of running the moveme classifier of y_i on the frame feature vector X. We can learn the unary model utilizing multiple standard machine learning methods, which we will explain later in this section.

(2) Pairwise model $\beta_{yi,y_{i+1}}^{T} \varphi(y_i, y_{i+1})$

This potential function captures the co-occurrence between a pair of movemes across different levels of the hierarchy. $\varphi(y_i, y_{i+1})$ is set to 1 if there is an edge between moveme y_i and moveme y_{i+1} in the hierarchy. Otherwise, $-\infty$. This means we exclude the co-occurrence of certain pairs of movemes: e.g. a person cannot be described by movemes

corresponding to the prior observations of different actions at the same time. $\beta^T_{yi,y_{i+1}}$ is the model parameter that favors certain pair of movemes. $\varphi(y_i, y_{i+1})$ is the same across different machine learning models since we are using the same training dataset and thus the same hierarchical structure, while parameters $\beta^T_{yi,y_{i+1}}$ are learnt and can be different.

For an example X that corresponds to a person in a single frame, our goal is thus solving the optimization problem:

$$Y = \text{argmax}_{yi:i=1\ldots L}\, \Phi(X, Y) \quad (2)$$

The inference for example X is on a chain structure where we jointly infer moveme labels at all levels together using Belief Propagation. The moveme at the top layer of the hierarchy y_1 corresponds to the future action label of the person and is used to text prediction accuracy. Our inference procedure also returns more detailed predictions of the person (e.g. viewpoint, temporal state) through movemes at finer-grained layers of the hierarchy (i.e. latent variables in our model): $\{y_2, \ldots y_L\}$.

After specifying the scoring function and the optimization problem we aim to solve for each example X, we now describe different methods we utilized to learn the model parameters α^T_{yi} and $\beta^T_{yi,y_{i+1}}$ with different L, i.e. L = {1, 2, 3}.

B. Using 1st layer moveme (L=1)

A coarse-level moveme models generic pose and viewpoint characteristics of a certain action that is going to happen in the future. Each frame within a moveme is associated with the same future action label, ex. hug. The models are trained on top of the features to predict how likely the person will perform an action in the near future. Thus, this is a standard multi-class classification problem with no pairwise model part. The models we utilized are briefly described below.

(1) SimpleCART

CART stands for Classification and Regression Trees [8]. SimpleCART builds a decision tree using recursive partitioning routine. Tree-based methods partition the feature space into a set of rectangles, and then fit a simple model (like a constant) in each one. To simplify matters, we restrict attention to recursive binary partitions. The space is first split into two regions, and the response is modelled by the mean of Y (label) in each region. We choose the variable and split-point to achieve the best fit. Then one or both of these regions are split into two more regions, and this process is continued, until some stopping rule is applied [9].

(2) Naïve Bayes

A naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. The central idea of Naïve Bayes model is presented in the following equation

$$p(C|X_1, X_2, \ldots . X_n) \propto$$
$$p(C) * p(X_1, X_2 \ldots X_n|C) \;=\; p(C) * \prod_{i=1}^{n} p(X_i|C) \quad (3)$$

where C is the class variable and $X_1, X_2, \ldots . X_n$ are the feature variables from example X. The above is true under the assumption that each X_i is conditionally independent of every other X_j where j≠i.

(3) Random Forest

Random forests are an ensemble learning method for classification (and regression) that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes output by individual trees. The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set $M = M_1, \ldots, M_n$ with responses Y = y_1 through y_m, bagging repeatedly selects a bootstrap sample of the training set and fits trees to these samples. Specifically, for b = 1 through B:

a) Sample, with replacement, n training examples from M, Y; call these M_b, Y_b. Train a decision or regression tree T_b on M_b, Y_b.

b) After training, predictions for unseen samples M' can be made by taking the majority vote in the case of decision trees or by averaging the predictions from all the individual regression trees on M':

$$\hat{y} = \frac{1}{B} * \sum_{b=1}^{B} T_b(M') \quad (4)$$

Random forests use a modified tree learning algorithm that selects, at each candidate split in the learning process, a random subset of the features. The reason for doing this is the correlation of the trees in an ordinary bootstrap sample, i.e. if one or a few features are very strong predictors for the response variable (target output), these features will be selected in many of the B trees, causing them to become correlated. Typically, for a dataset with p features, $\sqrt{p}$ features are used in each split.

(4) Support Vector Machine

A multi-class SVM with linear kernel is trained to predict the action label for each frame. However, instead of using a standard loss function of Structural SVM, i.e. the $0-1$ loss, which equally penalizes all incorrect predictions at any time prior to the future action, we introduce a new loss function that depends on the temporal distance to the future action:

$$Loss = \begin{cases} 1 - \mu t, if\ the\ prediction\ is\ wrong \\ 0,\ otherwise \end{cases}, \quad (5)$$

where $t \in (0, T]$ is the temporal distance to the starting point of the action we wish to predict, and t = 0 corresponds to the first frame of the action. T is the maximum number of frames before the action that we consider. $\mu \in (0, 1/T]$ is a tuneable parameter. The original 0-1 loss is inadequate for the task of future action prediction, since prediction from a frame at a long time before the start point of an action is obviously more difficult than from those at a few frames before the action takes place. If we treated them equally in training, the learnt decision boundaries might become unreliable. Using the new loss, incorrect prediction from frames longer before the action takes place receives fewer penalties.

C. Using 1st and 2nd layer movemes (L=2)

A mid-level moveme models viewpoint-specific but pose-generic characteristics of the future action. Each motion segment within a moveme is associated to the same viewpoint and future action label. The differences between L=2 and L=1 include:

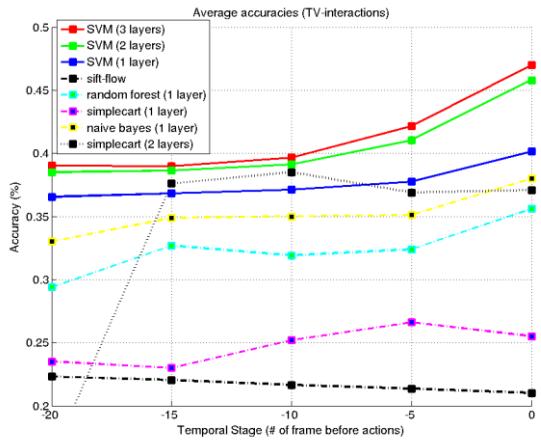

Fig.7 TV interaction actions prediction accuracy

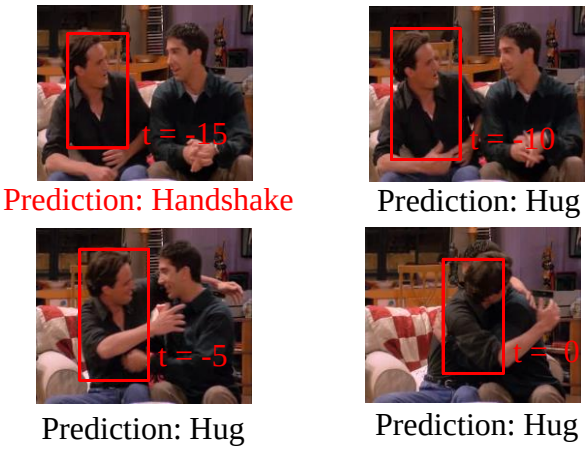

Fig.8 Result Visualization for different time t

a) We now also have to learn the parameters α_{y2}^{T} to predict the orientation (y_2) for X in the unary model.

b) The pairwise model is incorporated into the scoring function for L=2 so that $\beta_{yi,y_{i+1}}^{T}$ have to be learnt for features $\varphi(y_i, y_{i+1})$. SimpleCART and SVM, both have already been described earlier, are utilized again to learn the parameters.

D. Using movemes from all 3 layers (L=3)

A fine-grained moveme models viewpoint-specific and pose-specific characteristics of the future action. Each atomic motion segment within a moveme is associated with the fine-grained moveme label automatically discovered in the discriminative clustering process. Since SVM yields the best accuracy for both L=1 and L=2, we use SVM again for L=3. The differences between L=3 and L=2 include:

a) We now also have to learn the parameters α_{y3}^{T} to predict the fine-grained moveme label y_3 for X in the unary model. While y_1 and y_2 labels are annotated in the dataset, y_3 is not given but automatically discovered by our clustering algorithm.

b) In addition to β_{y1,y_2}^{T}, we now also have to learn β_{y2,y_3}^{T}, i.e. how we favor certain pair of movemes from layer 2 and 3.

VII. Experimental Results

Our goal is to test the performance of the proposed method on future action prediction in the challenging real world scenarios. We choose a challenging dataset collected from TV shows as described in Section II. We use the training/testing split provided along with the dataset. For training, we sample a collection of frames from all of the videos in the training set, which contains more than 25, 000 person examples.

TABLE II
SUMMARY OF RESULTS

Method	Number of layers	Mean class accuracy (%)
SimpleCART	1	25.2
Random Forest	1	31.9
Naïve Bayes	1	35.1
SVM	1	37.1
SimpleCART	2	38.5
SVM	2	39.1
SVM	3	39.5

For testing, we predict future actions from a single video frame with different settings on the temporal distances to the start point of the action we wish to predict. Specifically, we measure the performances with 5 different temporal stage settings, from −20 to 0, with a step size of 5. The numbers denote the temporal distance (in frames) from the input image to the start point of the action. For example, the methods' performances at a temporal stage −20 describe the classification accuracies given all of the testing frames within 20 frames before the start point of the action we wish to predict. The temporal stage of 0 indicates all testing images are taken within 5 frames after the start point of the action, making the problem a conventional action classification problem.

In addition to the methods described in Section VI, we also implemented siftflow [6] in order to compare with the results using other methods. Given a testing image, it first finds the nearest neighbor from the training data using the SIFT flow algorithm, which matches densely sampled SIFT features between the two images, while preserving spatial discontinuities. The future action label of the matched training image is directly transferred to the testing image.

The comparison of results is shown in Fig. 7. The 3-layer prediction result outperforms all the other methods at all different temporal settings. As for 2-layer methods, SVM outperforms SimpleCART. As for 1-layer methods, SVM has the best performance, followed by Naïve Bayes, random forest and then SimpleCART. Siftflow has the worst performance. Comparing the same method using different numbers of layers, we can see that 2-layer SimpleCART has 14.7% higher accuracy than 1-layer SimpleCART. In addition, 3-layer SVM is better than 2-layer SVM, which is better than 1-layer SVM. The resulting prediction accuracies using different models at t=-10 is shown in Table II for ease of comparison.

There is also a notable performance increase of our full model (3-layer SVM) as well as the 2-layer SVM moveme model, starting from 10 (around 0.5 s) frames before the action is executed. This is because the fine-grained appearance that characterizes the actions tends to appear around 10 frames before the action is executed. And the visualization of our prediction results are shown in Fig. 8.

VIII. Discussion

There are some possible improvements for the current method. The proposed framework has a deterministic final predicting result. However, it would be more interesting if we can propose a model that predicts the probability distribution of the human action, e.g. predicting the future human action with a probability of 0.9 that he would hug and with probability of 0.1 that he would shake hands with others. The other possible extension of the hierarchical movemes is that since we have already known the fine-grained action state of the actions, we can use this to predict how much time later the person would have really executed the action.

The other observation from the experimental result is that linear SVM outperforms the other classifiers. This may be due to the fact that our classification problem does not have a complicated decision boundary and thus SVM does a good job separating frames corresponding to different labels. Of course, the performance of different classifiers may vary as the setting of the problem and the dataset vary.

The best result we achieve is approximately 46.7%, and we have demonstrated that our proposed model and the finer grained movemes can improve the performance. However, the prediction accuracy can depend heavily on the dataset. Since the dataset we use is collected from YouTube and consists of 20 different TV shows, there is no clean background, which can affect the performance of the prediction. Thus, we also test the proposed method on the UT-Austin TV interaction dataset [10], and it has been shown that an accuracy of more than 83% is achieved on that dataset.

IX. Conclusion

We have presented hierarchical movemes - a new representation for predicting future action from still images in unconstrained data. Different movemes in our representation capture human movements at different levels of granularity. Movemes are organized in a structured hierarchical model and the model parameters are learned in a max-margin framework considering the temporal distance to the future action. Our experimental results demonstrate that our model is effective in capturing the fine-grained details that improve the accuracy for future action prediction.

X. Acknowledgement

We want to thank Professor Savarese and Dr. Lan from Stanford CVGL for providing us the dataset as well as giving us guidance on how to interpret the data.

XI. Reference

[1] WHO: http://www.who.int/ageing/en/

[2] Ryoo, M.S.: Human activity prediction: Early recognition of ongoing activities from streaming videos. In: IEEE International Conference on Computer Vision. (2011)

[3] Wang, Z., Deisenroth, M., Amor, H.B., D. Vogt, B.S.: Probabilistic modeling of human movements for intention inference. In: Robotics: Science and Systems (RSS). (2013)

[4] Koppula, H.S., Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. In: Robotics: Science and Systems (RSS). (2013)

[5] Luber, Matthias, et al. "People tracking with human motion predictions from social forces." Robotics and Automation (ICRA), 2010 IEEE International Conference on. IEEE, 2010.

[6] Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: SIFT flow: Dense correspondence across different scenes. In: European Conference on Computer Vision. (2008)

[7] F. Zhou, F.D.l.T., Hodgins, J.K.: Hierarchical aligned cluster analysis for temporal clustering of human motion. PAMI (2013)

[8] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J Stone. Classification and Regression Trees. Wadsworth, Belmont, Ca, 1983.

[9] J.H. Friedman, T. Hastie, R. Tibshirani, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Verlag, Heidelberg, 2001.

[10] Ryoo, M., Aggarwal, J.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: ICCV. (2009)