Early Action Prediction by Soft Regression

Jian-Fang Hu, Wei-Shi Zheng, Lianyang Ma, Gang Wang, Jianhuang Lai, and Jianguo Zhang

Abstract—We propose a novel approach for predicting on-going action with the assistance of a low-cost depth camera. Our approach introduces a soft regression-based early prediction framework. In this framework, we estimate soft labels for the subsequences at different progress levels, jointly learned with an action predictor. Our formulation of soft regression framework 1) overcomes a usual assumption in existing early action prediction systems that the progress level of on-going sequence is given in the testing stage; and 2) presents a theoretical framework to better resolve the ambiguity and uncertainty of subsequences at early performing stage. The proposed soft regression framework is further enhanced in order to take the relationships among subsequences and the discrepancy of soft labels over different classes into consideration, so that a Multiple Soft labels Recurrent Neural Network (MSRNN) is finally developed. For real-time performance, we also introduce a new RGB-D feature called "local accumulative frame feature (LAFF)", which can be computed efficiently by constructing an integral feature map. Our experiments on three RGB-D benchmark datasets and an unconstrained RGB action set demonstrate that the proposed regression-based early action prediction model outperforms existing models significantly and also show that the early action prediction on RGB-D sequence is more accurate than that on RGB channel.

Index Terms—Early action prediction, RGB-D, soft regression

1 INTRODUCTION

Recognizing actions before they are fully executed in real-time is very important for some real-world applications like visual surveillance, robot designing [26], [27], and clinical monitoring [28]. *Early* action prediction is to predict the label of *an on-going action* using the observed subsequences that only contain partial action execution. While action recognition is a long-term research topic with considerable progress on developing robust spatiotemporal features (Cuboids [6], interest point clouds [1], HOG3D [21], dense trajectory [53], and two-stream CNN [47], [51], [56], [58] etc.) and feature learning techniques (sparse coding [65], max-margin learning [12], [69], Fisher vector [53], rank pooling [8] etc.), conventional action recognition aims at developing algorithms and systems for after-of-the-fact prediction of human action (i.e. when action sequence is entirely observed), and thus these action recognition methods don't seek to build models for early action prediction at different progress levels, which in particular requires modeling the intrinsic expressive power of subsequences at different progress levels and processing them in real-time.

While there exists work for early action prediction [2], [29], [41], when applied in the deployment stage, most of them require manually labeling the progress level of an on-going video segment that tell how much an action has finished. However, this makes the system not applicable in practice, since the system will not have a chance to access the progress level of an ongoing action sequence until the action has been completely executed. Although, instead of labeling progress level, an alternative way is to simply label a subsequence[1] as the class label of the full sequence [67], sometimes, this naive labeling would make early action prediction ambiguous. An action sequence consists of several segments, and segments from different actions could be similar at its early performing stage, so that partially observed sequence could often be ambiguous in predicting the class label of the expected full action sequence. Taking Figure (1) as an example, the subsequence "*taking out a cell-phone*" appears in both the action sequences "*calling with a cell-phone*" and "*playing with a cell-phone*", from which it is hardly to tell the difference between the two actions. We argue that such an uncertainty and ambiguity should be considered in the modeling stage. If, in those cases, these were made certain by naively assigning hard labels (e.g.,simply treating the subsequence of taking out phone in the first row as "*calling with phone*" while defining the similar subsequence in the second row as "*playing with a cell-phone*".), it will *not* benefit the predictor learning.

To address the above problems, we formulate a soft regression based early action prediction framework. In this framework, we learn a soft label for the subsequence at each progress level. The learned soft label tells how likely the subsequence is performing the action depicted in the corresponding full sequence, and the learned soft label thus allows similar labeling for any two subsequences at the same or similar progress level. The soft label would be automatically updated as more parts of an action are observed such that it tends to be the real action class label. In order to enable soft label to work in such a way, we learn it jointly with early action predictor in a soft regression framework, where subsequences with partial action executions are also employed for refining the predictor learning. By using our model, the usual

- *J.-F. Hu, W.-S. Zheng, and J. Lai are with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. J.-F. Hu is also with the Guangdong Province Key Laboratory of Computational Science, Guangzhou, China. W.-S. Zheng is also with the Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University), Ministry of Education, China, and Inception Institute of Artificial Intelligence, United Arab Emirates.*
  E-mail:hujianf5@mail.sysu.edu.cn, wszheng@ieee.org and stsljh@mail.sysu.edu.cn
- *L.-Y. Ma is with the Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China and Tencent, Beijing, China.*
  E-mail:lianyangma2012@gmail.com
- *G. Wang is with Alibaba AI Labs.*
  E-mail:gangwang6@gmail.com
- *J. Zhang is with the School of Science and Engineering (Computing), University of Dundee United Kingdom.*
  E-mail:j.n.zhang@dundee.ac.uk

[1] In this work, the subsequence of an action means the accumulation of consecutive segments from the start of the action.

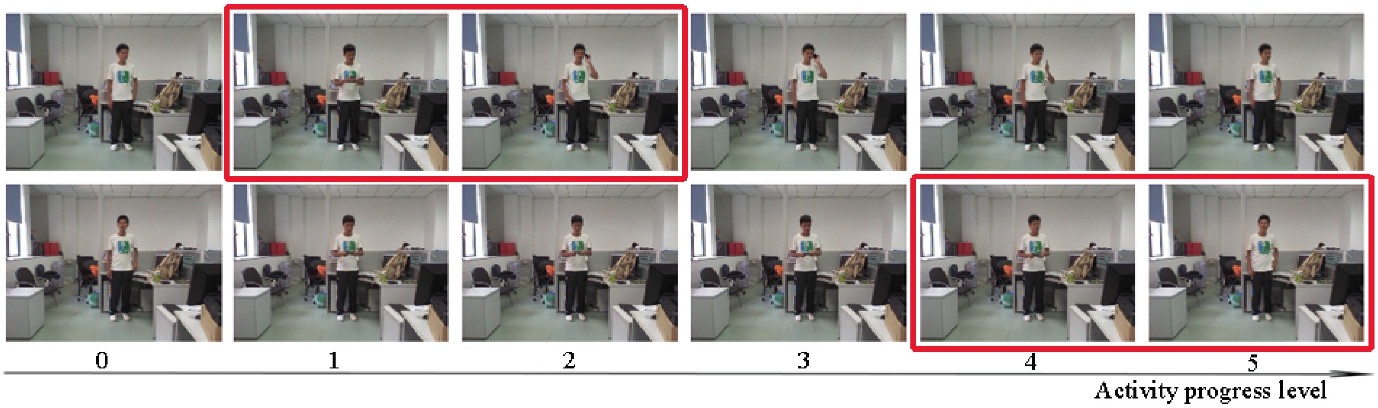


Fig. 1. Snapshots from activities *calling with a cell-phone* (the first row) and *playing with a cell-phone*(the second row). As presented in the first row, it is hard to recognize the action when its progress level is less than 2. However, if the segment with temporal interval [1, 2] (marked with red box) are provided, it becomes clear that the subject is performing the action *calling with a cell-phone*. We also observe that the subsequences at progress level 1 (temporal interval [0, 1]) in the two activities contain the same action "taking out a cell-phone".

assumption in early action prediction on given the progress level of subsequence in the testing stage [2], [23] is no longer necessary.

We further explore specifically the temporal relationship between successive frames of on-going subsequences by developing a soft RNN-based framework, and thus a Soft-RNN regression model is developed. Since subsequences at the same progress level from different actions have different predictive powers for their underlying actions, that means soft labels could be different for different actions at the same progress level, we finally develop a Multiple Soft labels Recurrent Neural Network (MSRNN).

Our early action prediction model works on RGB-D channels, while existing ones are relying on using RGB videos by matching visual appearance and human motion among the activities executed at different progress levels [2], [28], [29], [41], [63]. However, RGB features are intrinsically limited in capturing highly articulated motions due to the inherent visual ambiguity caused by clothing similarity among people, appearance changes from view point difference, illumination variation, cluttered background and occlusions [13], [55]. The recently introduced low-cost depth cameras can alleviate the ambiguity due to the availability of more modal data for describing action such as depth of scene and 3D joint positions of human skeleton. Hence, in this work, we further explore real-time early action prediction with the assistance of depth sensors.

In addition, towards making our prediction model work on RGB-D sequence in real-time, we design local accumulative frame feature (LAFF) to characterize the action context of RGB-D sequence with arbitrary action progress levels. The RGB-D context will include the appearance, shapes and skeletons of human body parts, manipulated objects and even scene (background). By employing the popularly used integral map computing technique, we demonstrate that the formulated LAFF can be efficiently computed in a recursive manner and thus suitable for real-time prediction. As shown by the flowchart of our method in Figure 2, our proposed method can process more than 34 frames per second on a normal PC using MATLAB without code optimisation and the use of more efficient programming languages, which can be made for real-time early action prediction.

In summary, the main contributions of this work are multi-fold: 1) a soft regression (SR) framework is formulated for early action prediction; 2) both conventional and deep soft regression-based early prediction models are developed; 3) a local accumulative frame feature (LAFF) is introduced for real-time early action prediction; 4) our method is tested on three RGB-D action sets and a unconstrained RGB action set (i.e., UCF101), and obtains more reliable performances in predicting activities at varied progress levels. The results have shown the benefit of modeling based on soft labels for overcoming the challenge of early action prediction and demonstrated that the early prediction on RGB-D sequences works much better than that on RGB videos only.

2 Related Work

Early Action Prediction. In many real-world scenarios like surveillance, it would be more important to correctly predict an action before it is fully executed. Many efforts are on developing early action detectors or future action prediction systems [11], [20], [23], [33], [41], [52], [63]. For example, Hoai et al. and Huang et al. explored the application of max-margin learning in early event recognition and detection [11], [17]. Ryoo developed an early action prediction system according to the change of feature distribution as more and more video streams are observed [41]. Lan et al. proposed to represent human movements in a hierarchical manner and use a max-margin learning framework to select the most discriminative features for prediction [28]. Li et al. proposed to mine some sequential patterns that frequently appear in the training samples for prediction [29]. Vondrick et al. intended to predict actions by anticipating the features of future video frames in a unsupervised manner [52]. However, they assumed that the progress level of ongoing action is provided along with the observed sequence in the testing phase, which renders their method less applicable in the real-world applications, since it is hard to access the progress level of ongoing action until it has been fully executed and observed when applied on unknown sequence. Recently, some researchers intend to conduct early action prediction [22] or detection [37] by learning action progress levels in an automatic manner. Yet, their prediction/detection is easily hindered by the performance of learning action progression, which is still a challenging problem needed to be carefully addressed. In contrast, we aim at learning an early action predictor that can be used for predicting action at any progress level. Thus our system does not need to learn the progress level of ongoing actions while performing early action prediction.

Recent research on human early action prediction is mainly focusing on predicting activities based on on-going RGB videos

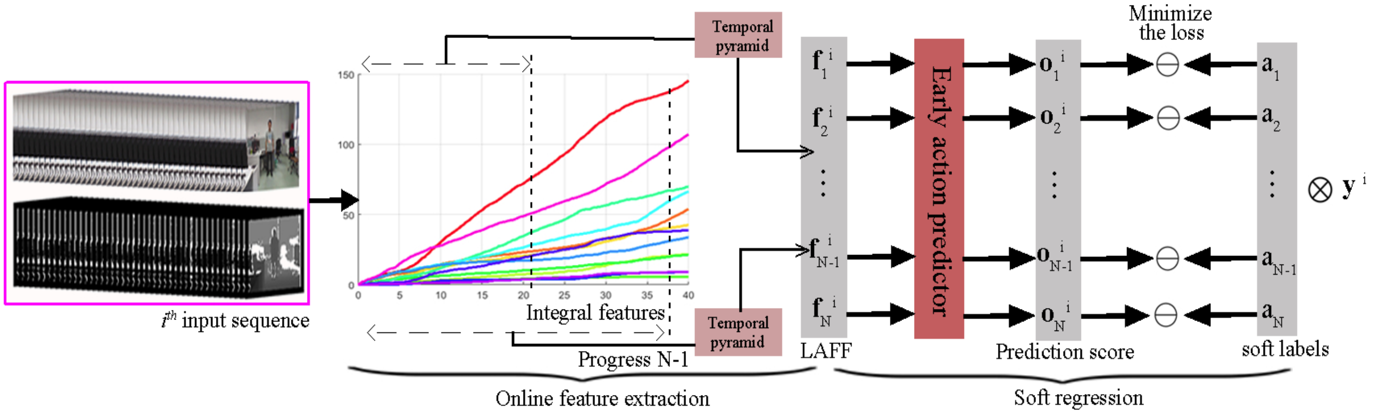


Fig. 2. A graphical illustration of the training principle for our soft regression based early action prediction framework. In this framework, we develop a novel online RGB-D action feature extractor and a soft regression model. The early action predictor and soft labels are learned jointly. In the figure, operator $\otimes$ is a Kronecker product and $\ominus$ for the loss computation. Each curve in the second block (from the left hand side) corresponds to one dimension of the computed integral features. For clarity and simplicity, we only present a small set of the extracted integral features.

[11], [23], [41], [63], while less work has been reported on RGB-D sequences captured by low-cost depth cameras. In this work, we consider the early prediction of RGB-D action sequence and develop a real-time system for predicting human activities without any extra prior information about the progress level of on-going action when applied in practice. The most closest to our approach is the online RGB-D early action prediction system developed in [67]. However, the system in [67] is based on frame-level prediction and the long-term motions are discarded in their model. Moreover, the subsequences with partial action executions are not exploited for prediction, which renders their method less accurate for early action prediction.

Action recognition with depth cameras. The emergence of Kinect device has lit up the research of human action recognition with depth cameras in these years. In the literatures, how to acquire a robust feature representation for the depth sequences is one of the most fundamental research topics. A lot of RGB video descriptors have been extended in order to characterize 3D geometries depicted in depth sequences [5], [36], [39], [54], [66]. For example, Oreifej et al. developed their depth descriptors by extending the idea of constructing histogram of oriented gradient [5], [39], [66]. Considering the close relationship between human pose(skeleton [44]) and action, some researchers seeked to represent human activities using positional dynamics of each skeleton joint [7], [18], [62] or joint pairs [32], [38], [43], [64], [68]. Human action may contain complex interactions between the actor and objects, using depth and skeleton channels is not sufficient for describing the interactions. RGB channel is also utilized for feature representation [13], [55], [59]. In this work, we construct a RGB-D sequence feature by combining the local descriptors extracted from color patterns, depth patterns and skeletons. Different from the previous work that extracts features for off-line computation [13], [14], [55], [59], we formulate our feature modeling in a recursive manner so that it can be computed in real-time. The conception of soft label has been recently explored in [16] for improving RGB-D action recognition but not for early action prediction, where the authors allow the human annotators to assign a soft label for the video segment with ambiguity.

Recurrent Neural Network for Sequential Prediction. Recurrent Neural Networks (RNN) have been widely used to address the sequential prediction problems in literatures, such as speech recognition [9], human action/activity recognition [34], [35], [42], scene labeling [40], [46], image caption [19], and object segmentation [31]. RNN and its variants LSTM [34], GRNN [4], etc. have achieved promising progresses on modeling temporal dependency by sharing weights among the sequential data and explicitly transferring information from the current time-step to the next. The information passing strategy enables RNN models to have a deep architecture over time and thus can model the complex dependencies between temporal states. Most of existing RNN models are developed for the completely executed sequences, and don't seek for predicting ongoing subsequences with partial action executions. In this work, we focus on developing soft RNN regression based early action prediction models without using any progress level labeling of on-going sequence in testing, where we embed learning soft label into the RNN framework on early action prediction. Moreover, we further consider the discrepancy of soft labels over different classes, and a Multiple Soft labels Recurrent Neural Network (MSRNN) model is presented in this work.

A preliminary version of this work was reported in [15]. In this work, we have further extended our soft regression-based early action prediction model in the following three aspects. Firstly, two more advanced deep soft regression models are further formulated based on RNN, which are capable of 1) capturing the complex dependency between successive action subsequences and 2) taking the discrepancy of soft labels over different classes into consideration, and thus can obtain better prediction performance. Secondly, we have reported more analysis on comparing our proposed three soft regression-based early action prediction models from non-deep to deep in order to show the extra benefits of taking more cues by soft RNN regression for early action prediction. Thirdly, we have conducted more experiments and reported extensive comparison on one additional large scale dataset consisting of a set of complex actions [42].

3 Our Approach

3.1 Problem Statement

We concern a real-time early action prediction system for predicting on-going action sequence. In early action prediction, the action depicted in observed sequence is always uncompleted before it has been fully executed. Unlike the early action prediction considered in [2], [23], we do not hold the assumption that the progress level of on-going action is known when applied in applications (i.e. testing phase), since it is hard (if not impossible) to access progress level of on-going action until it has been fully observed. In this work, we propose a soft regression-based prediction framework that can be generally used for performing early prediction of an (on-going) action sequence at any progress level.

Notation. Throughout this paper, we use bold uppercase characters to denote matrices and bold lowercase characters (or Greek letters) to denote vectors. For any matrix $\boldsymbol{A}$, we use $\boldsymbol{A}(i,\cdot)$, $\boldsymbol{A}(\cdot,j)$ and $\boldsymbol{A}(i,j)$ to denote the i^{th} row, the j^{th} column and the (i,j)-element of $\boldsymbol{A}$, respectively. $\boldsymbol{A}^T$ denotes the transpose matrix of $\boldsymbol{A}$. In this work, we denote the Frobenius norm of a matrix $\boldsymbol{A}$ as $||\boldsymbol{A}||_F$ and the $L_{1,2}$-norm as $||\boldsymbol{A}||_{1,2}$ and the l_2 norm of a vector $\boldsymbol{a}$ as $||\boldsymbol{a}||_2$. The $L_{1,2}$-norm for matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is defined as:

$$||\boldsymbol{A}||_{1,2} = \sum_{j=1}^{n} \sqrt{\sum_{i=1}^{m} \boldsymbol{A}(i,j)^2} = \sum_{j=1}^{n} \boldsymbol{r}(j). \tag{1}$$

Here, $\boldsymbol{r}(j)$ represents the l_2 norm of the j^{th} column $A(\cdot,j)$. Then we can obtain the generalized gradient[2] $\frac{\partial ||\boldsymbol{A}||_{1,2}}{\partial \boldsymbol{A}(i,j)} = \frac{\partial \boldsymbol{r}(j)}{\partial \boldsymbol{A}(i,j)} = \frac{\boldsymbol{A}(i,j)}{\boldsymbol{r}(j)}$. This equation indicates that the gradient of $||\boldsymbol{A}||_{1,2}$ with respect to $\boldsymbol{A}$ can be easily obtained by performing a column-wise normalization on the matrix $\boldsymbol{A}$. Here we denote this column-wise normalization operator as $\phi(\cdot)$ for convenience.

3.2 Local Accumulative Frame Feature (LAFF)

Since existing RGB-D sequence features [13], [55], [59] are not for online feature extraction, they are less applicable for online early action prediction. To address this issue, we propose an effective RGB-D sequence feature representation by employing the widely used integral map computing technique. In detail, we extract local HOG descriptors from RGB and depth patches around each body part and extract relative skeleton features for each frame in order to capture action contexts including human motions, appearance, shapes of human body parts, the manipulated objects and even the scene (background).

Inspired by the successful use of spatial pyramid for modeling the spatial structures among different image patches, we construct a three-level temporal pyramid to capture temporal structures by repeatedly partitioning the observed sequence into increasingly finer sub-segments along temporal dimension. The features of the frames found in each sub-segment are accumulated together using a mean pooling method. The concatenation of all accumulative features forms our local accumulative frame feature (LAFF).

In the following, we show that LAFF can be calculated efficiently by constructing an integral feature map $\boldsymbol{Int}$:

$$\boldsymbol{Int}(\cdot,T) = \sum_{t=1}^{T} \boldsymbol{F}(\cdot,t), \tag{2}$$

where $\boldsymbol{F} \in \mathbb{R}^{d\times T}$ is the local features extracted from the frames in a sequence, d denotes the feature dimension, and T is the total number of observed frames. Note that the integral map $\boldsymbol{Int}$ could be computed recursively along the temporal dimension. Based on $\boldsymbol{Int}$, we can compute the accumulative feature $\boldsymbol{x}$ between frames t_1 and t_2 ($t_2 > t_1$) as follows:

$$\boldsymbol{x} = \frac{\boldsymbol{Int}(\cdot,t_2) - \boldsymbol{Int}(\cdot,t_1-1)}{t_2 - t_1 + 1}. \tag{3}$$

Therefore, the LAFF features with seven temporal intervals (1+2+4 sub-segments in the three-level pyramid) can be efficiently computed from the formulated integral feature map using Eq. (3). This enables online and real-time computation.

3.3 Soft Linear Regression (SLR) based Early Action Prediction

We assume that each *training* action sequence contains complete action execution. To train an action predictor, similar to existing works [23], [41], we uniformly partition each fully observed training sequence into N segments of equal length. Let $V(\cdot,\cdot,\cdot)$ be the full sequence, and we use a vector[3] $\boldsymbol{\pi} \in \mathbb{R}^{N+1}$ to indicate the temporal locations of the segments. For example, $V(\cdot,\cdot,\boldsymbol{\pi}(1):\boldsymbol{\pi}(2))$ represents the sequence of the first segment. Always, we call $V(:,:,\boldsymbol{\pi}(1),\boldsymbol{\pi}(n+1))$ an action's subsequence with *progress level* n. And correspondingly, its *observation ratio* can be defined as $\frac{n}{N}$.

Let $\{(\boldsymbol{X}_1,\boldsymbol{y}_1),(\boldsymbol{X}_2,\boldsymbol{y}_2),...,(\boldsymbol{X}_I,\boldsymbol{y}_I)\}$ be the training dataset that consists of I examples from C classes, where $\boldsymbol{y}_i \in \mathbb{R}^C$ is a label vector of $\boldsymbol{X}_i$, each $\boldsymbol{X}_i \in \mathbb{R}^{d\times N}$ has N instances, and each instance $\boldsymbol{X}_i(\cdot,n)$ is represented by the LAFF feature of the subsequence of progress level n. The label vector $\boldsymbol{y}_i$ is a binary vector, having its j^{th} entry set to 1 if it is from the j^{th} class and 0 otherwise.

Indeed, a subsequence is ambiguous, because action sequences of different types may contain similar subsequences and the unobserved duration of an action sequence could contain some important cues for identifying the whole action. Thus, labeling it as the label of its full sequence could cause confusion. To overcome this problem, we learn a soft label for each subsequence and define the label of the subsequence with a progress level n as $\boldsymbol{\alpha}(n)\boldsymbol{y}_i$ where $0 \leq \boldsymbol{\alpha}(n) \leq 1$. $\boldsymbol{\alpha}(n)\boldsymbol{y}_i$ can be conceived as how likely the subsequence is from the action class $\boldsymbol{y}_i$. Using and learning soft labels can alleviate the confusion caused by the fact that subsequences from different actions could contain similar action elements (See Figure 1 for example). In addition, it also enables the prediction of ongoing action at any progress level in our modeling.

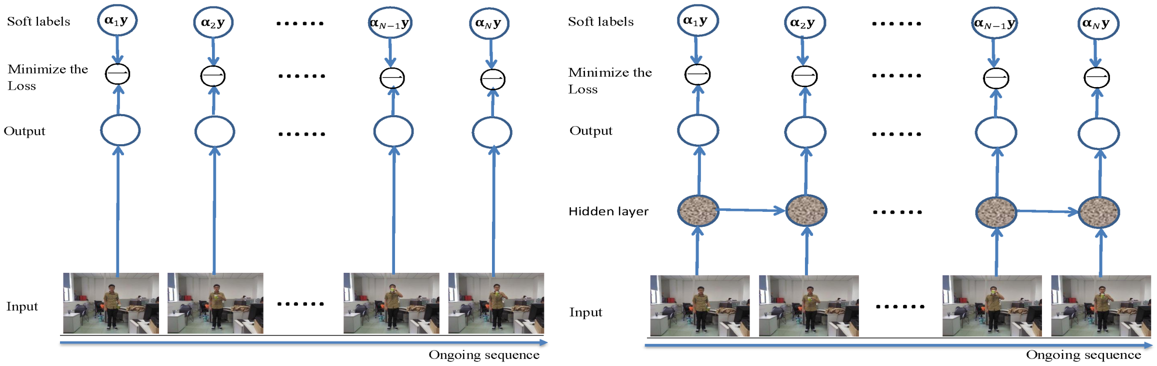
To learn the soft labels rather than setting them empirically and manually, we form a soft linear regression (SLR) model for learning soft labels and action predictor jointly as follows:

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{\alpha}} \sum_{i=1}^{I}\sum_{n=1}^{N} & \overbrace{\boldsymbol{s}(n)||\boldsymbol{W}^T\boldsymbol{X}_i(\cdot,n) - \boldsymbol{y}_i\boldsymbol{\alpha}(n)||_{1,2}}^{Prediction\ loss\ term} + \frac{\xi_2}{2}||\boldsymbol{W}||_F^2 \\ s.t. \quad & \boldsymbol{\alpha}^T\boldsymbol{e}_N = 1, 0 \leqslant \boldsymbol{\alpha} \leqslant 1, \xi_2 \geq 0, \end{aligned} \tag{4}$$

where $\boldsymbol{W} \in \mathbb{R}^{d\times C}$ is the transformation matrix of a multi-class discriminative linear predictor and it is constrained by a conventional ridge regularization. Since the prediction loss of subsequences at different progress levels should contribute differently to the prediction, we introduce $\boldsymbol{s}(n)$ to weight the regression loss of each subsequence. $\boldsymbol{s}$ is set to monotonically increase w.r.t n, and its effect is tested in Section 4.5. By letting $\boldsymbol{S}$ denote the diagonal matrix generated by $\boldsymbol{s}$, the prediction loss can be expressed in a matrix form as $||(\boldsymbol{W}^T\boldsymbol{X}_i - \boldsymbol{y}_i\boldsymbol{\alpha}^T)\boldsymbol{S}||_{1,2}$. The $\|\cdot\|_{1,2}$ norm is used to measure the regression loss because it is robust to noise and outliers [30].

[2] We would add a small positive constant ϵ to $\boldsymbol{r}(j)$ when it is zero.

[3] Intuitively, we need the vector to satisfy the boundary constraint $\boldsymbol{\pi}(1) = 1$, $\boldsymbol{\pi}(N+1) = T$ and the monotonicity constraint $\boldsymbol{\pi}(t_1) \leq \boldsymbol{\pi}(t_2)$ for any $t_1 \leq t_2$.



(a) soft linear regression (SLR)

(b) soft RNN (SRNN) regression

Fig. 3. Illustration of the main difference between our soft linear regression (SLR) model and soft RNN (SRNN) model. In the SLR model, prediction is achieved based on the currently observed subsequences. While in the SRNN model, the dependency between all the sequentially arrived subsequences is explicitly explored by a temporally connected hidden layer. Hence, the output is obtained based on both the currently observing subsequence and historical subsequences. The objectives of both the SLR and SRNN models are to minimize the loss between outputs and the corresponding soft labels.

In the above formulation, we constrain $\boldsymbol{\alpha}(N) = \boldsymbol{\alpha}^T \boldsymbol{e}_N = 1$ in order to ensure that a strong label can be derived if the entire sequence is observed, where $\boldsymbol{e}_N$ is a binary vector with only the N^{th} entry being 1. In addition, we also restrict all entries in $\boldsymbol{\alpha}$ within $[0, 1]$.

In order to make sure the variation of soft label is smooth, we further impose a consistency constraint on $\boldsymbol{\alpha}$ as follows:

$$\min_{\boldsymbol{W},\boldsymbol{\alpha}} \sum_{i=1}^{I} \sum_{n=1}^{N} \underbrace{||(\boldsymbol{W}^T \boldsymbol{X}_i - \boldsymbol{y}_i \boldsymbol{\alpha}^T)\boldsymbol{S}||_{1,2}}_{Prediction\ loss\ term} + \frac{\xi_1}{2} \underbrace{||\nabla \boldsymbol{\alpha}||_2^2}_{Consistency\ term} \quad (5)$$

$$+ \frac{\xi_2}{2} \overbrace{||\boldsymbol{W}||_F^2}^{Regularization\ term}$$

$$s.t. \quad \boldsymbol{\alpha}^T \boldsymbol{e}_N = 1, 0 \leqslant \boldsymbol{\alpha} \leqslant 1, \xi_1, \xi_2 \geq 0. \quad (6)$$

Consistency term $||\nabla \boldsymbol{\alpha}||_2^2$. This constraint is to enforce the soft-label smoothness between subsequences at two consecutive progress levels for an action. We compute the gradient of soft labels and measure its norm in order to control the variations of soft labels between subsequences. The effect of the consistency term is controlled by ξ_1. As the gradient operator $\nabla \boldsymbol{\alpha}$ is a linear operator, the consistency term can be rewritten in a matrix form $\boldsymbol{G\alpha}$ equivalently, where we set $\boldsymbol{G}$ as $\begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \in \mathbb{R}^{(N-1)\times N}$. In this way, we can rewrite our soft linear regression (SLR) model as follows:

$$\min_{\boldsymbol{W},\boldsymbol{\alpha}} \sum_{i=1}^{I} ||(\boldsymbol{W}^T \boldsymbol{X}_i - \boldsymbol{y}_i \boldsymbol{\alpha}^T)\boldsymbol{S}||_{1,2} + \frac{\xi_1}{2} ||\boldsymbol{G\alpha}||_2^2 + \frac{\xi_2}{2} ||\boldsymbol{W}||_F^2$$

$$s.t. \quad \boldsymbol{\alpha}^T \boldsymbol{e}_N = 1, 0 \leqslant \boldsymbol{\alpha} \leqslant 1, \xi_1, \xi_2 \geq 0. \quad (7)$$

Model Optimization for SLR. We solve our soft linear regression (SLR) model (7) by a coordinate descent algorithm which optimizes over one parameter at each step while holding the others fixed. The objective function (7) of SLR can be monotonically decreased with a guaranteed convergence by iterating over the following two steps. At step 1, we optimize the predictor $\boldsymbol{W}$ with $\boldsymbol{\alpha}$ fixed. At step 2, we optimize it over $\boldsymbol{\alpha}$ with $\boldsymbol{W}$ fixed. Please refer to our conference version [15] for more details.

3.4 A Soft RNN (SRNN) Regression based Early Action Prediction

In real-time system, subsequences are observed sequentially, and the prediction on the previous subsequence could be useful for performing prediction on the subsequence at the next progress level. In this section, we further explore the relationship between the prediction of successive on-going subsequences by developing a soft RNN-based framework, which embeds learning soft labels into the RNN model. A graphical illustration of the difference between our SLR and SRNN can be found in Figure 3.

To detail our soft RNN-based framework, we would first re-formulate the soft linear regression (SLR) model (formula (7)). In the soft regression framework, each element in the soft label vector $\boldsymbol{\alpha}$ is associated with an subsequence at certain progress level. Our objective is to minimize the prediction loss (i.e. the gap between the output of score function for the t^{th} subsequence and $\boldsymbol{\alpha}(t)\boldsymbol{y}_i$). In general, the objective function of SLR model (7) can be re-formulated as

$$\min_{\boldsymbol{\theta},\boldsymbol{\alpha}} \sum_{i=1}^{I} L(f(\boldsymbol{\theta}, \boldsymbol{X}_i), \boldsymbol{y}_i \boldsymbol{\alpha}^T, \boldsymbol{S}) + \frac{\xi_1}{2} ||\boldsymbol{G\alpha}||_2^2 + \frac{\xi_2}{2} R(\boldsymbol{\theta}). \quad (8)$$

Here, we use $\boldsymbol{\theta}$ to denote the learnable model parameters. In the above formulation, $R(\boldsymbol{\theta})$ is a regularizer used to penalize the parameters with large values. $L(f(\boldsymbol{\theta}, \boldsymbol{X}_i), \boldsymbol{y}_i \boldsymbol{\alpha}^T, \boldsymbol{S})$ is a loss function used for guiding the predictor and soft labels learning, where $f(\boldsymbol{\theta}, \boldsymbol{X}_i)$ is a score function indicating the confidences of predicting the class label of each subsequence, whose LAFF features are indicated by $\boldsymbol{X}_i$. Taking the model SLR in formula (7) for example, $R(\boldsymbol{\theta})$ is is defined as $||\boldsymbol{W}||_F^2$ ($\boldsymbol{\theta}$ represents $\boldsymbol{W}^T$ in this case). The score function $f(\boldsymbol{\theta}, \boldsymbol{X}_i)$ is defined as the inner production between the model parameters and input feature (i.e., $\boldsymbol{\theta}^T \boldsymbol{X}_i$). $L(f(\boldsymbol{\theta}, \boldsymbol{X}_i), \boldsymbol{y}_i \boldsymbol{\alpha}^T, \boldsymbol{S})$ is defined as $||(\boldsymbol{W}^T \boldsymbol{X}_i - \boldsymbol{y}_i \boldsymbol{\alpha}^T)\boldsymbol{S}||_{1,2}$.

We extend SLR by replacing the linear score function with a deep recurrent neural network (RNN) architecture. Such an extension enables explicitly modeling the dependencies among the sequentially observed subsequences. Specifically, for speeding up the training of our deep regression model, here we set the prediction loss $L(f(\boldsymbol{\theta}, \boldsymbol{X}_i), \boldsymbol{y}_i \boldsymbol{\alpha}^T, \boldsymbol{S})$ as $||(f(\boldsymbol{\theta}, \boldsymbol{X}_i) - \boldsymbol{y}_i \boldsymbol{\alpha}^T)\boldsymbol{S}||_F^2$. The score function $f(\boldsymbol{\theta}, \boldsymbol{X}_i)$ is given by the outputs of recurrent neural network $[\mathbf{o}(1), \mathbf{o}(2), ..., \mathbf{o}(N)]$, which can be formulated as follows:

$$\begin{aligned}&\mathbf{h}(t) = ReLu(\mathbf{U}_{hh}^T\mathbf{h}(t-1) + \mathbf{U}_{xh}^T\mathbf{X}_i(:,t) + \mathbf{b}),\\ &\mathbf{o}(t) = \mathbf{U}_{ho}^T\mathbf{h}(t) + \mathbf{c}.\end{aligned} \tag{9}$$

where $\mathbf{U}_{hh}, \mathbf{U}_{xh}, \mathbf{U}_{ho}, \mathbf{b}$, and $\mathbf{c}$ form the model parameters $\boldsymbol{\theta}$, which would be learned in the training stage. $\mathbf{h}(t)$ is the hidden state of the t^{th} subsequence and its information will be transformed to the future subsequences through matrix $\mathbf{U}_{hh}$. Here we do not employ an additional softmax layer to normalize the output $\mathbf{o}$ into a probability vector as done in other RNN architectures [9], [46]. This is because the softmax layer is unsuitable for our soft regression model as the sum of the elements in the output of softmax operator is 1, while the sum of the elements in $\boldsymbol{\alpha}(t)\boldsymbol{y}_i$ is $\boldsymbol{\alpha}(t)$, whose value is within $[0, 1]$. We use the $ReLu$ operator as our activation function to non-linearly map the input features and the previous time-step hidden state into the hidden state space of the current time-step. In addition, we formulate the regularization term $R(\boldsymbol{\theta})$ as the square of L_2-norm (l_2-norm) of the model parameters $\boldsymbol{\theta}$ (i.e. $\mathbf{U}_{hh}, \mathbf{U}_{xh}, \mathbf{U}_{ho}, \mathbf{b}$, and $\mathbf{c}$) and it can serve as a weight decay term to penalize weights with large values.

Model Optimization for SRNN. Similar to the optimization for the soft linear regression (SLR) model, we also develop a two-step optimization algorithm to solve the SRNN model. The optimization is achieved by iterating over the following two steps.

- STEP 1. For fixed soft labels $\boldsymbol{\alpha}$, optimize the RNN parameters $\boldsymbol{\theta}$ (i.e., $\mathbf{U}_{hh}, \mathbf{U}_{xh}, \mathbf{U}_{ho}, \mathbf{b}$ and $\mathbf{c}$):

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{I} L(f(\boldsymbol{\theta}, \boldsymbol{X}_i), \boldsymbol{y}_i\boldsymbol{\alpha}^T, \boldsymbol{S}) + \frac{\xi_2}{2}R(\boldsymbol{\theta}), \tag{10}$$

This is a standard RNN optimization problem, and the parameters can be determined by a stochastic gradient descent (SGD) algorithm with momentum and back propagation through time (BPTT) [60].

- STEP 2. For fixed RNN parameters $\boldsymbol{\theta}$, optimize the soft labels $\boldsymbol{\alpha}$:

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{I} L(f(\boldsymbol{\theta}, \boldsymbol{X}_i), \boldsymbol{y}_i\boldsymbol{\alpha}^T, \boldsymbol{S}) + \frac{\xi_1}{2}||\boldsymbol{G}\boldsymbol{\alpha}||_2^2. \tag{11}$$

$$s.t. \quad \boldsymbol{\alpha}^T\boldsymbol{e}_N = 1, 0 \leqslant \boldsymbol{\alpha} \leqslant 1. \tag{12}$$

It is hard to directly solve the above problem because the existence of the bounded constraints (12). Here, we introduce a method to find an approximate solution based on the popularly used projected gradient descent technique. Firstly, we would solve the following optimization problem without any constraint:

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{I} L(f(\boldsymbol{\theta}, \boldsymbol{X}_i), \boldsymbol{y}_i\boldsymbol{\alpha}^T, \boldsymbol{S}) + \frac{\xi_1}{2}||\boldsymbol{G}\boldsymbol{\alpha}||_2^2. \tag{13}$$

The above unconstrained problem can be optimized using a gradient descent method. The updated point is then projected into the feasible solution space $\{\boldsymbol{\alpha} \in R^N \mid 0 \leq \boldsymbol{\alpha} \leq 1, \boldsymbol{\alpha}^T\boldsymbol{e}_N = 1\}$ to obtain a feasible solution.

3.5 Soft RNN Regression with Multi-soft Labels

Different actions could perform at different progress levels at a time even if they start at the same time. That means the soft labels should be different for different actions at a time. Hence, we further extend the proposed soft RNN regression model by explicitly learning a category-specific soft label vector for each action type. We call this extended model as Multiple Soft labels Recurrent Neural Network (MSRNN), which is formulated as

$$\begin{aligned}&\min_{\boldsymbol{\theta},\{\boldsymbol{\alpha}_c\},\overline{\boldsymbol{\alpha}}} \underbrace{\sum_{c=1}^{C}\sum_{i=1}^{I_c}\Delta(c,\boldsymbol{y}_i)L(f(\boldsymbol{\theta},\boldsymbol{X}_i),\boldsymbol{y}_i\boldsymbol{\alpha}_c^T,\boldsymbol{S})}_{Prediction\ loss\ term} + \underbrace{\frac{\xi_1}{2}\sum_{c=1}^{C}||\boldsymbol{G}\boldsymbol{\alpha}_c||_2^2}_{Consistency\ term}\\ &+ \overbrace{\frac{\xi_2}{2}R(\boldsymbol{\theta})}^{regularization\ term} + \overbrace{\frac{\xi_3}{2}\sum_{c=1}^{C}||\boldsymbol{\alpha}_c-\overline{\boldsymbol{\alpha}}||_F^2}^{Between-class\ consistency\ loss\ term}.\end{aligned} \tag{14}$$

Here, $\boldsymbol{\alpha}_c$ is the soft labels of class c, I_c indicates the number of training samples for action class c, $\Delta(c, \boldsymbol{y}_i)$ is a class consistency term, whose value is 1 if the action class of the i^{th} sample is c; otherwise, it is 0. Compared with the SRNN model described in the last section, MSRNN would learn a category-specific soft label vector for each action class along with minimizing the between-class consistency loss $||\boldsymbol{\alpha}_c - \overline{\boldsymbol{\alpha}}||_F^2$. In the between-class consistency loss, we expect that the soft labels learned for each action class can share some variation tendencies by making them approach a common soft label vector $\overline{\boldsymbol{\alpha}}$. This is reasonable in practice, as for most of the actions, the more about the actions are observed, the more confident the system generally becomes for early action prediction [17], [23]. In general, the soft labels for each action overall increase over time, although it could be observed that sometimes the actor has completed the action execution in the last two snapshots, which is not informative for prediction and thus becomes redundant as shown in Figures 6 and 7, which would be discussed elaborately in Section 4.3.

Model Optimization for MSRNN. Similar to our optimization for the SLR and SRNN models, we use a coordinate descent method to solve MSRNN (14) and obtain a set of optimal parameters $\{\boldsymbol{\theta}, \boldsymbol{\alpha}_c, \overline{\boldsymbol{\alpha}}\}_{c=1,2,...,C}$ by iterating the following three steps.

- STEP 1. We optimize the RNN parameters $\boldsymbol{\theta}$ with the soft labels $\{\boldsymbol{\alpha}_c\}_{c=1,2,...,C}$ and $\overline{\boldsymbol{\alpha}}$ fixed. The updating of $\boldsymbol{\theta}$ is identical to that in the optimization of SRNN problem (10).

- STEP 2. We optimize the class-specific soft labels $\{\boldsymbol{\alpha}_c\}_{c=1,2,...,C}$ with the others fixed. Specifically, we optimize the following problem:

$$\begin{aligned}&\min_{\{\boldsymbol{\alpha}_c\}} \sum_{c=1}^{C}\sum_{i=1}^{I_c}\Delta(c,\boldsymbol{y}_i)L(f(\boldsymbol{\theta},\boldsymbol{X}_i),\boldsymbol{y}_i\boldsymbol{\alpha}_c^T,\boldsymbol{S}) + \frac{\xi_1}{2}\sum_{c=1}^{C}||\boldsymbol{G}\boldsymbol{\alpha}_c||_2^2\\ &+ \frac{\xi_3}{2}\sum_{c=1}^{C}||\boldsymbol{\alpha}_c-\overline{\boldsymbol{\alpha}}||_F^2.\\ &s.t. \quad \boldsymbol{\alpha}_c^T\boldsymbol{e}_N = 1, 0 \leqslant \boldsymbol{\alpha}_c \leqslant 1, c = 1, 2, ..., C.\end{aligned}$$

The above problem can be decomposed into C independent soft label learning sub-problems:

$$\begin{aligned}&\min_{\boldsymbol{\alpha}_c} \Delta(c,\boldsymbol{y}_i)L(f(\boldsymbol{\theta},\boldsymbol{X}_i),\boldsymbol{y}_i\boldsymbol{\alpha}_c^T,\boldsymbol{S}) + \frac{\xi_1}{2}||\boldsymbol{G}\boldsymbol{\alpha}_c||_2^2\\ &+ \frac{\xi_3}{2}||\boldsymbol{\alpha}_c-\overline{\boldsymbol{\alpha}}||_F^2.\\ &s.t. \quad \boldsymbol{\alpha}_c^T\boldsymbol{e}_N = 1, 0 \leqslant \boldsymbol{\alpha}_c \leqslant 1.\end{aligned}$$

This is a standard single soft label learning problem and can be solved by the **STEP 2** algorithm described in Section 3.4.

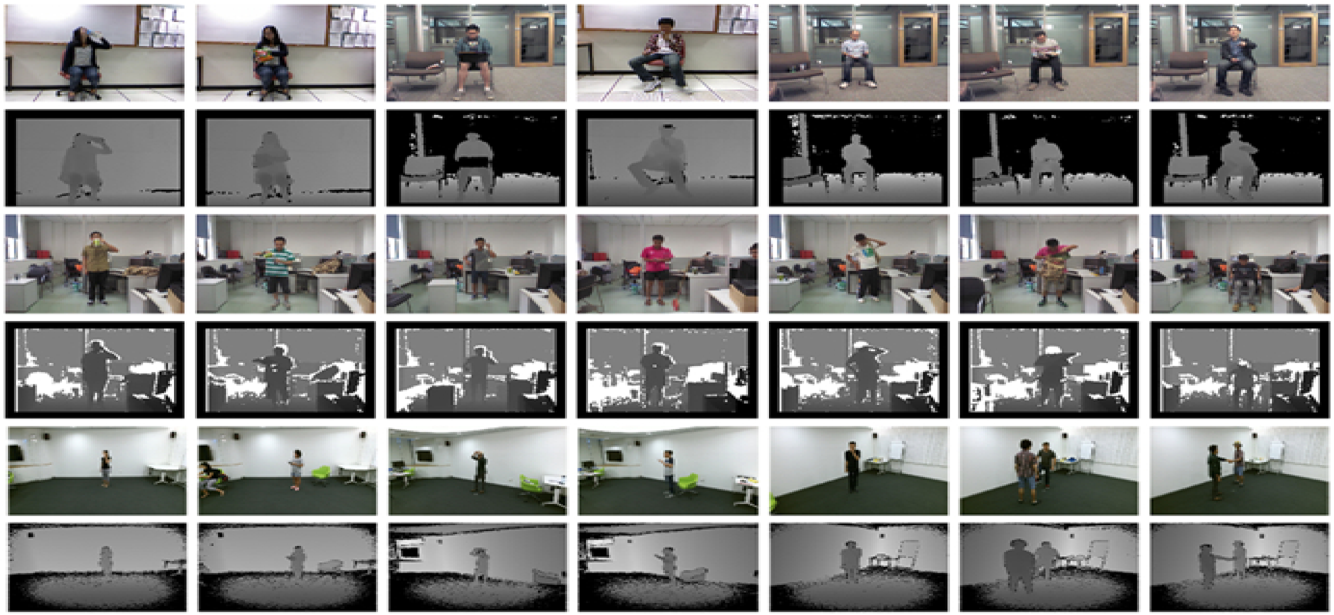


Fig. 4. Some examples from the ORGBD, SYSU 3DHOI and NTU Large Scale datasets. The first, third and fifth row present RGB snapshots from ORGBD, SYSU 3DHOI and NTU Large Scale set, respectively. The second, forth and sixth row present the corresponding depth images.

Algorithm 1 Optimization for MSRNN. The term $objUpdate$ indicates the value variation of the objective function of Formula (14).

```
Require:
    Input: $\mathbf{X}_i, \mathbf{y}_i, \xi_1, \xi_2, \xi_3$;
    Initialization: the RNN parameters $\boldsymbol{\theta}$ are random matrices, $\boldsymbol{\alpha}$ is a vector with $N$ elements monotonically increasing from 0.25 to 1, $IterOut = 1, maxIter = 400$;
Ensure:
while $objUpdate \geq thr$ and $IterOut < maxIter$ do
    Update RNN parameters $\boldsymbol{\theta}$ using SGD algorithms;
    Update soft label vectors $\boldsymbol{\alpha}_c$ using gradient descent method;
    $\overline{\boldsymbol{\alpha}} \leftarrow \frac{1}{C}\sum_{c=1}^{C}\boldsymbol{\alpha}_c$; $IterOut$++;
end while
return $\boldsymbol{\theta}, \boldsymbol{\alpha}$
```

- STEP 3. Finally, we update the common soft label vector $\overline{\boldsymbol{\alpha}}$ by solving the following minimization problem:

$$\min_{\overline{\boldsymbol{\alpha}}} \frac{\xi_3}{2}\sum_{c=1}^{C}||\boldsymbol{\alpha}_c - \overline{\boldsymbol{\alpha}}||_F^2.$$

The above problem can be solved by generally setting $\overline{\boldsymbol{\alpha}}$ as $\frac{1}{C}\sum_{c=1}^{C}\boldsymbol{\alpha}_c$. We summarize the overall procedure for the optimization of MSRNN model in Algorithm 1.

3.6 Early Prediction

Given a probe ongoing action subsequence where the progress level is unknown, our soft regression models (SLR, SRNN and MSRNN) output the prediction scores using the corresponding LAFF features $\{\boldsymbol{x}(t)\}_{t=1,2,...,}$. Then the prediction was made by finding the class label that has the maximum score. Our methods can predict the labels of ongoing actions without knowing their progress levels in testing.

4 Experiments

We mainly evaluated our methods on three benchmark 3D action datasets: *Online RGB-D Action* dataset [67], *SYSU 3D HOI* dataset [14], and NTU Large Scale dataset [42]. In the following, we first briefly introduce the compared methods and implementation details, and then describe the experimental results.

4.1 Compared Methods & Implementation

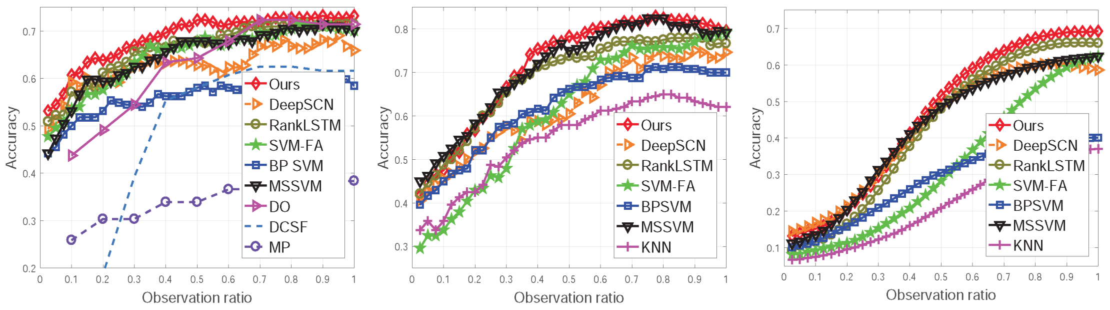
4.1.1 Compared Methods

We have implemented the following approaches using the same proposed features (i.e. LAFF) for comparison:

SVM on the Complete Activities (SVM-FA). In order to see how well our early action prediction model approximates the action recognition results when whole sequence but not partial of it is observed, we trained a generic action classifier (SVM) on the completely executed action sequences. During the testing phase, all the ongoing subsequences were predicted using the learnt action classifier.

Brute-force Prediction using SVM (BPSVM). It learns an action predictor from all the available subsequences. In this baseline, we assigned the label of each subsequence with the label of its full sequence (i.e., hard labeling). That means these labels were not soft labels as described in our methods. This baseline is introduced in order to show the benefits of using soft labels. We denote it as "BPSVM".

Multiple Stages SVM (MSSVM). We trained a SVM predictor on the sequences obtained at each progress level separately. While in the testing phase, we followed the same assumption in [23] that the progress level of ongoing action is known and thus we can directly make the prediction using the predictor specifically trained for that progress level. Although practical system is hard to have a chance to obtain the progress level of ongoing action sequence until the action has been completely executed, it still serves as a good reference for benchmarking early action prediction models. We denote this baseline as "MSSVM".



(a) ORGBD set (b) SYSU 3D HOI set (c) NTU Large Scale set

Fig. 5. Comparison results on the ORGBD (a), SYSU 3DHOI (b), and NTU Large Scale (c) sets. “MP”, “DCSF”, and “DO” denote “Moving Pose [10], [67]”, “DSTIP+DCSF [61], [67]”, and “Discriminative Order-let [67]”, respectively. Best viewed in color.

DeepSCN and RankLSTM. We also compared our soft regression model with the DeepSCN [25] and RankLSTM [37] models. The source codes for DeepSCN and RankLSTM are not available for benchmarking, we re-implemented them strictly by following the descriptions in [25], [37]. For a fair comparison, we fed our LAFF features into their learning frameworks and reported the best results among a large range parameter settings.

Other early action prediction methods. In addition to the above comparison, we also reported the results of using KNN classifier to predict actions, where K is set to 5. We further compared our method with the state-of-the-art early action prediction systems developed for RGB-D sequences [67] and RGB videos [2], [41].

The baselines “BPSVM”, “MSSVM” and “SVM-FA” were implemented by ourselves and tested with different SVM parameters. Our main experimental comparison was conducted based on the features introduced in Sec. 3.2. While using CNN features as the basic features would further improve the early prediction performance, using CNN features will largely slow down the speed of early prediction, from real time to non-real time. We will discuss this issue in Section 4.6.

4.1.2 Implementation

For our LSR model, we set the regularization parameters ξ_1 and ξ_2 as 5000 and 1 throughout all the experiments, respectively. While for the SRNN and MSRNN models, we set the parameters ξ_1, ξ_2 and ξ_3 as 500, 0.005 and 0.1, respectively. Note that parameter ξ_2 also serves as a decay rate in the optimization of our RNN based models. The maximum progress level N was set as 40. The weight $s(n)$ can be understood as a prior weighting on the regression loss of the n^{th} subsequence in Eq. (6). In general the loss of the subsequence at the end of the sequence is more important as the type of action becomes more clear. Hence, we increased $s(n)$ in Eq. (6) from 0.25 to 1 uniformly. Its influence will be studied in Section 4.5. We employed the stochastic gradient descent with momentum approach [50] to optimize the proposed SRNN and MSRNN, where the momentum factor was set as 0.9. To further speed up our optimization, we used PCA to reduce the dimension of the extracted LAFF features, where 98% of variance is retained.

4.1.3 Evaluation Criteria

For quantitative comparison, we present the accuracies of an early action prediction system for predicting actions at different progress levels. These accuracies can indicate how well the early action prediction system predicts early actions. We also plot the accuracies against observation ratios as a curve (See Figure 5 for example) and the area under the curve (*AUC*) is computed to measure the overall performance of early action prediction system. In addition, the prediction speed is also important for some real-applications. We also report the speeds of the prediction systems.

TABLE 1
Prediction (%) on ORGBD set. “MP”, “DCSF”, and “DO” denote “Moving Pose [10], [67]”, “DSTIP+DCSF [61], [67]”, “Discriminative Order-let [67]”, respectively. The last row is for AUC.

Observation ratio	MSSVM	BPSVM	SVM-FA	MP	DCSF	DO	RankLSTM	DeepSCN	Ours
10%	53.1	54.5	53.6	25.9	14.3	43.8	57.1	59.2	**60.7**
60%	67.4	64.7	67.4	33.9	55.4	63.4	69.2	62.3	**71.4**
80%	70.1	66.1	70.1	38.4	61.6	71.4	70.5	66.8	**73.2**
100%	70.1	66.1	70.1	38.4	61.6	71.4	71.4	65.9	**73.2**
AUC	64.7	61.8	64.9	34.3	49.5	63.0	66.1	62.8	**68.7**

4.2 Results on Online RGB-D Action Datasets

The Online RGB-D Action Dataset (ORGBD) was collected for online action recognition and early action prediction [67]. Some action examples are shown in Figure 4. For evaluation, we used the same-environment evaluation protocol detailed in [67], where half of the subjects were used for training a predictor and the rest were used for testing. In this setting, there are totally 224 RGB-D sequences of sixteen subjects, including seven human-object interaction activities (*drinking*, *eating*, *using laptop*, *reading cell phone*, *making phone call*, *reading book* and *using remote*). The accuracies are computed as an average over a two-fold validation.

We compared the proposed MSRNN with the baselines and other related prediction methods as described in Section 4.1. The results are presented in Figure 5(a) and Table 1. As shown, our method can produce better prediction results at most of the observation ratios than the competitors. We also find that the performance gap became larger if fewer action frames were observed. This is as expected because our soft regression model explicitly makes use of the subsequences that contain partial action executions for obtaining a reliable predictor. By carefully examining the comparisons of our method MSRNN, the baselines BPSVM, and SVM-FA, we can find that the introduction of the soft-label learning mechanism can significantly improve prediction performance. It also worked better than the MSSVM model, which predicts ongoing activities with known progress level using multiple pre-trained predictors. We also observe that our soft regression model performed better than DeepSCN [25]

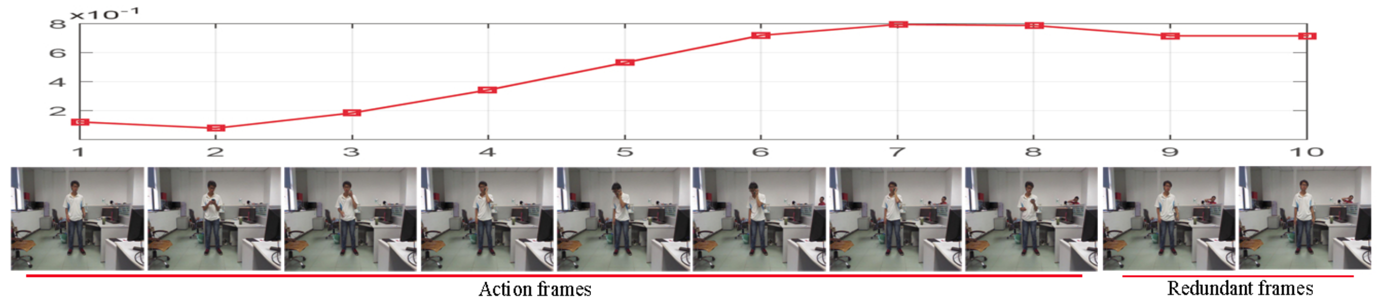


Fig. 6. An example from the SYSU dataset with prediction scores. We uniformly selected 10 snapshots from a sequence of *calling with a phone* for this illustration. The top row of the figure shows the confidences of our model correctly predicting the underlying subsequences observed until the time of each respected snapshot. It could be observed that the actor has completed the action execution in the last two snapshots, which is not helpful for prediction and thus becomes redundant.

and RankLSTM [37] with the same inputs, which demonstrates the effectiveness of our model for predicting early actions.

In addition, we also compared our method with the state-of-the-art prediction algorithms on this set [61], [67]. The proposed method outperformed the state-of-the-art model (discriminative order-let model) [67] by a large margin (more than 10 percent) when only 10% of the sequence were observed, which clearly indicates that our system can perform early prediction of action more accurately. It could also be observed that when full sequences were provided, our predictor still performed better and obtained an accuracy of 73.2%, which is 1.8% higher than the discriminative order-let model. This is probably because that the long-duration motion information, ignored by the frame-level prediction model [67], is very important for identifying human activities, especially at early action stages where the observed action evidence (such as human pose and object appearance etc.) is not sufficient for accurate early action prediction.

4.3 Results on SYSU 3D Human-Object Interaction Set

The SYSU 3D Human-Object Interaction Set (SYSU 3D HOI Set) [4] consists of 12 different activities from 40 participants, including *playing with a cell-phone*, *calling with a cell-phone*, *mopping* and *sweeping* etc. Some snapshots of the activities are presented in Figure 4. In this dataset, each action involves a type of human-object interaction, and the participant's motions and the objects he/she manipulated are similar among some activities [13]. For evaluation, we employed the cross-subject setting popularly used in RGB-D action recognition. In particular, we trained our predictor and all compared methods using the samples performed by the first 20 subjects and then tested on the rest subjects.

The prediction comparison results are presented in Table 2 and Figure 5(b). As shown, our predictor can obtain a good performance at most of the progress levels and it significantly outperformed BPSVM and SVM-FA. We observed that, when additionally using the progress levels in prediction, the model MSSVM performs slightly better than our MSRNN at early stages, but both performed comparably after that. And our methods perform better when sufficient action sequences are observed (e.g., the whole action sequences are used for prediction). We also observe that our method and most of the competitors obtain the best prediction accuracy when 80% of the action sequences are observed. The performance began to drop if more frames were used, and we found that it is because some videos clipped in this set contain some redundant frames as illustrated in Figure 6.

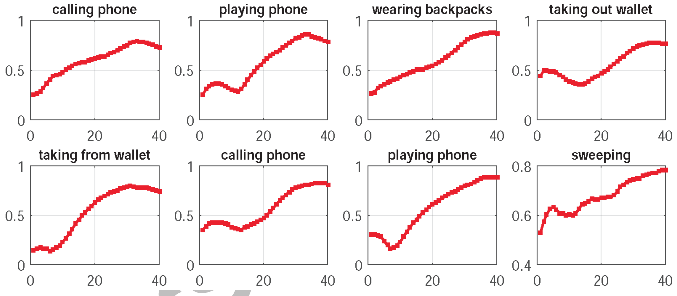


Fig. 7. Prediction scores for eight specific samples from different action classes. The vertical axis indicates prediction scores and the horizontal axis is the progress level of subsequence.

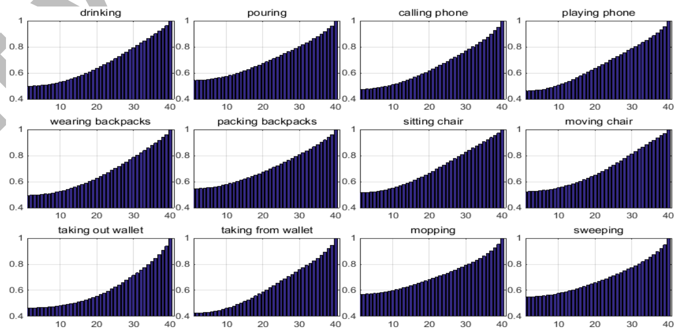


Fig. 8. Example soft labels learned on the SYSU 3D HOI set. The vertical axis indicates the values for the soft labels and the horizontal axis is the progress level of subsequence.

We also plot the prediction scores (i.e., the values of $\mathbf{o}(t+1)$) in Eq. (9) obtained for some specific action samples in Figure 7. It can be seen that the prediction confidences in general, but not always, increase with the observation ratio. This is as expected because more action observation does not always mean more discriminative. Ongoing observations sometimes could confuse the system for some specific action samples. However, from a global view, more action observations are in general beneficial for early action prediction.

The values of the learned soft labels are shown in Figure 8. It could be observed that the soft labels start from different positive values for different activities. This is as expected because different activities contain contextual cues of different strength at their early stages, for instance, *mopping* contains stronger action context (e.g., shapes and textures of objects) than *taking out wallet*. In general, the values of soft labels increase as when more sequences of actions are observed for prediction.

[4] It can be downloaded from http://isee.sysu.edu.cn/~hujianfang/

TABLE 2
Prediction (%) on the SYSU 3D HOI set. The last row is for AUC.

Observation ratio	KNN	MSSVM	BPSVM	SVM-FA	RankLSTM	DeepSCN	Ours
10%	35.8	**50.8**	45.0	33.8	48.7	45.5	47.5
60%	61.3	78.8	68.3	72.9	75.4	67.2	**80.4**
80%	65.0	**82.5**	70.8	75.8	77.5	73.8	**82.5**
100%	62.1	79.2	70.0	79.2	76.6	74.7	**79.6**
AUC	54.7	71.1	61.9	61.0	68.5	62.2	**71.6**

TABLE 3
Prediction (%) on the NTU Large Scale set. The last row is for AUC.

Observation ratio	KNN	MSSVM	BPSVM	SVM-FA	RankLSTM	DeepSCN	Ours
10%	7.5	13.5	11.7	9.2	11.5	16.8	**15.2**
60%	26.0	53.5	34.0	36.8	55.9	54.6	**59.1**
80%	34.5	59.6	39.1	53.5	64.4	60.1	**67.4**
100%	37.0	62.4	40.1	62.4	65.9	58.6	**69.2**
AUC	21.9	43.0	28.4	32.2	43.1	43.2	**46.6**

4.4 Results on the NTU Large Scale dataset

This dataset was collected for the research of large scale RGB-D action recognition. It contains a total of 56,880 RGB-D video clips involving 60 action classes. This set is very challenging as most of the considered activities involve complex interactions, such as human-object interactions (e.g. drinking and eating etc.) and human-human interactions (e.g. hugging etc). All the activities were performed by 40 subjects and captured from different view points using several Kinect v2 cameras. For evaluation, we followed the cross-subject setting described in [34], [42] and used the samples performed by 20 certain subjects for the model training and the rest for testing. In total, we have 40,320 training samples and 16,560 testing samples.

We present our comparison results in Figure 5(c) and Table 3. As shown, our MSRNN model obtained the best prediction performance and outperformed the competitors SVM-FA, BPSVM and KNN by a large margin (more than 14% in terms of AUC). We can observe that additionally using the progress level of an action to train predictors is beneficial. So, MSSVM performed comparably to our MSRNN model at the early stages (≤ 0.5), but performed significantly worse after that. Especially, when complete action executions were observed, our model can obtain a recognition accuracy of 69.24%, which is comparable with the state-of-the-art recognition result (69.2%) reported in [34], where a complex LSTM architecture with trust gate, forget gate and input gate was developed. This implies that our proposed soft regression framework is also beneficial for the task of action recognition and can obtain the state-of-the-art recognition result. As expected, our soft regression model outperformed the RankLSTM [37] and DeepSCN [25] approaches again, which demonstrates the efficacy of our soft label learning framework for early action prediction. We also note that the prediction results of most methods on the first 10% frames on this set is much lower than that on the ORGBD and SYSU 3D HOI sets. This is because that the NTU Large Scale set is much more challenging with larger scales and action class diversity, including body actions, gestures, human-object interactions, and even human-human interactions etc. Moreover, some actions in this set contain similar appearance information (e.g., wearing on glasses vs. taking off glasses), which plays an important role for the early action prediction, especially when the actions at early stages do not include enough motion cues for characterizing actions. Please refer to our supplementary material for more details and experimental analysis.

4.5 More Evaluations

We further evaluated the performances of our developed early action prediction system. Since evaluating our methods on the NTU large scale dataset is quite time-consuming and it takes about 4-5 days for a single round of optimization, in this section, we mainly conducted experiments on the SYSU 3DHOI set with cross subject setting, which is quite larger than the ORGBD dataset.

TABLE 4
Comparison of our RGB-D early action prediction methods and the conventional RGB video based early action prediction approaches on the SYSU 3D HOI set. SLR (RGB) and MSRNN (RGB) indicate the SLR and MSRNN models tested on the RGB channel, respectively.

Observation ratio	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	AUC
DBOW [41]	31.7	40.0	43.8	46.7	52.1	54.2	58.8	59.6	62.1	62.5	49.8
SC [2]	30.4	41.3	50.8	53.3	57.1	57.9	57.9	58.8	60.4	61.3	51.5
MSSC [2]	30.4	40.8	47.1	55.0	56.7	59.6	57.5	60.8	62.1	62.9	51.9
SLR(RGB)	38.3	45.8	52.9	60.8	63.3	67.5	69.6	70.4	70.4	70.8	59.5
SLR	45.8	55.0	64.6	71.3	73.8	76.3	80.8	81.3	80.8	**80.0**	69.6
MSRNN(RGB)	37.9	46.3	57.9	62.1	67.5	70.4	71.7	71.3	70.8	71.3	62.7
MSRNN	**47.5**	**56.7**	**66.7**	**75.4**	**78.3**	**80.4**	**81.7**	**82.5**	**81.7**	79.6	**71.6**

RGB-D prediction vs. RGB. Intuitively, the RGB-D early action prediction can be casted as a RGB early action prediction problem by discarding the depth and skeleton modalities and then RGB early action prediction methods can be easily implemented. Here, we tabulate the results on the SYSU 3DHOI set obtained by methods (Dynamic BOW [41], SC, and MSSC [2]) developed in [2][5] as well as our method in Table 4. As shown, our soft regression models (both SLR and MSRNN) have a significant advantage over these methods even using the same input data (RGB data). From the results, we can conclude that a RGB-D based early action prediction system has its unique benefit.

Evaluation on the elements used in the RGB-D. Results in Table 5 show that the predictor (using MSRNN model) learned from the combination of RGB, depth and skeleton channels is better than only using one of them. This is reasonable because RGB, depth and skeleton sequences indeed characterize activities from different aspects, and any single channel is intrinsically limited in overcoming the inherent visual ambiguity caused by human (object) appearance changes, cluttered background, view variation, and occlusions etc.

Benefits of learning soft labels α. For comparison, we imple-

TABLE 5
Evaluation on the elements in RGB-D early action prediction on SYSU 3D HOI set. We use RGB-D to indicate that the predictor is learned from all the available data for prediction (i.e. RGB, DEP and SKL).

Observation ratio	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	AUC
RGB	37.9	46.3	57.9	62.1	67.5	70.4	71.7	71.3	70.8	71.3	61.3
DEP	32.5	44.2	53.8	62.9	67.9	71.3	72.9	72.5	72.5	70.4	60.6
SKL	30.8	36.3	44.2	45.8	53.3	56.7	58.7	63.3	65.4	65.8	50.7
RGB-D	**47.5**	**56.7**	**66.7**	**75.4**	**78.3**	**80.4**	**81.7**	**82.5**	**81.7**	**79.6**	**71.6**

[5] The codes are downloaded from http://www.visioncao.com/index.html.

TABLE 6
Evaluations of different soft regression models on SYSU 3D HOI set (%).

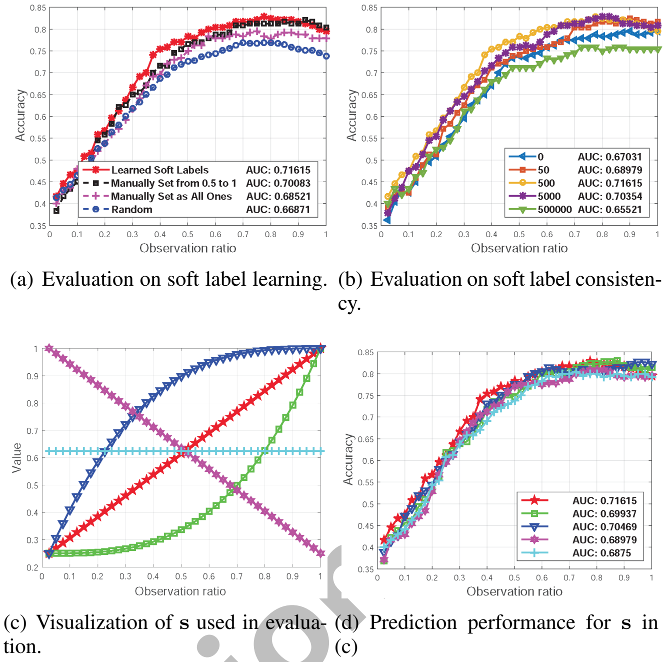
Observation ratio	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	AUC
SLR	45.8	55.0	64.6	71.3	73.8	76.3	80.8	81.3	80.8	**80.0**	69.6
MSLR	45.4	56.7	65.4	74.6	75.8	76.7	79.6	80.4	80.8	**80.0**	70.1
SRNN	44.2	55.8	65.4	74.6	77.5	78.8	79.2	80.4	81.7	79.6	70.2
MSRNN	**47.5**	**56.7**	**66.7**	**75.4**	**78.3**	**80.4**	**81.7**	**82.5**	**81.7**	79.6	**71.6**

mented two baselines based on the formulated MSRNN model, where different strategies were employed to manually determine the soft labels: 1) all the elements of $\boldsymbol{\alpha}$ were set as 1; 2) $\boldsymbol{\alpha}$ was set as a vector whose elements uniformly increase from 0.5 to 1[6]; and 3) we randomly generated a set of soft labels for our regression model with 20 replicates. As shown in Figure 9(a), the prediction accuracy would decrease if the soft labels were simply defined as the label of the whole sequence or randomly generated, which justifies that learning soft labels can benefit our predictor learning. As expected, using the randomly generated soft labels for prediction obtains the worst performance. We also observed that using uniformly increasing soft labels can also obtain an acceptable prediction result in our experiments, which illustrates that sequences with more action executions often contain more action cues for prediction.

The influence of consistency constraint. We studied the influence of the parameter ξ_1, which is employed to control the effect of the consistency term in our MSRNN model (Formula (14)). Figure 9(b) shows the performances of setting ξ_1 as 0, 50, 500, 5000 and 500000, respectively. As shown, our method can obtain a promising result with $\xi_1 = 500$. In general, a small or large ξ_1 would lead to a lower AUC. Especially, when ξ_1 is larger than a certain number (e.g. 500000), the learned soft label is unhelpful as all of its entries will be the same and thus lower prediction performances were observed.

Impact of the $\boldsymbol{s}$. In our soft regression model (6) and (14), a vector $\boldsymbol{s}$ is introduced to control the contribution of the regression losses caused by the subsequences of different progress levels. Here, we tested its influence. In the evaluation, we considered five different settings for $\boldsymbol{s}$, which were shown in Figure 9(c). More specifically, in the settings colored in 'blue', 'green', and 'red', the values for $\boldsymbol{s}$ increase from 0.25 to 1 based on different curves. For the settings 'magenta', $\boldsymbol{s}$ decreases from 1 to 0.25. And for the setting 'cyan', the $\boldsymbol{s}$ remains unchanged. The prediction performance obtained by each setting was presented in Figure 9(d) with the same color. As shown, the performance with an $\boldsymbol{s}$ with incremental values is better than the constant or even the diminishing ones. This is because the subsequences at the latter progress levels should contain more action execution and mis-predicting them should lead to a relatively large loss.

Comparisons of SLR, MSLR, SRNN, and MSRNN. We have developed different soft regression models to learn a reliable RGB-D action predictor, i.e., linear regression models with a single soft label vector (SLR) or multi-soft labels (MSLR)[7], RNN regression with a single soft label vector (SRNN) or multi-soft labels (MSRNN). Here, their performances are reported in Table 6. As shown, the MSRNN model produced the best prediction results on the SYSU 3DHOI set. And the performance gaps between MSRNN and other soft regression models are clear, especially with the observation ratios are smaller than 60%. It could also be observed that all of our soft regression models produced the best results when 80% $\sim$ 90% of the actions were observed. This is because some videos clipped in this set contain some redundant frames as illustrated in Figure 6. We can also observe that the models with multi-soft labels (MSLR and MSRNN) outperformed the models with a single soft label vector (SLR and SRNN), which means that learning a category-specific soft label vector for each action type is beneficial for our early prediction. Similar results can be especially observed on the much larger NTU dataset in the supplementary.



(a) Evaluation on soft label learning.

(b) Evaluation on soft label consistency.

(c) Visualization of $\mathbf{s}$ used in evaluation.

(d) Prediction performance for $\mathbf{s}$ in (c)

Fig. 9. More evaluations on the system performance. "AUC" indicates the area under all the presented observation ratios. Best viewed in color.

The convergence of the soft regression models. Our method converged to a minimum after a limited number of iterations. We empirically observed that 400 iterations were sufficient for obtaining a reliable solution for both SLR and MSRNN models in our experiments. One convergence example is shown in Figure 10. Excluding the time for computing the LAFF features, it took about 2 hours for our MSRNN (implemented in MATLAB on a machine with i3-2130 CPU and 16G memory) to obtain a reliable solution on the SYSU 3D HOI dataset with 240 training samples.

The speed of prediction. We report the average speed (in fps) of the developed early action prediction systems in Table 7. As shown, our prediction system can identify an action of ongoing RGB-D sequences in real time. Especially, our SLR system processed more than 40 frames per second using MATLAB on a normal desktop PC (CPU i5-4570), which is about 15 fps faster than the-state-of-the-art early prediction system developed in [67]. Our MSRNN model utilizes a more flexible non-linear predictor and thus it predicts activities slightly slower. The prediction speed of MSRNN is about 34 fps, which means that our MSRNN model

[6] We observe that the learned soft labels in general start from around 0.5. Thus, here we set the start of the soft labels as 0.5. We also tested the case of increasing soft-labels monotonically from 0 to 1 with a step of 1/N, and results were worse.

[7] The MSLR model is formulated by learning a set of class-specific soft labels for the SLR model, in a similar way to that of MSRNN.

TABLE 7
The comparison of prediction speed. The speed reported in the table includes the time of feature computation.

Method	SC [2]	MSSC [2]	Orderlet [67]	SLR [15]	MSRNN
Speed(fps)	< 0.02	< 0.0024	25	40	34

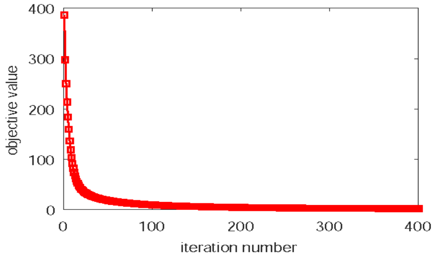


Fig. 10. Illustration of the convergence of our MSRNN model.

can still predict activities in real-time with better accuracy. We can also observe that the reported speeds of RGB video based prediction methods (SC [2] and MSSC [2]) are significantly slower than our early action prediction system.

4.6 Complement to the CNN features

In the previous sections, we have developed our prediction system based on an accumulative feature called "LAFF", which is computed from hand-crafted HOG features for the purpose of fast (real-time) prediction. Here, we experimentally demonstrate that our soft regression models can also be used to process the CNN features learned from RGB-D videos. In the implementation, we calculated our accumulative features based on the CNN descriptors extracted from RGB, depth and optical flow streams and fed these features into our MSRNN model. All the CNN features were computed using a TITAN X GPU. We tested the new system on the SYSU 3D HOI set. For computing CNN features from RGB and optical flow frames, we finetuned the two-stream VGG networks [47] pre-trained on the UCF101 set [49]. For calculating CNN features for depth stream, we trained a VGG network [48] from depth frames. As expected, the new system can obtain an AUC of 75.4%, which outperforms the hand-crafted HOG features based system by a margin of 3.8%. However, the prediction speed of using CNN features is about 1.5 FPS, which is significantly slower than using HOG features.

By directly using the two-stream action recognition model [47] for early action prediction (for both feature extraction and prediction), an AUC of 64.9% is achieved, which is much lower than our system. When feeding the two-stream CNN features into our soft regression model, we can obtain an AUC of 66.2%, which is much higher than our system with RGB inputs (62.7%), but much lower than that with RGB-D inputs (71.6%). Note that all of those results are obtained using our MSRNN but with different features, which demonstrates that our soft regression framework is flexible and can work with different inputs.

4.7 Soft regression for predicting unconstrained RGB actions

Here, we further illustrated that our soft regression model can also be used to predict actions from unconstrained RGB videos. To directly compare to other early action prediction models [2], [3], [23], [24], [25], [41] on RGB videos only, we tested our model on the UCF101 set [49], which contains 13320 unconstrained RGB videos from 101 action classes. For evaluation, we followed the same experimental settings as in [24], [25]. And we used the first 15 groups of videos for training, the next 3 groups for validation, and the rest for testing (note that those groups were pre-partitioned in [25]). It is worth noting that the body parts of many actions are only partially observable (e.g., action "apply eye makeup" and "apply lipstick"). Therefore, we constructed our "LAFF" features based on the two-stream CNN[8] features extracted for each individual frame. Finally, the calculated "LAFF" features are fed into our soft regression framework for prediction. We compared our system with several state-of-the-art RGB action prediction models, including Mem-LSTM [24], RGBBiLSTM [24], RGBLSTM [24], DeepSCN [25], MSDA [3], DBOW [41], IBOW [41], MSSC [2], MTSSVM [23]. Figure 11 presents the comparison results. As shown, our system obtained clearly better prediction performances over the state-of-the-art approaches. In particular, the best result among the existing is achieved by Mem-LSTM [24], with an AUC of 84.1%, which is about 3% lower than our method (87.3%). For the prediction of actions at early stages (e.g., 10% observation ratio), our approach can achieve an accuracy of 68%, which outperformed other competitors by a large margin ($\geq$17%). This validates the effectiveness of our approach for predicting actions captured in unconstrained scenario.

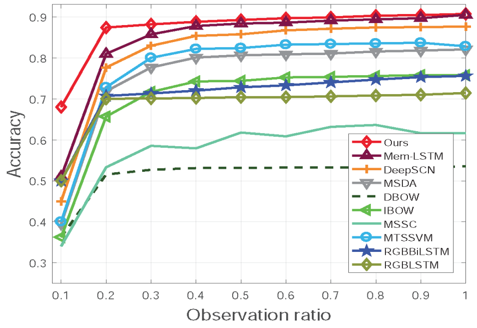


Fig. 11. Comparison results on the UCF101 set.

5 Conclusions

In this paper, we have developed a real-time RGB-D early action prediction system to identify ongoing actions under a soft regression framework. In the framework, we learn soft labels for regression on subsequences that contain partial action executions, so that it is not necessary to assume that the progress level of each subsequence is given at the testing stage. We learn both the soft labels and predictor jointly from linear to deep models, and finally a Multiple Soft labels Recurrent Neural Network (MSRNN) that takes the relation between subsequence and the discrepancy of soft labels over different classes into consideration is developed. In addition, a new RGB-D sequence feature called "local accumulative frame feature (LAFF)", which can be computed efficiently by constructing an integral feature map, is designed to characterize action contexts. We have demonstrated the efficacy of our approach on RGB-D and RGB early action prediction and show that soft labelling and depth information are important for achieving more robust early action prediction performance. In the future, we would consider integrating the early action prediction with action localization [45], [57] together to form a more complete early action analysis system.

[8] We followed the settings in [24] and trained two ResNets on the UCF101 dataset for the RGB and optical flow stream, respectively. The employed ResNet for RGB stream was pre-trained on ImageNet.

ACKNOWLEDGMENT

We would like to thank Dr. Yu Kong et al. for kindly sharing their detailed prediction results on the UCF101 set in [24]. This work was supported partially by the National Key Research and Development Program of China (2018YFB1004903), NSFC (No. 61522115, 61702567, 61472456, 61661130157, 61628212), the Royal Society Newton Advanced Fellowship (NA150459), Guangdong Province Science and Technology Innovation Leading Talents (2016TX03X157), Guangdong NSF (2016A030313292,2015A030310450n). The corresponding author for this paper and the principal investigator for this project is Wei-Shi Zheng.

REFERENCES

[1] M. Bregonzio, S. Gong, and T. Xiang. Recognising action as clouds of space-time interest points. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1948–1955, 2009.

[2] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J. Siskind, and S. Wang. Recognize human activities from partially observed videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2658–2665, 2013.

[3] M. Chen, Z. Xu, K. Q. Weinberger, and F. Sha. Marginalized denoising autoencoders for domain adaptation. In *International Coference on International Conference on Machine Learning*, pages 1627–1634, 2012.

[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Gated feedback recurrent neural networks. In *International Conference on Machine Learning*, pages 2067–2075, 2015.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[6] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005.

[7] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1110–1118, 2015.

[8] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars. Rank pooling for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):773–787, 2017.

[9] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649, 2013.

[10] A. Gupta and L. S. Davis. Objects in action: An approach for combining action understanding and object perception. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[11] M. Hoai and F. De la Torre. Max-margin early event detectors. *International Journal of Computer Vision*, 107(2):191–202, 2014.

[12] J.-F. Hu, W.-S. Zheng, J. Lai, S. Gong, and T. Xiang. Exemplar-based recognition of human–object interactions. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(4):647–660, 2016.

[13] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5344–5352, 2015.

[14] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2186–2200, 2017.

[15] J.-F. Hu, W.-S. Zheng, L. Ma, G. Wang, and J. Lai. Real-time rgb-d activity prediction by soft regression. In *European Conference on Computer Vision*, pages 280–296. Springer, 2016.

[16] N. Hu, Z. Lou, G. Englebienne, and B. Kröse. Learning to recognize human activities from soft labeled data. *Proceedings of Robotics: Science and Systems, Berkeley, USA*, 2014.

[17] D. Huang, S. Yao, Y. Wang, and F. De La Torre. Sequential max-margin event detectors. In *European Conference on Computer Vision*, pages 410–424. Springer, 2014.

[18] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *International Joint Conference on Artificial Intelligence*, pages 2466–2472, 2013.

[19] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[20] K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert. Activity forecasting. *European Conference on Computer Vision*, pages 201–214, 2012.

[21] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference*, pages 275–1, 2008.

[22] Y. Kong and Y. Fu. Max-margin action prediction machine. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1844–1858, 2016.

[23] Y. Kong, D. Kit, and Y. Fu. A discriminative model with multiple temporal scales for action prediction. In *European Conference on Computer Vision*, pages 596–611, 2014.

[24] Y. Kong, G. Shangqian, S. Bin, and Y. Fu. Action prediction from videos via memorizing hard-to-predict samples. In *Conference on Artificial Intelligence*, 2018.

[25] Y. Kong, Z. Tao, and Y. Fu. Deep sequential context networks for action prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1473–1481, 2017.

[26] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013.

[27] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):14–29, 2016.

[28] T. Lan, T.-C. Chen, and S. Savarese. A hierarchical representation for future action prediction. In *European Conference on Computer Vision*, pages 689–704. Springer, 2014.

[29] K. Li and Y. Fu. Prediction of human activity by discovering temporal sequence patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1644–1657, 2014.

[30] Z. Li, J. Liu, J. Tang, and H. Lu. Robust structured subspace learning for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2085–2098, 2015.

[31] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan. Semantic object parsing with graph lstm. *arXiv preprint arXiv:1603.07063*, 2016.

[32] I. Lillo, A. Soto, and J. C. Niebles. Discriminative hierarchical modeling of spatio-temporally composable human activities. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 812–819, 2014.

[33] L. Lin, L. Huang, T. Chen, Y. Gan, and H. Cheng. Knowledge-guided recurrent neural network learning for task-oriented action prediction. *arXiv preprint arXiv:1707.04677*, 2017.

[34] J. Liu, A. Shahroudy, D. Xu, A. K. Chichung, and G. Wang. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[35] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot. Global context-aware attention lstm networks for 3d action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[36] C. Lu, J. Jia, and C.-K. Tang. Range-sample depth feature for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 772–779, 2014.

[37] S. Ma, L. Sigal, and S. Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1942–1950, 2016.

[38] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Sequence of the most informative joints (smij): A new representation for human skeletal action recognition. *Journal of Visual Communication and Image Representation*, 25(1):24–38, 2014.

[39] O. Oreifej and Z. Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2013.

[40] P. H. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *International Conference on Machine Learning*, pages 82–90, 2014.

[41] M. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *International Conference on Computer Vision*, pages 1036–1043, 2011.

[42] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. *arXiv:1604.02808*, 2016.

[43] A. Shahroudy, T.-T. Ng, Q. Yang, and G. Wang. Multimodal multipart learning for action recognition in depth videos. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2123–2129, 2016.

[44] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[45] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang. Cdc: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[46] B. Shuai, Z. Zuo, G. Wang, and B. Wang. Dag-recurrent neural networks for scene labeling. *arXiv:1509.00552*, 2015.

[47] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information*

Processing Systems, pages 568–576, 2014.

[48] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[49] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[50] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[51] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[52] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. *arXiv preprint arXiv:1504.08023*, 2015.

[53] H. Wang and C. Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[54] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3d action recognition with random occupancy patterns. In *European Conference on Computer Vision*, pages 872–885. 2012.

[55] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Learning actionlet ensemble for 3d human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11), 2013.

[56] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4305–4314, 2015.

[57] L. Wang, Y. Xiong, D. Lin, and L. Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *EEE Conference on Computer Vision and Pattern Recognition*, 2017.

[58] Y. Wang, J. Song, L. Wang, L. Van Gool, and O. Hilliges. Two-stream sr-cnns for action recognition in videos. In *British Machine Vision Conference*, 2016.

[59] P. Wei, Y. Zhao, N. Zheng, and S.-C. Zhu. Modeling 4d human-object interactions for event and object recognition. In *International Conference on Computer Vision*, pages 3272–3279, 2013.

[60] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[61] L. Xia and J. Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2834–2841, 2013.

[62] L. Xia, C.-C. Chen, and J. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 20–27, 2012.

[63] Z. Xu, L. Qing, and J. Miao. Activity auto-completion: Predicting human activities from partial videos. In *IEEE International Conference on Computer Vision*, pages 3191–3199, 2015.

[64] X. Yang and Y. Tian. Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 14–19, 2012.

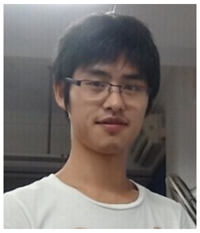
[65] X. Yang and Y. Tian. Action recognition using super sparse coding vector with spatio-temporal awareness. In *European Conference on Computer Vision*, pages 727–741. 2014.

[66] X. Yang and Y. Tian. Super normal vector for activity recognition using depth sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 804–811, 2014.

[67] G. Yu, Z. Liu, and J. Yuan. Discriminative orderlet mining for real-time recognition of human-object interaction. In *Asian Conference on Computer Vision*, pages 50–65. Springer, 2015.

[68] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *IEEE International Conference on Computer Vision*, pages 2752–2759, 2013.

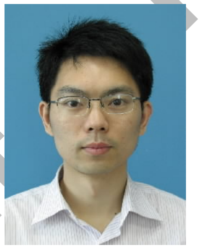
[69] J. Zhu, B. Wang, X. Yang, W. Zhang, and Z. Tu. Action recognition with actons. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3559–3566, 2013.



Jian-Fang Hu is currently a research associate professor at Sun Yat-sen University. He received the PhD and B.S. degrees from the School of Mathematics, Sun Yat-Sen University, Guangzhou, China, in 2016 and 2010, respectively. His research interests include human-object interaction modeling, 3D face modeling, and RGB-D action recognition. He has published several scientific papers in the international conferences and journals including ICCV, CVPR, ECCV, IEEE TPAMI, IEEE TCSVT, and PR.

Wei-Shi Zheng is now a professor at Sun Yat-sen University. His research interests include person association and activity understanding in visual surveillance. He has now published more than 100 papers, including more than 70 publications in main journals (TPAMI,TNN,TIP,PR) and top conferences (ICCV, CVPR,IJCAI,AAAI). He served as an area chair for AVSS 2012, ICPR 2018 and BMVC 2018. He has joined Microsoft Research Asia Young Faculty Visiting Programme. He is a recipient of Excellent Young Scientists Fund of the National Natural Science Foundation of China, and a recipient of Royal Society-Newton Advanced Fellowship, United Kingdom. He is an associate editor of the Pattern Recognition Journal.

Lianyang Ma received his Ph.D. degree from Shanghai Jiao Tong University, China, in 2014. From March 2015 to March 2016, he worked as a Research Fellow in Nanyang Technological University, Singapore. His current research interests include computer vision, machine learning and personalized recommender systems.



Wang Gang is currently a researcher/senior director and a distinguished scientist in Alibaba AI Labs. He was an Associate Professor with the School of Electrical and Electronic Engineering at Nanyang Technological University (NTU). He had a joint appointment at the Advanced Digital Science Center (Singapore) as a research scientist from 2010 to 2014. He received his B.Eng. degree from Harbin Institute of Technology in Electrical Engineering and the PhD degree in Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. He is a recipient of MIT technology review innovator under 35 award (Asia). He is an associate editor of TPAMI and an area chair of ICCV 2017.

Jian-Huang Lai received the Ph.D. in mathematics from Sun Yat-Sen University in 1999. He is a Professor and the Dean of the School of Information Science and Technology. His current research interests are in the areas of digital image processing, pattern recognition, multimedia communication, wavelet, and its applications. He has published over 100 scientific papers in international journals and conferences including IEEE TPAMI, IEEE TNN, IEEE TIP, IEEE TSMC-B, PR, ICCV, CVPR, and ICDM.

Jianguo Zhang is currently a Reader at Computing in the School of Science and Engineering, University of Dundee, UK. He received a PhD in National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, 2002. His research interests include visual surveillance, object recognition, image processing, medical image analysis and machine learning.