Progressive Teacher-student Learning for Early Action Prediction

Xionghui Wang[1] , Jian-Fang Hu[1,3*] , Jianhuang Lai[1,3] , Jianguo Zhang[2] , and Wei-Shi Zheng[1,4]

[1]Sun Yat-sen University, China; [2]University of Dundee, United Kingdom
[3]Guangdong Province Key Laboratory of Information Security Technology, China
[4]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
wxiongh@mail2.sysu.edu.cn, hujf5@mail.sysu.edu.cn, stsljh@mail.sysu.edu.cn,
j.n.zhang@dundee.ac.uk, wszheng@ieee.org

Abstract

The goal of early action prediction is to recognize actions from partially observed videos with incomplete action executions, which is quite different from action recognition. Predicting early actions is very challenging since the partially observed videos do not contain enough action information for recognition. In this paper, we aim at improving early action prediction by proposing a novel teacher-student learning framework. Our framework involves a teacher model for recognizing actions from full videos, a student model for predicting early actions from partial videos, and a teacher-student learning block for distilling progressive knowledge from teacher to student, crossing different tasks. Extensive experiments on three public action datasets show that the proposed progressive teacher-student learning framework can consistently improve performance of early action prediction model. We have also reported the state-of-the-art performances for early action prediction on all of these sets.

1. Introduction

Early action prediction, i.e., predicting the label of actions before they are fully executed, is one of the most fundamental tasks in video analysis with many real-world applications in surveillance, self driving, and human-computer interaction etc. Different from the traditional action recognition task that intends to recognize actions from full videos, early action prediction aims to predict the label of actions from partially observed videos with incomplete action executions.

As shown in Figure 1, recognizing actions from partial videos is very challenging, especially when the depicted actions are performed at very early stages (e.g., when 10% of an action is executed). However, the recognition would become much easier if the actions are fully executed and observed [16, 24]. Videos with different observation ratios often contain different degree of action context. How to mine as much action knowledge as possible from these partially or fully observed videos for prediction is one of the major challenges in the community.

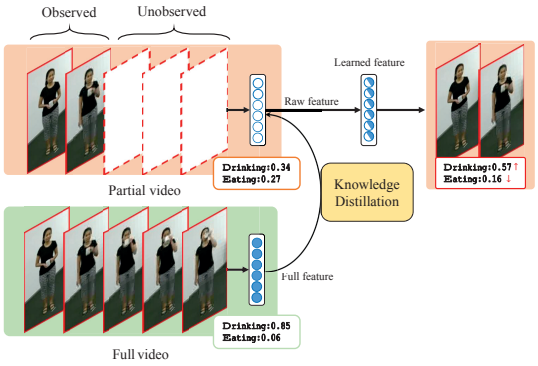


Figure 1: Schematic diagram showing our motivation of proposing distilling knowledge from action recognition system for early action prediction.

Many works have been proposed to exploit the partially and fully observed videos for early action prediction. For instance, Kong et.al. [21] assume that the prediction confidences are monotonically increasing as more video frames are observed. Hu et.al. [16] intends to learn a soft label for the video of each progress level so that the full and partial videos can be learned in a unified regression framework. More recently, Kong et.al. [24] learns a reconstruction map from all partially observed videos to full videos. These works mainly develop a joint learning framework to learn early action predictor from partial and full videos, they do not seek to distill some disriminative action knowledge from the full videos to improve the early action prediction with partial videos. As illustrated in Figure 1, the action

*Corresponding author

knowledge gained from full videos can be used to drive the early action prediction with partial videos.

In this paper, we formulate a novel knowledge distillation framework for early action prediction. Our framework involves a teacher model for recognizing actions from full videos, a student model for predicting early actions from partial videos, and a teacher-student learning block for distilling knowledge from teacher to student. To the best of our knowledge, we are the *first* to explicitly formulate a teacher-student learning framework for early action prediction, particularly for casting it as a problem of progressive knowledge distillation across different tasks, with both mean square error (*MSE*) and maximum mean discrepancy (*MMD*) loss considered in an unified framework to distill local progress-wise and global distribution knowledge, respectively. The experimental results show that the proposed progressive teacher-student learning framework is beneficial for early prediction of actions, especially when the actions are performed at very early stages.

In summary, the main contributions of this work are three-fold: 1) a novel teacher-student learning framework for distilling progressive action knowledge from action recognition model (teacher) to early action prediction model (student), across different tasks; 2) based on the proposed teacher-student learning, an early action prediction system integrating the early action prediction task with action recognition in the spirit of knowledge distillation; and 3) extensive experimental analysis on early action prediction with RGB-D and RGB videos on three datasets, showing that our early action prediction system achieves state-of-the-art performances and the proposed teacher-student learning framework can efficiently improve the prediction performance by knowledge distillation.

2. Related Work

Action recognition. Action recognition has been widely studied in the community. The existing methods are mainly developed for extracting some discriminative action features from videos with complete action executions. Some representative handcrafted features like Cuboids [7] [44] [6], interest point clouds [4], 3DHOG [20], SIFT [33], and dense trajectory [41] etc. are developed for characterizing the spatio-temporal motion information, which is critical for describing human actions. Recently, with the rise of deep learning, many deep learning based methods, including 3D CNN [39] [5] [12] [40] [13] and two stream CNN [36] [8] [46] etc., are proposed to encode the spatio-temporal information and achieve satisfactory recognition results in many datasets, including UCF-101 [37] and Kineticts [5] datasets. Besides these advances for recognizing actions from RGB videos, action recognition with depth camera has also made some encouraging progress in these years. Some researchers found that combining the multi-modality features extracted from RGB, depth and skeleton sequences can capture more useful action information and obtain a better recognition performance [45] [17] [35] [31]. However, these action recognition approaches are specifically developed for the after-the-fact prediction of human action (i.e. when actions are entirely observed), and they didn't seek to build models for predicting early actions at different progress levels, which in particular requires modeling the intrinsic expressive power of partial videos.

Early action prediction. Different from action recognition, where actions are assumed to be fully executed and observed, early action prediction aims to recognize actions before they are completely executed [32, 25, 23, 24, 2, 16, 22]. Actions at early stage are very difficult to be recognized due to the lacking of sufficient information. Ryoo [32] intended to recognize ongoing actions by observing some evidences from the features accumulated over time. Lan et al. [25] employed a max-margin framework to predict actions from a hierarchical feature representation. Kong et al. [23] developed a structured SVM formulation to capture the temporal evolution of human actions. Hu et al. [16] proposed a soft regression framework to learn a robust action predictor from both the partial videos and full videos. Aliakbarian et al. [2] introduced a multi-stage LSTM architecture to model context-aware and action-aware information. Recently, Kong et al. [24] proposed a deep sequential context networks (DeepSCN), with an aim to reconstruct the features of full videos from those extracted from partial videos. None of these works are proposed to exploit the action recognition task for early action prediction, which could discover some informative action knowledge for early action prediction. With that in mind, we develop a novel teacher-student learning framework to distill knowledge from an action recognition model, for the purpose of improving early action prediction.

Knowledge distillation. Recent studies show that the knowledge learned by a teacher network can be used to improve the performance of student network [14, 27, 1, 43]. In the literature, teacher network often refers to a heavily, cumbersome model and student network refers a simple, lightweight model. Both the teacher and student networks are oriented for addressing the same task. For example, Romero et al. [1] proposed to minimize the MSE between the outputs of teacher and student models. Yim et al. [42] used a Gram matrix loss to distill knowledge for improving image classification. Li el at. [27] showed that minimizing Gram matrix loss in neural style transfer [9] is equivalent to minimize the MMD loss [11]. These works are mainly developed for distilling knowledge in static images and within the same task. In contrast, we aim to transfer the sequential knowledge gained in recognition model, to improve our prediction performance. Thus, our method seeks to transfer knowledge across different video analysis tasks.

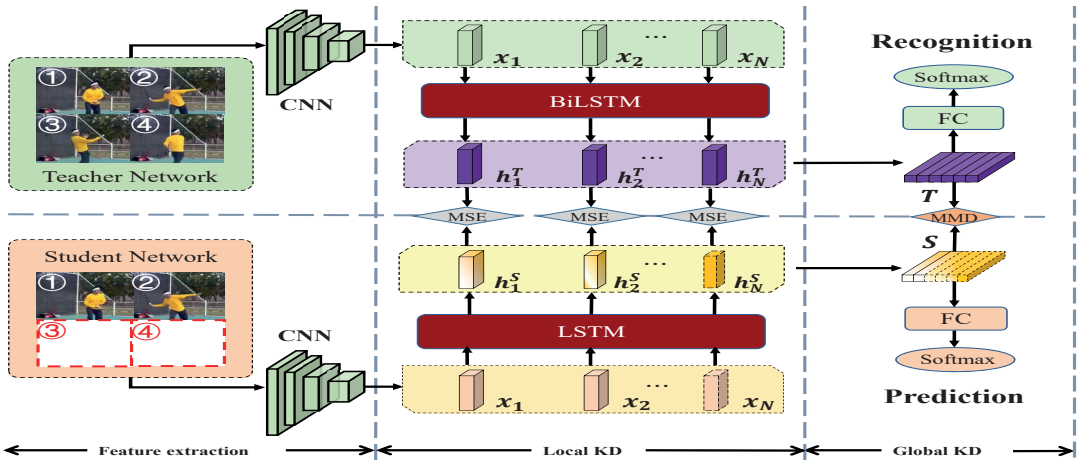


Figure 2: The overall framework of our progressive teacher-student learning for early action prediction.

3. Our Approach

We tackle the same problem as in [16, 24, 22], i.e., to learn a common early action predictor for predicting early actions. Following the existing setting in [16, 24, 21, 32, 25, 26], we assume that each training video contains complete action execution and uniformly partition it into N shorter segments. The first n segments ($n = 1, 2, ..., N$) form a partial video with a progress level of n, whose observation ratio is defined as n/N. Let's denote $\boldsymbol{x}_n$ as the feature extracted from the partial video of progress level n.

In this work, we focus on developing a teacher-student learning framework to improve an early action prediction model (referred as student) with the assistance of an action recognition model (referred as teacher). In the following, we first describe our teacher and student models and then show how to distill useful knowledge from the teacher model to improve our student model.

3.1. Teacher and Student Networks

Student model. Hu et. al. [16] observed that explicitly learning the temporal dependencies among the videos of different progress levels is beneficial for early action prediction. Here, we follow this observation and employ a standard 1-layer long short-term memory (LSTM) [15] architecture as our student prediction model to predict early actions at any progress level.

Teacher model. Here, we specify our teacher model by a 1-layer bidirectional LSTM (BiLSTM) [10] architecture, which has been widely used for addressing the video recognition problem. We use the BiLSTM model as our teacher model for two aspects. First, it can provide a latent feature representation for the videos of any progress level, which is consistent with the student model. Second, since the BiLSTM has a forward LSTM and a backward LSTM layer, which can receive information from the historical frames and future frames, respectively, the latent features obtained by BiLSTM often contain more discriminative action information than those obtained by the student LSTM model, especially for the actions at very early stages. However, the BiLSTM model is not applicable for early action prediction, as the frames after the current observation is often unreachable in practice. Even though, we demonstrate that it could still be used for early action prediction. In particular, it can be treated as a teacher model to guide our student learning. To this end, we propose a teacher-student learning method to make use of the rich latent features obtained by the teacher model for improving our early action prediction.

3.2. Progressive Teacher-student Learning

With the teacher model described previously, our goal is to distill some useful knowledge from the teacher model to facilitate the student prediction model learning. Here, we achieve the knowledge distillation by developing a teacher-student learning block, which would link the progress-wise latent feature representations obtained by the teacher network and student network, as illustrated in Figure 2. In the following, we describe our formulation for the teacher-student learning block in detail.

Teacher-student learning block. Let us denote the latent feature representations of the teacher and student networks over all the progress levels for the $i-th$ video sample by $\boldsymbol{S}_i$ and $\boldsymbol{T}_i$, respectively. Here, $\boldsymbol{S}_i$ and $\boldsymbol{T}_i$ are two $D \times N$-sized matrices. D indicates the feature dimension and N is the total number of progress levels used for early action prediction. Then, our knowledge distillation can be achieved by minimizing

$$L = \frac{1}{I}\sum_{i=1}^{I}(L_C(\boldsymbol{S}_i, \boldsymbol{y}_i) + L_{TS}(\boldsymbol{S}_i, \boldsymbol{T}_i)) \qquad (1)$$

where L_{TS} indicates the knowledge distillation (KD) loss and L_C is the prediction loss of the student model. $\boldsymbol{y}_i$ indicates the ground truth action label for the i-th video sample.

KD Loss $L_{TS}(\boldsymbol{S}_i, \boldsymbol{T}_i)$. We define the KD loss as $\alpha L_{MSE} + \beta L_{MMD}$, where L_{MSE} is used to distill knowledge in a progress-wise manner by computing the mean square error (MSE) between the latent features of teacher and student models. Thus, it can capture some *local* action knowledge with respect to the videos at each individual progress level for distillation. The loss L_{MMD} is employed to measure the maximum mean discrepancy (MMD) between the teacher recognition model (with full videos) and the early action prediction model (with partial videos). Minimizing L_{MMD} can distill knowledge for the videos of all the progress levels from a *global* distribution perspective.

We formulated L_{MSE} as $\|\boldsymbol{S}_i \odot \boldsymbol{w} - \boldsymbol{T}_i \odot \boldsymbol{w}\|_F^2$. $\boldsymbol{w}$ is a weight vector indicating the contribution of MSE losses with respect to the videos of each individual progress level. $\odot$ is an element-wise multiplication operator, multiplying each column of $\boldsymbol{S}_i$ with the corresponding element of $\boldsymbol{w}$. Minimizing L_{MSE} is to decrease the discrepancy between the knowledge gained by the teacher and student models for the videos of each individual progress level.

In contrast, MMD is widely used to measure the distance between two distributions [11]. Here, the loss L_{MMD} is employed to control the global distribution discrepancy for the videos of all the progress levels. Our MMD loss can be defined as

$$L_{MMD}(\boldsymbol{S}_i, \boldsymbol{T}_i) = \| \frac{1}{N} \sum_{n=1}^{N} \phi(\boldsymbol{S}_i(:,n)) - \frac{1}{N} \sum_{n=1}^{N} \phi(\boldsymbol{T}_i(:,n)) \|_2^2 \tag{2}$$

ϕ is a function mapping the latent feature representation to Reproducing Kernel Hilbert Space (RKHS), which corresponds to a kernel function $k(\boldsymbol{x}, \boldsymbol{y}) = < \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) >$. We follow the suggestions in [27] and set it as a specific second order polynomial kernel function $k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^\top \boldsymbol{y})^2$. Then the MMD loss can be equivalently rewritten as:

$$L_{MMD}(\boldsymbol{S}_i, \boldsymbol{T}_i) = \|\boldsymbol{G}^{\boldsymbol{S}_i} - \boldsymbol{G}^{\boldsymbol{T}_i}\|_F^2 \tag{3}$$

Here, $\boldsymbol{G}^{\boldsymbol{S}_i}$ and $\boldsymbol{G}^{\boldsymbol{T}_i}$ are Gram matrices:

$$\boldsymbol{G}^{\boldsymbol{S}_i} = \boldsymbol{S}_i \boldsymbol{S}_i^\top, \boldsymbol{G}^{T_i} = \boldsymbol{T}_i \boldsymbol{T}_i^\top \tag{4}$$

where $\boldsymbol{G}^* \in \mathbb{R}^{D \times D}$, $(\cdot)^\top$ stands for matrix transposition. Note that the representations $\boldsymbol{S}_i$ and $\boldsymbol{T}_i$ are l_2 normalized at each progress level to avoid significant discrepancy.

Overall, the KD Loss can be expressed as

$$\begin{aligned} L_{TS}(\boldsymbol{S}_i, \boldsymbol{T}_i) = \alpha \|\boldsymbol{S}_i \odot \boldsymbol{w} - \boldsymbol{T}_i \odot \boldsymbol{w}\|_F^2 \\ + \beta \|\boldsymbol{S}_i \boldsymbol{S}_i^\top - \boldsymbol{T}_i \boldsymbol{T}_i^\top\|_F^2 \end{aligned} \tag{5}$$

where α and β are used to control the impact of MSE loss and MMD loss respectively when combined with the prediction loss in Eq. (1).

Prediction loss. As for prediction, we treat the early action prediction as a problem of recognizing actions from ongoing videos (partial or full) with unknown progress levels. For simplification, we directly feed the latent feature representation of student LSTM model into a FC layer (with a parameter $\boldsymbol{W}_F$) to conduct prediction. Note that the classifier weight $\boldsymbol{W}_F$ is shared for all the videos of different progress levels. Then our prediction loss can be defined as:

$$L_C(\boldsymbol{S}_i, \boldsymbol{y}_i) = \sum_{n=1}^{N} l(\boldsymbol{p}_i^n, \boldsymbol{y}_i) \tag{6}$$

Here, $l(\boldsymbol{p}_i^n, \boldsymbol{y}_i)$ is the standard cross-entropy loss between prediction results $\boldsymbol{p}_i^n$ and the ground truth action label $\boldsymbol{y}_i$ at the progress level n, where $\boldsymbol{p}_i^n = softmax(\boldsymbol{W}_F \boldsymbol{S}_i(:,n))$.

Model learning. In our teacher-student learning framework, the teacher model is assumed to be previously prepared, which means that it is trained from the training data and then fixed for learning the student model[1]. Similar to other teacher-student learning framework [1], we also employ a two-stage optimization method to obtain a robust estimation for the student model. At the first stage, we directly minimize the KD loss over the LSTM parameters (without FC layer), without taking into account the prediction loss, which requires the student to predict the latent features of the teacher network. We empirically find that training the LSTM layer in this way can provide a good initialization for tuning the student model. At the second stage, we learn the LSTM parameter and classifier together by minimizing L in Eq. (1). Our experiments in Section 5 show that the student model learned in this way can achieve better results.

4. Experiments

We tested our method for early action prediction with RGB-D and RGB videos on three benchmark datasets: NTU RGB-D action [34], SYSU 3DHOI [17], and UCF-101 set [37]. In the following, we will describe the implementation details, and experimental results with detailed analysis.

4.1. Implementation details

For early action prediction on the RGB-D action datasets (i.e., NTU RGB-D and SYSU 3DHOI sets), we followed the settings in [16] and partition each video clip into $N = 40$ shorter segments. While for the prediction on RGB action dataset (i.e., UCF-101 set), we used the settings in [24] and divided each full video into $N = 10$ shorter segments.

Details for feature extraction. For extracting visual features from videos in RGB-D dataset (NTU and SYSU), we followed the implementations in [16] and uniformly sampled 16 frames from each video clip, from which a set of image patches containing the actors are cropped, in order to reduce the influence of cluttered backgrounds. These patches

[1]Indeed, we do not observe any improvement on the prediction performance by training the teacher recognition model and student student model jointly. Please refer to Section 5 for details.

Figure 3: Some frame examples from the NTU RGB-D, SYSU 3DHOI and UCF-101 datasets, The first two rows present RGB and depth frames from NTU set. The next two rows provide some examples from SYSU 3DHOI set. And the last row gives examples from the UCF-101 set.

are then concatenated (after resized) along the temporal dimension to form a $16\times299\times299$-sized tensor. We then finetuned a 16-channel-InceptionResNetV2[2] [38] model based on the tensors generated from RGB and depth videos, respectively. Since all the actions in SYSU set involve human-object interactions, we followed the suggestion in [18] and extracted CNN features from the image patches of human body parts. For extracting features from 3D skeleton sequences, we followed the pre-processing step in [34] and transformed the 3D locations of human joints from camera coordinate system to body coordinate system. We sampled 10 frames from each partial or full skeleton sequence and then fed them into a RNN model to extract corresponding features. Finally, the features extracted from RGB, depth, and skeleton data were concatenated to obtain representation $\{\boldsymbol{x}_n\}_{n=1}^N$ for the videos of each progress level.

For extracting features from videos in unconstrained RGB action set UCF-101, we used the 3D ResNeXt-101 [13] pre-trained on Kinetics dataset [19] to extract some spatio-temporal features without finetuning on the training data[3]. More specifically, we sampled 16 frames from each video clip (partial or full) and then re-sized them into a $3\times16\times112\times112$-sized tensor. Finally, these tensors were fed into the 3D ResNeXt-101 to extract visual features.

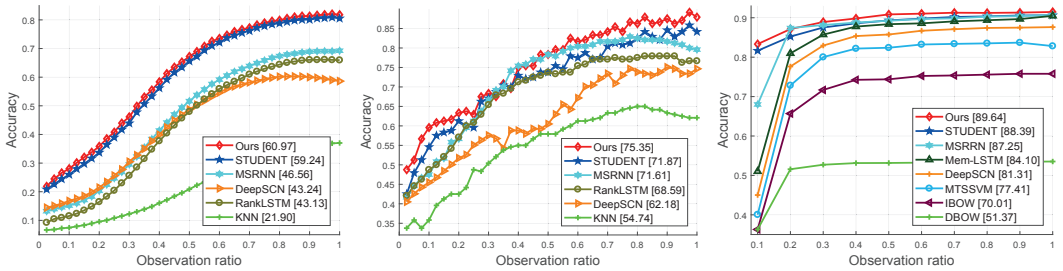
Details for teacher-student learning. We trained a 1-layer sequence-sequence Bi-LSTM network as our teacher model, where the dimension of hidden layer in each direction (forward and backward) was set as 256. Hence, the dimension of the latent features output by teacher model is 512. We employed the cross-entropy loss over all the progress levels as the loss of our teacher learning. For the student model, we trained a 1-layer sequence-sequence LSTM network with the size of the hidden layer set as 512, in order to match with the latent features output by teacher model. We set $\boldsymbol{w}$ as that in [16], as we experimentally find that learning it from scratch can only earn a minor improvement. The weights (α, β) for controlling MSE and MMD loss were set to (0.1, 0.02), (25, 0.002), (4, 0.02) on NTU RGB-D, SYSU 3DHOI, and UCF-101, respectively. The learning rate and batch sizes were set to 0.01 and 30 for SYSU dataset, 0.1 and 256 for both NTU and UCF-101 datasets, respectively. Dropout was utilized in student model to mitigate overfitting. We used SGD optimizer [3] with a momentum rate 0.9 to train both teacher and student networks. All the experiments were conducted in PyTorch [30].

[2]https://github.com/Cadene/pretrained-models.pytorch

[3]We do not observe any improvement on the performance by finetuning 3D ResNeXt-101 in our experiments.

4.2. NTU RGB-D Action Dataset

To the best of our knowledge, the NTU RGB-D action dataset [34] is by far the largest public set for 3D action recognition and prediction. It contains more than 56,000 video samples with about 4 million frames from 60 action categories. All of these action samples were recorded by Kinect v2 devices from three different views. For collecting this set, 40 subjects were asked to perform certain actions several times. Some action frames can be found in Figure 3. This set is very challenging for early action prediction mainly due to its larger scales in quantity, greater diversity in action categories and more complexity in human-human interaction and human-object interaction. Moreover, many actions considered in this set are easily confused with each other at the beginning stages. Taking the actions "*eating*"



(a) NTU RGB-D action set (b) SYSU 3D HOI set (c) UCF-101 set

Figure 4: Comparison results on the (a) NTU RGB-D Action, (b) SYSU 3DHOI and (c) UCF-101 sets. [*] in the legend of the figure stands for the AUC(%) performance obtained by the corresponding method.

Table 1: Prediction results (%) on the NTU RGB-D Action set.

Observation ratio	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%	AUC
KNN [16]	7.45	9.56	12.25	16.04	20.89	25.97	30.85	34.49	36.15	37.02	21.90
RankLSTM [28]	11.54	16.48	25.66	37.74	47.96	55.94	60.99	64.41	66.05	65.95	43.13
DeepSCN [24]	16.80	21.46	30.51	39.93	48.73	54.61	58.18	60.18	60.01	58.62	43.24
MSRNN [16]	15.17	20.33	29.53	41.37	51.64	59.15	63.91	67.38	68.89	69.24	46.56
STUDENT	25.99	33.68	43.91	56.20	65.59	72.12	76.16	78.82	80.09	80.53	59.24
Ours	**27.80**	**35.85**	**46.27**	**58.45**	**67.40**	**73.86**	**77.63**	**80.06**	**81.47**	**82.01**	**60.97**

and "*drinking*" for example, both of them have the moments of fetching object and holding it up. For experiments, we followed exactly the cross-subject setting in [34, 16] and used the samples performed by 20 certain subjects to train our model. The samples performed by the rest subjects are employed to evaluate the learned model. In total, we have 40,320 full videos for training and 16,560 full videos for testing, which means that we have a total of 662,400 partial and full samples to test the trained model.

The detailed prediction results are presented in Figure 4(a) and Table 1, where we denote the student model without distilling knowledge from teacher model as STUDENT. As shown, with the help of a teacher model, the performances of our student model are improved at all of the 40 progress levels, especially for the actions at very early stage. For instance, when only using the first 30% segments for prediction, our system achieves an accuracy of 46.27%, outperforming the STUDENT model by 2.36%. We also observe that the accuracy of prediction actions from full videos is 82.01%, which is 1.48% higher than STUDENT. This demonstrates that the knowledge distillation framework is also beneficial for the action recognition task. From the perspective of area under the curve (AUC), which stands for the average prediction accuracy, it increases from 59.24% to 60.97% with an improvement of 1.73%, as compared to STUDENT, which means that more than 11,400 mis-predicted action samples are correctly predicted when using our progressive teacher-student learning method.

Table 1 shows the comparison of our method with other state-of-the-art methods [16, 24, 28]. It could be observed that our method outperforms the competitors by a large margin (more than 14% in terms of AUC), which is a significant breakthrough for early action prediction on this challenging dataset. The results demonstrate the effectiveness of our early action prediction system with knowledge distillation.

4.3. SYSU 3DHOI dataset

The SYSU 3D Human-Object Interaction (3DHOI) dataset [17] was captured by Kinect v1, with 480 RGB-D sequences from 12 action categories, including "*playing phone*", "*calling phone*", "*pouring*", "*drinking*", etc. Each action involves a kind of human-object interactions. Similar to the NTU RGB-D action set, the collectors invited 40 actors to perform 12 human-object interaction actions with six different objects. Some frame examples can be found in Figure 3. This set is challenging for early action prediction as the actions are quite similar to each other, especially at the beginning stages. For instance, the actions of "*calling phone*" and "*playing phone*" have the same movement of picking up a phone. Thus, it is not easy for the system to accurately infer action by only observing a small part of sequences. Following the same evaluation setting with [16], sequences performed by the first 20 subjects were used for training and the rest for testing. For evaluation, each full sequence was uniformly partitioned into 40 segments. Therefore, we have a total of 9,600 video clips (both full and partial) to test the learned prediction models in this set.

Figure 4(b) and Table 2 present the detailed prediction results on this set. As shown, our method obtains an AUC of 75.35%, outperforming all of the competitors including STUDENT, RankLSTM [28], DeepSCN [24], and MSRNN [16]. As expected, the proposed teacher-student learning

Table 2: Prediction results (%) on the SYSU 3DHOI set.

Observation ratio	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%	AUC
KNN [16]	35.83	42.50	50.42	55.00	57.92	61.25	63.33	65.00	63.33	62.08	54.74
RankLSTM [28]	48.75	57.08	65.42	71.25	73.75	75.42	77.08	77.50	77.92	76.67	68.59
DeepSCN [24]	45.50	51.75	57.58	58.83	60.50	67.17	73.42	73.83	75.08	74.67	62.18
MSRNN [16]	47.50	56.67	66.67	**75.42**	**78.33**	80.42	81.67	82.50	81.67	79.58	71.61
STUDENT	54.58	61.25	67.08	72.92	73.75	77.50	80.42	82.50	84.58	84.17	71.87
Ours	**59.58**	**63.33**	**68.33**	75.00	**78.33**	**81.67**	**84.17**	**86.25**	**87.50**	**87.92**	**75.35**

framework has consistently improved the performance of our student model by a large margin (about 3.5%), especially for the prediction of actions at very early stages. By only observing the first 10% videos, our system can obtain an accuracy of 59.58%, which clearly exceeds the performances obtained by our student model without teacher student learning and other competitors. These aspects demonstrate that the proposed progressive teacher-student learning framework can efficiently facilitate the learning of early action prediction model.

4.4. UCF-101 dataset

The UCF-101 set is unconstrained RGB video based dataset, which has been widely used for action recognition. It consists of 13,320 full videos from 101 action classes, such as "*Playing Guitar*" and "*Basket-ball Dunk*". Most of the considered actions involve human-object interaction, body-motion, human-human interaction and sports. Figure 3 presents some frame examples in this set. For evaluation, we employed the same settings as [22, 16] and used the first 15 groups of videos for training, the next 3 groups for validation, and the rest for testing. In this setting, we have 3,682 full action videos for test and each video is split into $N = 10$ segments, which means that we need to predict the actions of 36,820 clips in this experiment.

The detailed prediction results are presented in Figure 4(c) and Table 3. As expected, the results obtained in this study are consistent with those obtained on the NTU RGB-D action and SYSU 3DHOI sets. Our proposed teacher-student learning framework can consistently improve the prediction performance of our student model and outperforms the other state of the arts [22, 16, 24]. It is worth noting that our system can obtain an accuracy of 83.3% for predicting partial videos with a progress level of 10%, outperforming the state of the art approach [16] by a margin of 5.32%. When more video frames are provided, the accuracy will keep rising until all frames are observed. Overall, the prediction accuracies obtained on this set are much higher than those on the NTU RGB-D and SYSU 3DHOI sets, especially for the prediction of actions at very early stages. This is because that many actions in this set can be recognized by only observing the scene context depicted in each single frame, e.g., "*playing billiards*" and "*archery*".

Table 4: More evaluation on the influence of MSE and MMD losses. S stands for STUDENT without knowledge distillation, L stands for local knowledge distillation with MSE, G stands for global knowledge with MMD.

Observation ratio		10%	30%	50%	70%	100%	AUC
SYSU	S	54.58	67.08	73.75	80.42	84.17	71.87
	S+L	57.08	67.08	75.83	80.42	85.83	73.53
	S+G	57.50	66.67	76.67	80.42	85.00	73.08
	S+L+G	**59.58**	**68.33**	**78.33**	**84.17**	**87.92**	**75.35**
UCF-101	S	81.64	87.53	89.33	90.20	90.63	88.39
	S+L	83.19	88.43	90.22	91.20	90.98	89.27
	S+G	**83.57**	88.02	90.14	90.63	90.71	89.01
	S+L+G	83.32	**88.92**	**90.85**	**91.28**	**91.47**	**89.64**

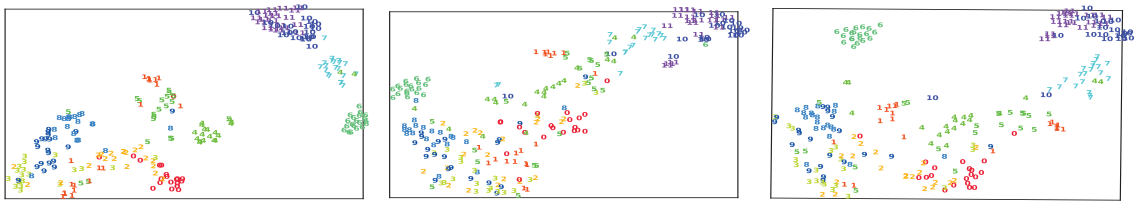
5. Ablation study

Here, we provide more evaluation results on the SYSU 3DHOI and UCF-101 sets.

Influence of MSE and MMD loss. Note that our KD loss for the teacher-student learning consists of two components, MSE and MMD, where MSE is employed to capture local progress-wise knowledge, and MMD is used to distill global distribution knowledge. Here, we study their influence and report the results in Table 4. As can be seen, distilling action knowledge, either local or global, is always beneficial for early action prediction. And a proper combination of them can obtain the best performance in most of the test cases.

Evaluation on the model optimization. In this paper, we have used a two-stage optimization method for determining the student parameters (denoted by Two). Intuitively, we can also directly optimize the objective function L in a one-stage manner (denoted by One). Here, we report the results of using the two strategies in Table 5. As shown, both optimization methods can improve our early action prediction, compared to the STUDENT only. We also note that the two-stage optimization approach can obtain better results than the one-stage training in our experiments. Especially on the SYSU 3DHOI set, the two-stage based method has a performance gain about 1.8%, which means that more than 140 samples are correctly predicted by the student model.

Visualization for the benefits of introducing teacher-student learning. Here, we use t-SNE [29] to visualize the latent features output by our teacher model and student model with/without teacher-student learning, respectively.



(a) Teacher (b) Student without teacher-student learning (c) Student with teacher-student learning

Figure 5: Visualization results. Samples from different actions are marked by different colors and numbers.

Table 3: Prediction results (%) on the UCF-101 set.

Observation ratio	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%	AUC
DBOW [32]	36.29	51.57	52.71	53.13	53.16	53.24	53.24	53.34	53.45	53.53	51.37
IBOW [32]	36.29	65.69	71.69	74.25	74.39	75.23	75.36	75.57	75.79	75.79	70.01
MTSSVM [23]	40.05	72.83	80.02	82.18	82.39	83.21	83.37	83.51	83.69	82.82	77.41
DeepSCN [24]	45.02	77.64	82.95	85.36	85.75	86.70	87.10	87.42	87.50	87.63	81.31
Mem-LSTM [22]	51.02	80.97	85.73	87.76	88.37	88.58	89.09	89.38	89.67	90.49	84.10
MSRNN [16]	68.00	**87.39**	88.16	88.79	89.24	89.67	89.85	90.28	90.43	90.70	87.25
STUDENT	81.64	85.23	87.53	88.59	89.33	89.79	90.20	90.36	90.58	90.63	88.39
Ours	**83.32**	87.13	**88.92**	**89.82**	**90.85**	**91.04**	**91.28**	**91.23**	**91.31**	**91.47**	**89.64**

Table 5: More evaluation on the optimization strategies. S stands for STUDENT without teacher-student learning.

Observation ratio		10%	30%	50%	70%	100%	AUC
SYSU	S	54.58	67.08	73.75	80.42	84.17	71.87
	One	57.08	66.25	77.08	82.50	85.42	73.57
	Two	**59.58**	**68.33**	**78.33**	**84.17**	**87.92**	**75.35**
UCF-101	S	81.64	87.53	89.33	90.20	90.63	88.39
	One	**83.41**	88.51	90.47	**91.31**	91.23	89.51
	Two	83.32	**88.92**	**90.85**	91.28	**91.47**	**89.64**

Table 6: Comparison on with vs. without joint learning.

Observation ratio		10%	30%	50%	70%	100%	AUC
SYSU	with	53.33	66.25	74.58	81.67	84.58	72.49
	without	**59.58**	**68.33**	**78.33**	**84.17**	**87.92**	**75.35**
UCF-101	with	**83.60**	88.35	89.82	90.20	90.85	89.07
	without	83.32	**88.92**	**90.85**	**91.28**	**91.47**	**89.64**

The results on the test videos from the SYSU 3DHOI set are shown in Figure 5. The teacher model performs better in separating samples of different action types than the student model, which illustrates that teacher model contains more powerful action information. By distilling these knowledge to the student model, the samples are better separated by our student model as illustrated in Figure 5(b) and Figure 5(c). This also demonstrates that some useful knowledge are distilled by our model to improve early action prediction.

Joint learning of teacher and student. During our model training, the teacher model was pre-trained and then fixed, we also test the case of jointly learning teacher and student networks simultaneously. The results are reported in Table 6. It is interesting to note that jointly learning teacher and student model obtains an inferior performance in our experiments, which could be attributed to the intractability of optimizing two highly non-convex problem simultaneously.

6. Conclusion

In this paper, we present a novel teacher-student learning framework for early action prediction. In the framework, the progressive knowledge gained in an action recognition model (teacher) is explicitly distilled for facilitating the learning of early action prediction model (student) learning. We achieve knowledge distillation by minimizing the local progressive-wise and global distribution knowledge discrepancy between the teacher and student models. Extensive experiments on two RGB-D action sets and one unconstrained RGB action set have been reported to demonstrate the efficacy of the proposed framework.

Acknowledgments

This work is partially supported by the National Key Research and Development Program of China (2018YFB1004903), NSFC (61702567, 61628212), SF-China (61772570), Pearl River S&T Nova Program of Guangzhou (201806010056), Guangdong Natural Science Funds for Distinguished Young Scholar (2018B030306025), and FY19-Research-Sponsorship-185. Jian-Fang Hu is also supported by the Opening Project of GuangDong Province Key Laboratory of Information Security Technology(2017B030314131) and the CCF-Tencent open research fund. The corresponding author is Jian-Fang Hu.

References

[1] Romero Adriana, Ballas Nicolas, K Samira Ebrahimi, Chassang Antoine, Gatta Carlo, and B Yoshua. Fitnets: Hints for thin deep nets. *International Conference on Learning Representations*, 2015. 2, 4

[2] Mohammad Sadegh Aliakbarian, F Sadat Saleh, Mathieu Salzmann, Basura Fernando, Lars Petersson, and Lars Andersson. Encouraging lstms to anticipate actions very early. In *IEEE International Conference on Computer Vision*, volume 1, 2017. 2

[3] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010. 5

[4] Matteo Bregonzio, Shaogang Gong, and Tao Xiang. Recognising action as clouds of space-time interest points. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1948–1955, 2009. 2

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4733, 2017. 2

[6] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge 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. 2

[7] Alireza Fathi and Greg Mori. Action recognition by learning mid-level motion features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2

[8] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016. 2

[9] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016. 2

[10] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005. 3

[11] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012. 2, 4

[12] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *IEEE International Conference on Computer Vision Workshop*, volume 2, page 4, 2017. 2

[13] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 2, 5

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. 2014. 2

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 3

[16] J. Hu, W. Zheng, L. Ma, G. Wang, J. Lai, and J. Zhang. Early action prediction by soft regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018. 1, 2, 3, 4, 5, 6, 7, 8

[17] Jian-Fang Hu, Wei-Shi Zheng, Jianhuang Lai, and Jianguo Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2186–2200, 2017. 2, 4, 6

[18] Jian-Fang Hu, Wei-Shi Zheng, Jiahui Pan, Jianhuang Lai, and Jianguo Zhang. Deep bilinear learning for rgb-d action recognition. In *European Conference on Computer Vision*, pages 346–362, 2018. 5

[19] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, 2017. 5

[20] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference*, pages 275–1, 2008. 2

[21] Yu Kong and Yun Fu. Max-margin action prediction machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1844–1858, 2016. 1, 3

[22] Yu Kong, Shangqian Gao, Bin Sun, and Yun Fu. Action prediction from videos via memorizing hard-to-predict samples. In *AAAI Conference on Artificial Intelligence*, 2018. 2, 3, 7, 8

[23] Yu Kong, Dmitry Kit, and Yun Fu. A discriminative model with multiple temporal scales for action prediction. In *European Conference on Computer Vision*, pages 596–611, 2014. 2, 8

[24] Yu Kong, Zhiqiang Tao, and Yun Fu. Deep sequential context networks for action prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1473–1481, 2017. 1, 2, 3, 4, 6, 7, 8

[25] Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. A hierarchical representation for future action prediction. In *European Conference on Computer Vision*, pages 689–704. 2014. 2, 3

[26] Kang Li and Yun Fu. Prediction of human activity by discovering temporal sequence patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1644–1657, 2014. 3

[27] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. In *International Joint Conference on Artificial Intelligence*, pages 2230–2236, 2017. 2, 4

[28] Shugao Ma, Leonid Sigal, and Stan 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. 6, 7

[29] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. 7

[30] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[31] Hossein Rahmani and Mohammed Bennamoun. Learning action recognition model from depth and skeleton videos. In *IEEE International Conference on Computer Vision*, 2017. 2

[32] Michael S Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *IEEE International Conference on Computer Vision*, pages 1036–1043, 2011. 2, 3, 8

[33] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACM international conference on Multimedia*, pages 357–360, 2007. 2

[34] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1010–1019, 2016. 4, 5, 6

[35] Amir Shahroudy, Tian-Tsong Ng, Yihong Gong, and Gang Wang. Deep multimodal feature analysis for action recognition in rgb+ d videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1045–1058, 2018. 2

[36] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 2

[37] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012. 2, 4

[38] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI Conference on Artificial Intelligence*, volume 4, page 12, 2017. 5

[39] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *IEEE International Conference on Computer Vision*, pages 4489–4497, 2015. 2

[40] Du Tran, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. Convnet architecture search for spatiotemporal feature learning. *CoRR*, abs/1708.05038, 2017. 2

[41] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, pages 3551–3558, 2013. 2

[42] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2017. 2

[43] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*, 2017. 2

[44] Zhang Zhang and Dacheng Tao. Slow feature analysis for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):436–450, 2012. 2

[45] Yu Zhu, Wenbin Chen, and Guodong Guo. Fusing spatiotemporal features and joints for 3d action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 486–491, 2013. 2

[46] Yi Zhu, Zhen-Zhong Lan, Shawn D. Newsam, and Alexander G. Hauptmann. Hidden two-stream convolutional networks for action recognition. *CoRR*, abs/1704.00389, 2017. 2