Analysis of Motion Sequence based on Iterative-transfer-learning

Yang Wang, Cheng Chen, Ke Yi, Tongxi Wang, YunCai Zhou, Hua Xiang

***Abstract*—Transfer learning can solve the problem of low recognition accuracy caused by dataset insufficiency. However, the improvement in performance for conventional transfer learning is limited. In this paper, we propose an iterative transfer learning (ITL) framework to solve the problem. Using a predetermined iteration strategy to perform ITL, the model with the best performance is selected to generate a new extended dataset. The standard dataset and the extended dataset are mixed for training in the next round. This type of training process fully demonstrates the effects of transfer learning and data amplification. The experimental results show that the ITL framework proposed in this paper improves the accuracy of the optimal model from 91.63% to 97.70%. The ITL framework has practical significance for improving model performance in small datasets. It is suitable for the analysis of action sequences on video streams with temporal characteristics and normative definitions.**

***Index Terms*—Transfer learning, iterative model, ITL, data amplification, motion sequence analysis.**

I. Introduction

As an extension of human vision, cameras are widely used in many fields such as bank security and traffic management [1]–[3]. At present, there are a large number of standard datasets on pedestrians and vehicles [4], [5], and intelligent monitoring is widely used in pedestrian and vehicle detection. However, due to the private nature of production materials and the confidentiality of production techniques, the analysis of standard industrial production lines does not have a standard dataset. The lack of a standard dataset has been an important bottleneck that affects the accuracy of motion sequence recognition [6].

To solve the problem of a lacking dataset, the three mainstream methods are the following. (1) Generate data by cropping, flipping, scaling, and adding noise to the original image. This method is relatively low-cost and relatively easy to implement. However, the generated image is prone to distortion, and the accuracy of the target detection algorithm is limited [7], [8]. (2) Use GAN-based image conversion to generate data. This method has a high calculation cost, is time-consuming, and has the problems of low image quality, lack of detail and realistic texture [9]–[11]. (3) Using transfer learning, pre-train on a large-scale general dataset, and then use the model transfer for recognition in a specific field. The iterative transfer learning(ITL) framework in this paper is a new method for data augmentation based on the third way, i.e., using the previous round of the transfer learning model to identify the video stream, and then generating a new dataset to expand the data. This method of generating data via the model and then using them to optimize the model greatly eases the problem of scarcity of datasets in specific fields, making it possible to use large amounts of data for deep learning training.

During the experiment, we divide the model training process into iteration processes. In each iteration, we improve the recognition accuracy of the model. With these multiple iteration processes, our model recognition accuracy rate achieves a good result. In our model training process, we performed a total of 3 iterations, and the final accuracy was 6% higher than the first round of model recognition.

The major contributions of this paper are summarized as follows:

1. We propose an iterative training framework suitable for fields with small sample datasets. Experiments show that the framework significantly improves the recognition accuracy.

2. A new data amplification method is proposed. With this method, we generate a large number of standard datasets in the industrial production field, thus solving the problem of dataset scarcity in the industrial production field.

3. We compare and analyze the performance of the models in different stages, and found the transfer learning model suitable for the motion analysis of time-series and normative videos.

II. Related Work

A. Transfer learning

The main purpose of transfer learning is to solve the problem of insufficient datasets in the specific field of interest. The concept of transfer learning includes source tasks and target tasks. If we define the tagged source tasks as D_s=$\{x_i, y_i\}_{i=1}^{n}$, and the unlabeled target tasks as D_t=$\{y_j\}_{j=n+1}^{n+m}$, then we use the knowledge in D_s to improve the accuracy of the prediction function in D_t, where the relation between D_s and D_t are not equivalent.

Transfer learning can be divided into four categories based on "the type of transfer": 1) the instance-based approach, 2) the feature-representation-based approach, 3) the parameter-based approach, and 4) the relational-knowledge-based approach. The first approach believes that part of the data in

Manuscript received November 11, 2020; revised May 14, 2021. This work is supported by the National Natural Science Foundation of China under Grant (No. 61703278).

Yang Wang is a postgraduate student in the School of Computer Science, Yangtze University, Jingzhou, Hubei, 434023 China. (e-mail:201872349@yangtzeu.edu.cn).

Cheng Chen is a postgraduate student in the School of Computer Science, Yangtze University, Jingzhou, Hubei, 434023 China. (e-mail:201871339@yangtzeu.edu.cn).

Ke Yi is a postgraduate student in the School of Computer Science, Yangtze University, Jingzhou, Hubei, 434023 China. (e-mail:201671360@126.com).

Tongxi Wang is a professor in the School of Computer Science at Yangtze University, Jingzhou, Hubei, 434023 China (Corresponding author, phone: 0086-139-9760-1255; e-mail: txwang@yangtzeu.edu.cn).

YunCai Zhou is a professor in the School of Computer Science at Yangtze University, Jingzhou, Hubei, 434023 China. (e-mail: 123474336@qq.com).

Hua Xiang is a lecturer in the School of Computer Science at Yangtze University, Jingzhou, Hubei, 434023 China. (e-mail: xianghua@yangtzeu.edu.cn).

the source domain can be reused by adjusting the weights. Weight adjustment and importance sampling are two main techniques used in the instance-based approach. In the second approach, the knowledge for cross-domain transmission is encoded as a feature expression of the learning. The goal is to train a "good" feature representation. In the third approach, the hyperparameters of the source task and the target task are assumed shareable, and the goal of model training is to obtain the hyperparameters in the source task. The fourth approach differs from the first three methods because it deals with the domains where the source domain is related to the target domain, and the relations between the data in the domains are transferred [12].

The typical application research of transfer learning includes text processing [13], sentiment classification [14]–[16], image classification [17]–[19], collaborative filtering [20], [21], etc.

We use a parameter-based approach, the COCO dataset as the source dataset and the actual image on the industrial production line as the target dataset. When the first round of transfer learning identifies the actions on the industrial production line, the initial parameters of the model are derived from the model parameters obtained based on the fine adjustment of the COCO images.

B. Action sequence analysis

In the process of the motion analysis of video actions, many different methods have been applied. According to the different modeling methods, the classification of motion analysis can be divided into four categories: (1) temporal templates [22]–[24], (2) state space [25], [26], (3) grammar model [27], [28],and (4) deep learning.

The motion analysis method based on temporal templates considers the motion characteristics of time and space at the same time, and recognizes the video as a whole [22]–[24]. The main research method is template matching [23], [29]–[32]. Although the computational complexity of template matching is low, the method performs well on simple static actions and is not applicable to complex actions. Moreover, the inconsistency of the motion interval of the same action affects the accuracy of recognition.

The state space-based method considers the sequence of changes in human behavior in time. This method uses dense optical flow features to describe video content and a motion boundary histogram to describe dense optical flow features [33], [34]. However, the state space-based method is complex for long-scale action, and the process of state transition is not suitable for the monitoring of uninterrupted surveillance video [25], [26]. The grammatical model-based motion analysis method uses grammar analysis in natural language processing as its key technology [27], [28]. Common grammar models are context-free grammar (CFG) and random CFG [35]–[37]. A grammatical model-based motion analysis method describes a human motion as a series of symbols, each of which is an atomic decomposition of an action [38], [39]. The motion recognition process needs to identify these atomic actions first. Consequently, the robustness of its spatial scale mainly depends on the underlying description, and the computational complexity is high.

The motion analysis method based on deep learning is the current mainstream method used in motion recognition. The existing deep learning-based algorithms can be divided into region-based methods and region-free methods.

The former uses the algorithm to generate a series of region proposals as samples, and then classifies the samples through a convolutional neural network. The latter does not generate region proposals, but directly converts the problem of target positioning into a regression problem. It is precisely because of the differences between the two methods that there are also differences in performance. The former is better in detection accuracy and positioning accuracy, while the latter is better in algorithm speed.

Representative methods of the former include Fast R-CNN [40] and Faster R-CNN [41]. Representative methods of the latter include SSD [42]and YOLO [43]. Due to the high requirements for accuracy in motion analysis, the deep learning model used in this paper mainly considers the region-based method.

III. The Proposed Method

The entire framework is shown in Fig. 1. As shown in the figure, the ITL framework consists of five modules, i.e., the generation of the standard dataset, the pre-training of the models, the segmentation of the surveillance video, the generation of the extended dataset, and the training of the iterative transfer learning models. The segmentation of the surveillance video is a prerequisite for generating the extended dataset; therefore, we introduce it in the third part (Generate extended dataset)

A. Generate the standard dataset

The standard dataset is extracted from a field video recording in a standardized operation scenario. Our action decomposition standard is to divide the motion sequences into $N(N \in N^+)$ standard actions. Each standard action is a class, and the i-th standard action is recorded as C_i. In the process of generating the standard dataset, we collect the raw video by repeatedly recording $M(M \in N^+)$ videos of motion sequences on each collection site. We use $R = \{R_1, R_2...R_i...R_M\}$ to indicate the M raw video, where R_i is a motion sequence consisting of $\{C_1, C_2...C_i...C_N\}$. After obtaining M videos of motion sequences, we cut each video according to the decomposition standard, i.e., cutting each R_i into video clips by classes, such that the resulting video clips only include a single class. The j-th video clip about the class C_i obtained from the k-th raw video is recorded as $V_{k_j}^{C_i}$, then the resulting M videos about the class C_i are $V_{1_j}^{C_i}, V_{2_j}^{C_i}, V_{3_j}^{C_i}, , , V_{M_j}^{C_i}$.

In this paper, our surveillance video contains a total of 7 standard actions(N = 7), and C_1, C_2, C_3, C_4, C_5, C_6, and C_7 correspond to the classes Qu-shang, Shuli, Anya, Fang-zhua-you, Suo, Cha4, and Bai-you, respectively. During the experiment, we collected 30 video clips for each class(M = 30). Taking the class "Suo" as an example, 30 video clips of this class are shown in Fig. 2.

After obtaining the video clips of each class, the standard dataset is labeled using the annotation tool VOTT(Visual Object Tagging Tool) with manual assistance. The format of the dataset is PASCAL VOC and the resolution of the image is 1280 * 720. Due to the inconsistent lengths of the different actions, the labeled standard data volume is

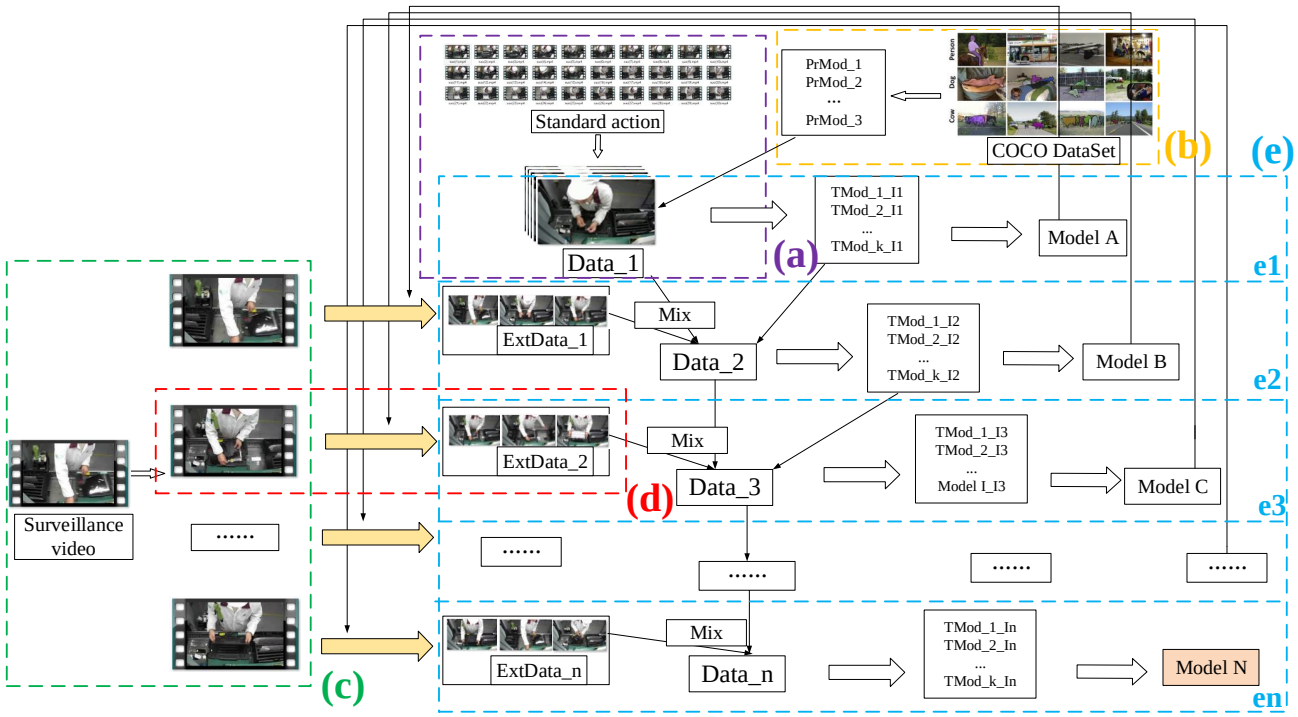

Fig. 1. ITL framework diagram. (a) Determine the standard actions and generate the standard dataset; (b) Pre-train the model on a large source dataset; (c) Segment the entire video into multiple small video parts; (d) Generate the extended dataset; (e) Mix the standard dataset with the extended dataset for iterative transfer training.

TABLE I
SAMPLE SIZE OF THE STANDARD DATASET.

	Anya	Bai-you	Fang-zhua-you	Cha4	Shuli	Qu-shang	Suo
Raw data	415	124	576	1290	2323	978	363
Data_1	415	372	400	400	400	400	363

unbalanced across categories. Considering that the imbalance of data will lead to larger training errors, we use a heuristic method to selectively extract and balance the original data. The standard dataset finally generated is Data_1. The number of labeled samples in each class is shown in Table 1.

B. Iterative transfer learning process

Conventionally, transfer learning only considers the single-round approach with a single model [44], [45]. In our ITL framework, we use multiple deep learning models for iterative transfer training. In the process of generating a new extended dataset, we choose the model that performs best in the previous round of transfer learning.

We use the first round of the model transfer I_1 as an example to describe an iterative process as follows. In the first round of iterative transfer learning, the COCO dataset is the source domain (denoted as $D_s^{I_1}$), the target domain (denoted as $D_t^{I_1}$) is Data_1. The training process in $D_s^{I_1}$ is the part(b) in Fig.1. The first round of target domain training uses $D_t^{I_1}$ shown in part(e1) in Fig.1. Data_1 is also the source domain in the second round(denoted as $D_s^{I_2}$). If our K pre-trained models are represented by PrMod_1, PrMod_2, ..., PrMod k, then after ignoring some details, the training processes of these K models can all be described using CNN models.

CNN is a deep neural network consisting of alternately stacked convolutional layers and pooling layers. The convolutional layer is used to extract features. The current neural layer is connected to the feature map of the previous layer through a convolution kernel that performs a convolution operation; then, biases are implemented to obtain the feature map of the current layer. The convolution kernel is shared by all neural units of the same feature map, that is, weight sharing. In this way, CNN greatly reduces the scale of the parameters.

The forward propagation process of the convolutional layer is as follows:

$$Z^l = a^{l-1} * W^l + b^l; \tag{1}$$

$$a^l = \sigma(Z^l). \tag{2}$$

where Z^l is the input of the convolutional layer in the l layer, a^{l-1} is the output of the $(l-1)$ layer, W^l is the weight, and b^l is deviation, and σ is the activation function.

In the process of backpropagation, to facilitate the derivation, the partial derivative of the error in the l layer is expressed as δ^l, and the partial derivative of the error in the $l-1$ layer is δ^{l-1}. Then the relation between δ^{l-1} and δ^l can be expressed as follows:

$$\begin{aligned}\delta^{l-1} &= \frac{\alpha J w,b}{\alpha z^{l-1}} = \frac{\alpha J w,b}{\alpha z^l}\frac{\alpha z^l}{\alpha a^{l-1}}\frac{\alpha a^{l-1}}{\alpha z^{l-1}} \\ &= \delta^l * rot180(W^l) \odot \sigma\prime(z^{l-1}).\end{aligned} \tag{3}$$

where $rot180(W^l)$ represents the rotation of the matrix W^l by 180 degrees.

In the process of backpropagation, the rules for updating the weights and biases of layer l are as follows:

$$\frac{\alpha J_{(w,b)}}{\alpha w^l} = \frac{\alpha J_{w,b}}{\alpha z^l}\frac{\alpha z^l}{\alpha w^l} = a^{l-1} * \delta_l; \tag{4}$$

Fig. 2. 30 video clips of "Suo".

$$\frac{\alpha J_{(w,b)}}{\alpha b^l} = \sum (\delta^l)_{u,v}. \tag{5}$$

To verify the effectiveness of the ITL framework and combine the advantages of each model, we select 4 models for iterative transfer. In this paper, PrMod_1, PrMod_2, PrMod_3 and PrMod_4, which represent the four commonly used pre-training models, faster_rcnn+inception, faster_rcnn+resnet101, rfcn+resnet101, and faster_rcnn + resnet50, respectively. During the pre-training process, the weights and biases in the CNN structure are continuously optimized and adjusted by training on the COCO dataset. After the first round of transfer learning I_1, the models that saved the adjustment parameters during the training of the dataset were named TMod_1_I1, TMod_2_I1, TMod_3_I1, and TMod_4_I1. In the transfer learning process, we use the model parameters obtained during the pre-training and modify the parameter settings of the last fully connected layer of the network to use the pre-training model. The schematic diagram is shown in Fig.3.

C. Generate the extended dataset

The extended dataset is obtained from surveillance video, which has the same shooting angle, height, and distance as the standard dataset. In theory, the length of the surveillance video we can obtain is infinitely long, and the length of the video is positively related to time, that is, L=$f(t)$, where t represents time. Unlike the generation process of standard datasets, the generation of extended datasets does not require the manual labeling of data. Its generation mainly goes through two steps. The first one is to segment the raw surveillance video into small video parts based on a time interval (e.g., 5 min), as shown in part(c) of Fig. 1. The second is to generate a new extended dataset by using the previous transfer learning model as shown in part(d) of Fig. 1. In step one, we segment a raw video with a length of L according to time. The process is R_{whole}=$R_{cut1} \bigcup R_{cut2} \bigcup ... \bigcup R_{cutK}$, the length of each video after segmentation is the same, and each video contains data of sufficient duration for the class with the least time.

The model with the highest testing accuracy is defined as the optimal model. The extended dataset is generated using the optimal model of the previous round of the training process, applying it to the segmented video R_{cutK} to detect the segmented video. After detection, the class C_i and the likelihood scores P_t^i of the corresponding frame in the video are generated. Take the process of generating the first batch of extended dataset as an example to illustrate the entire process. We select the optimal model in the first round of transfer learning to do the video target detection. After using our optimal model for target detection, from the t-th frame in R_{cut1}, we can obtain the data as $D(t)\_1 = \{(d_t^{C_1}, p_t^1), (d_t^{C_2}, p_t^2)...(d_t^{C_i}, p_t^i)\}$, where C_i is the class of an image; and p_i is the likelihood of belonging to the class C_i, where the value p_i is between 0 and 1.

TABLE II
SAMPLE SIZE OF THE EXTENDED DATASET.

	Anya	Bai-you	Fang-zhua-you	Cha4	Shuli	Qu-shang	Suo
Data_2	800	800	800	800	800	800	800
Data_3	1200	1200	1200	1200	1200	1200	1200

We set a threshold of p^0. When the value of p_t^{max}, where $p_t^{max} = max\{p_t^1, p_t^2...p_t^i\}$, greater than the threshold, we add it to the optional extended dataset. The data whose recognition probabilities are below the threshold p^0 are discarded. Then the data of the t-th frame image in the segmented video R_{cut1} generated by the iterative model is Ext(t)_1 = $(d_t^{C_i}, C_i)$. The data generated by the iterative model is not directly used for the next round of training. We select n samples that are closest to the category and add them to the extended dataset. The extended dataset formed by the final segmented video R_{cut1} is $ExtData\_1 = \{(d_t^{C_i}, C_i) | t \subset 1, 2, 3...T, i = 1, 2, 3...N\}$.

For each iterative transfer learning training, we select data similar to the quantity of the standard dataset, and mix the newly generated extended dataset with the previous round of data to generate the current round of data. Then the datasets for each iterative transfer learning training is Data_2 and Data_3. The quantity of data is shown in Table 2.

D. Overall process

The iteration process is divided into eight steps as follows:

Step 1: Generate the standard dataset Data_1: Determine the standard action of the motion sequence, and then generate the standard dataset Data_1 according to the standard action;

Step 2: Perform the first round of transfer learning: Use Data_1 as the dataset for the training models (PrMod_1, PrMod_2, PrMod_3 and PrMod_4), which are object_detection pre-training models from Tensorflow. Then the accuracy and recall rate of each model are recorded;

Step 3: Export the transfer learning models: Export the models from the first round of transfer training in Step 2 and name them TMod_1_I1, TMod_2_I1, TMod_3_I1 and TMod_4_I1. Select the model with the highest recognition accuracy as the optimal model and name it $Model\ A$.

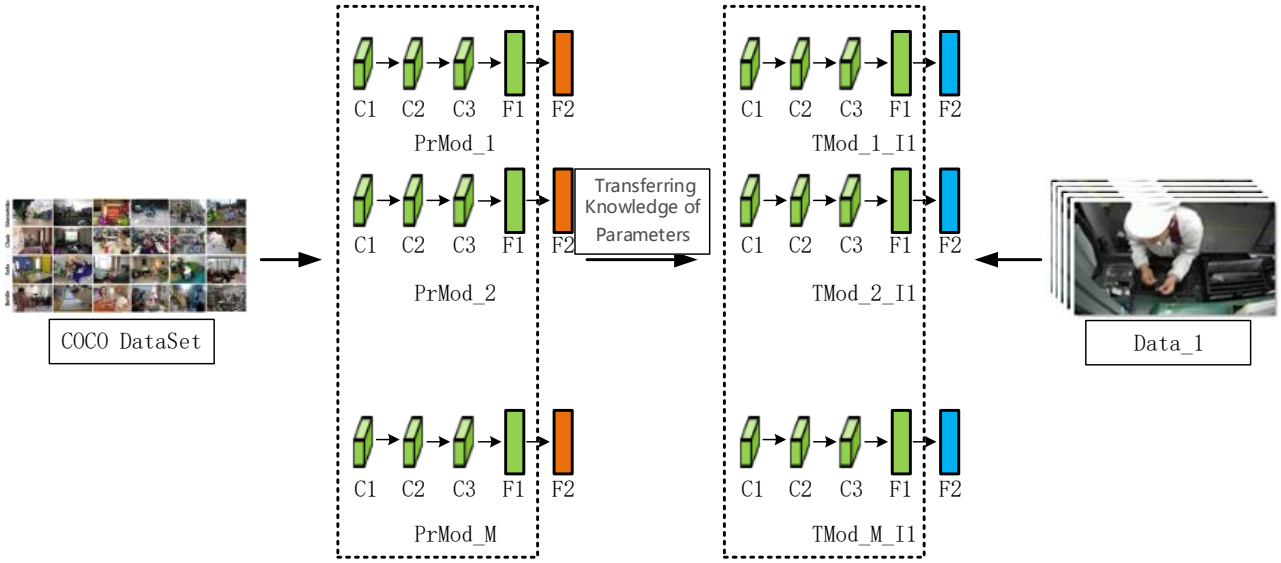

Fig. 3. Structure of the first round of iterative transfer training.

Use $Model\ A$ to detect the segmentation video R_{cut1}, and generate the extended dataset ExtData_1;

Step 4: Generate the extended dataset for the second round of transfer learning: Mix the datasets Data_1 and ExtData_1 to generate a second batch of training data Data_2;

Step 5: Repeat Step2 to Step4: Use TMod_1_I1, TMod_2_I1, TMod_3_I1 and TMod_4_I1 to perform a second iterative transfer training on Data_2, recording the accuracy and the recall rate. Export the model after the second round of training and name them TMod_1_I2, TMod_2_I2, TMod_3_I2 and TMod_4_I2. Select the optimal model, and name it $Model\ B$.

Step 6: Determine the iteration stop condition: Compare the accuracy of $Model\ A$ and $Model\ B$. If the accuracy rate of $Model\ B$ is greater than that of $Model\ A$, perform Step 2. If the accuracy rate of $Model\ B$ is less than that of $Model\ A$, or the training time exceeds the tolerance range, stop the iterative transfer training.

IV. Experiment and Analysis

A. Experimental equipment

The experiment was performed on two 32G memory alien Alienware machines with 64-bit Windows 10 and NVIDIA GTX 1070Ti dual GPU for the CNN calculations. The Software relies on the deep learning framework Tensorflow-GPU 1.5, on the NVIDIA parallel computing architecture CUDA9.0, and the deep learning GPU acceleration library cudnn V7.0.

B. Evaluation index

We evaluate the model using mAP values and recall values.

In machine learning, the common indicators for model evaluation are the accuracy and the recall. The accuracy determines the proportion of positive samples that are correctly classified in the detected results, and the recall determines the correct positive samples to account for the weight of all positive samples. The accuracy and recall can be expressed as follows:

$Recall = \frac{TP}{TP+FN}$

$Accuracy = \frac{TP+TN}{TP+FN+TN+FP}$

TP: Model predicts positive categories as positive.

FN: Model predicts positive categories as negative.

FP: Model predicts negative categories as positive.

TN: Model predicts negative categories as negative.

In model training, we hope for both greater accuracy and greater recall; however, the two are in contradiction in extreme cases. Generally, the precision-recall (PR) curve is used to measure the effect of the model, with recall as the horizontal axis and precision as the vertical axis. The optimization target requires the recall to increase and the precision to increase as well. The larger the area AP under the curve, the better the performance of the classifier. The mAP value is the average of all types of APs.

C. Experimental Results

1) Comparison of the accuracies of the three-round iterative models: We start training the 4 models: PrMod_1, PrMod_2, PrMod_3 and PrMod_4. The training is stopped when the model accuracy stops increasing.

When the first round of model training is completed, the trained models are exported and named TMod_1_I1, TMod_2_I1, TMod_3_I1 and TMod_4_I1. The accuracy and the recall of the four models are shown in Table 3 and Table 4. The data in the tables show that TMod_2_I1 performs better than the other three models. Therefore the model TMod_2_I1 based on the model PrMod_2 is our first-round optimal model, which is $Model\ A$ in Fig. 1.

We use $Model\ A$ to generate the first batch of extended dataset, ExtData_1. Then, ExtData_1 and Data_1 are mixed to form the second batch of standard dataset Data_2.

Then, TMod_1_I1, TMod_2_I1, TMod_3_I1 and TMod_4_I1 are used to perform a second round of iterative transfer training on Data_2, with a the training epoch equal

TABLE III
ACCURACY IN THE FIRST ROUND OF TRANSFER TRAINING, THE VALUE CHANGES WHEN TRAINING NUMBER CHANGES.

	10000	20000	40000	80000	120000	150000
TMod_1_I1	0.4658	0.5988	0.7836	0.8276	0.8492	0.8518
TMod_2_I1	**0.8145**	**0.8727**	**0.8994**	**0.9008**	**0.9102**	**0.9163**
TMod_3_I1	0.7757	0.8315	0.9079	0.9125	0.9149	0.9148
TMod_4_I1	0.7892	0.8491	0.8707	0.8973	0.9018	0.9064

TABLE IV
RECALL IN THE FIRST ROUND OF TRANSFER TRAINING, THE VALUE CHANGES WHEN TRAINING NUMBER CHANGES.

	10000	20000	40000	80000	120000	150000
TMod_1_I1	0.6007	0.6866	0.7774	0.8603	0.8806	0.8829
TMod_2_I1	**0.8663**	**0.8983**	**0.9259**	**0.9384**	**0.9355**	**0.9413**
TMod_3_I1	0.8099	0.8598	0.9323	0.9367	0.9383	0.9359
TMod_4_I1	0.8257	0.8824	0.9096	0.9243	0.9274	0.9340

to that in the first round. The recognition accuracy and the recall of the four models are shown in Table 5 and Table 6.

The data in these tables show that TMod_2_I2 performs better than the other three models. Therefore, the model TMod_2_I2 based on the model TMod_2_I1 is our second-round optimal model, which is $Model\ B$ in Fig.1.

We use $Model\ B$ to generate the second batch of extended dataset, ExtData_1. Then, ExtData_2 and Data_2 are mixed to form the third batch of standard dataset Data_3.

After the second round of model iteration is completed, the iterative models are exported and named TMod_1_I3, TMod_2_I3, TMod_3_I3, and TMod_4_I3. We then perform the third round of iterative transfer training on Data_3, with the number of trainings equal to that in the first round. The recognition accuracy and the recall of the four models are shown in Table 7 and Table 8.

The data in these tables show that TMod_2_I3 performs better than the other three models. Therefore, we export the trained model as our final model, which is $Model\ C$ in Fig. 1.

TABLE V
ACCURACY IN THE SECOND ROUND OF TRANSFER TRAINING, THE VALUE CHANGES WHEN TRAINING NUMBER CHANGES.

	10000	20000	40000	80000	120000	150000
TMod_1_I2	0.7678	0.7939	0.8372	0.8621	0.8678	0.8899
TMod_2_I2	**0.8907**	**0.9255**	**0.9408**	**0.9480**	**0.9485**	**0.9536**
TMod_3_I2	0.8091	0.8624	0.9360	0.9406	0.9424	0.9432
TMod_4_I2	0.8641	0.8943	0.9081	0.9237	0.9388	0.9390

TABLE VI
RECALL IN THE SECOND ROUND OF TRANSFER TRAINING, THE VALUE CHANGES WHEN TRAINING NUMBER CHANGES.

	10000	20000	40000	80000	120000	150000
TMod_1_I2	0.8106	0.8318	0.8714	0.8930	0.8984	0.9100
TMod_2_I2	**0.9159**	**0.9471**	**0.9613**	**0.9663**	**0.9664**	**0.9706**
TMod_3_I2	0.8363	0.8882	0.9573	0.9602	0.9604	0.9605
TMod_4_I2	0.8929	0.9187	0.9320	0.9465	0.9592	0.9601

TABLE VII
ACCURACY IN THE THIRD ROUND OF TRANSFER TRAINING, THE VALUE CHANGES WHEN TRAINING NUMBER CHANGES.

	10000	20000	40000	80000	120000	150000
TMod_1_I3	0.8036	0.8466	0.8710	0.8953	0.9011	0.9035
TMod_2_I3	**0.9024**	**0.9417**	**0.9554**	**0.9648**	**0.9715**	**0.9770**
TMod_3_I3	0.8451	0.8762	0.9465	0.9521	0.9575	0.9563
TMod_4_I3	0.8969	0.9237	0.9429	0.9477	0.9467	0.9570

TABLE VIII
RECALL IN THE THIRD ROUND OF TRANSFER TRAINING, THE VALUE CHANGES WHEN TRAINING NUMBER CHANGES.

	10000	20000	40000	80000	120000	150000
TMod_1_I3	0.8106	0.8318	0.8714	0.8930	0.8984	0.9100
TMod_2_I3	**0.9159**	**0.9471**	**0.9613**	**0.9663**	**0.9664**	**0.9706**
TMod_3_I3	0.8363	0.8882	0.9573	0.9602	0.9604	0.9605
TMod_4_I3	0.8929	0.9187	0.9320	0.9465	0.9592	0.9601

To better demonstrate the improvement of the model performance with our proposed ITL method, we show the mAP curves in the three rounds of training in Fig. 4, Fig. 5 and Fig. 6. The data in these figures show that, in the first round of model training, the accuracies of the four models are less than 95%. However, in the second round of model training, the accuracies of the three models are close to 95%. In the third round, there are three models with accuracy rates greater than 95%. The improvement of accuracy by ITL is significant.

As seen from the comparison, to make the recognition accuracy of the model reach 90%, the minimum training epochs of the first and second round are approximately 40,000 and 20,000, respectively. In the third round of iterations, two models already reach 90% accuracy when the number of training steps is less than 20,000 (using TMod_2_I3 or TMod_4_I3).

The mAP and recall curves of the optimal model for each round, $Model\ A$, $Model\ B$ and $Model\ C$ are shown in the Fig. 7 and Fig. 8.

The data show that the accuracy of the optimal model of each round increases, the order of the model accuracy rate is $Model\,A < Model\,B < Model\,C$. In addition, compared with the performance of the first round iterative transfer learning, the improvement of the performance of the last two rounds decreases. The $Model\ C$ with three rounds iterative transfer learning is our final model.

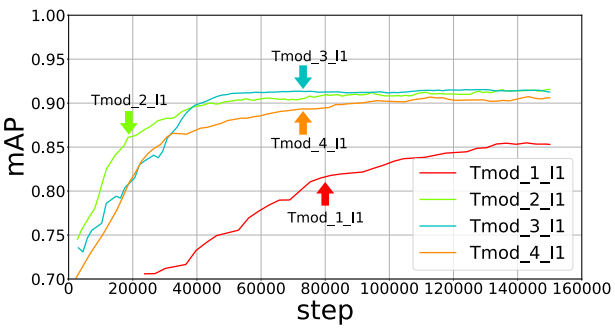

Fig. 4. The mAP curves of all models in the first-round model.

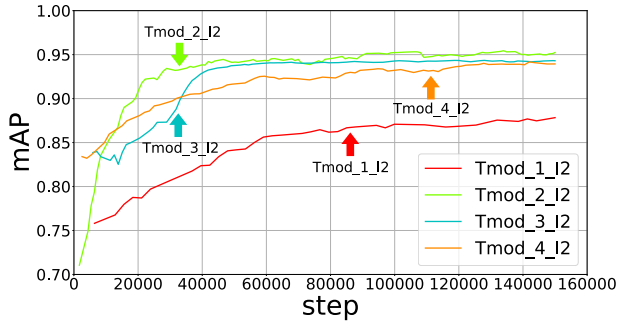

Fig. 5. The mAP curves of all models in the second-round model.

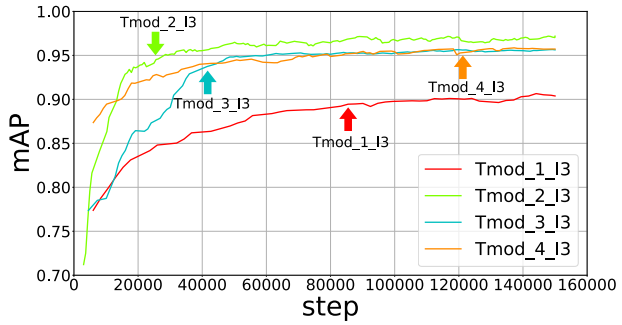

Fig. 6. The mAP curves of all models in the third-round model.

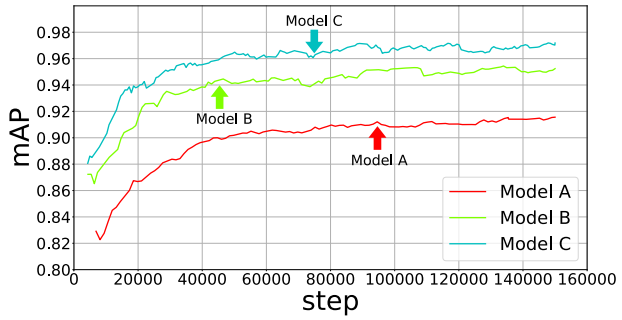

Fig. 7. The mAP curves of the three optimal models in three rounds of iterations

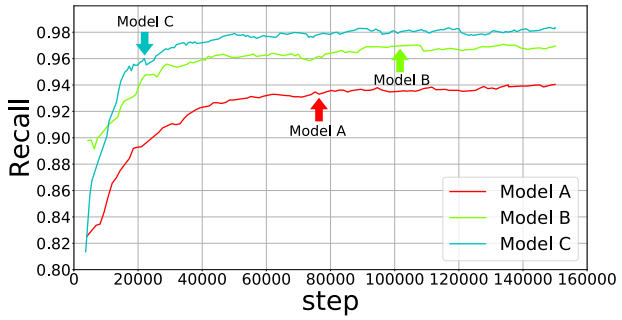

Fig. 8. The recall curves of the three optimal models in three rounds of iterations

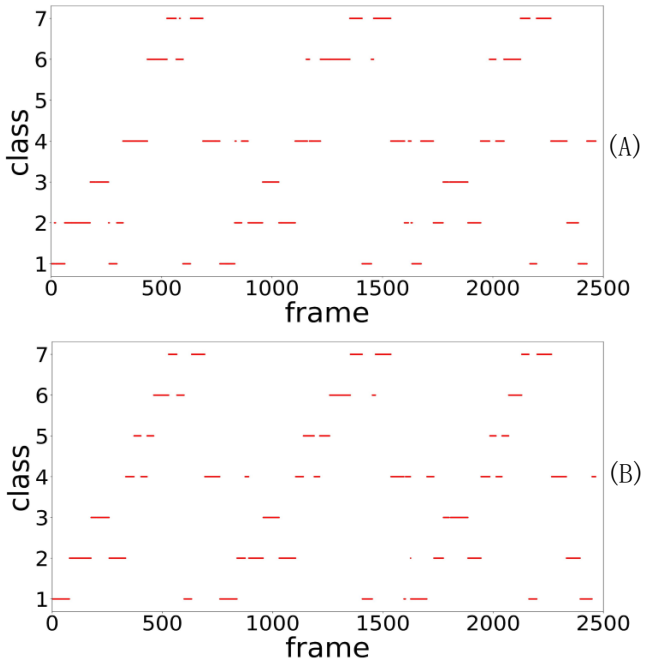

Fig. 9. (A) Action sequence diagram obtained from the first round of model recognition; (B) Action sequence diagram obtained from the second round of model recognition.

2) Comparison the of action sequence diagrams: The action sequence diagram is from our iterative transfer learning based on the frames' classes. The categories 1, 2, 3, 4, 5, 6 and 7 in the action sequence diagram correspond to the Qu-shang, Shuli, Anya, Fang-zhua-you, Suo, Cha4 and Bai-you.

A complete cycle of action sequences is $Qu-shang > Shuli > Anya > Shuli > Fang-zhua-you > Suo > Fang-zhua-you > Suo > Cha4 > Bai-you > Cha4 > Bai-you > Qu-shang > Fang-zhua-you.$

We identify a video containing multiple cycles of standard actions using the first-round optimal model $Model\ A$ (referred to as I1) and the second-round optimal model $ModelB$ (referred to as I2). This results in the two action sequence diagrams shown in Fig. 9, where (A) is from Model A and (B) is from Model B.

The comparison chart is shown in Fig.9.

The data show that when using the I2 for motion recognition, the recognition performance is better than when using the I1. The I1 is prone to misjudge two similar classes. The category Shuli is identified as the category Qu-shang, and the category Suo is identified as the category Cha4. In addition, the category Suo appears twice in a cycle, but in Fig.9, it is identified once in the first cycle, but not identified in cycle 2 or 3.

D. Discussion

The results of the experiment show that using our transfer learning improves the accuracy. The new extended dataset generated by the iterative model plays a key role in this improvement. To verify whether only by adding the extended dataset improved the accuracy or if both the dataset and the iterative model improve the performance, we use the same dataset, Data_3 (the total amount of data is 8600 examples), to compare the models with different iteration cycles.

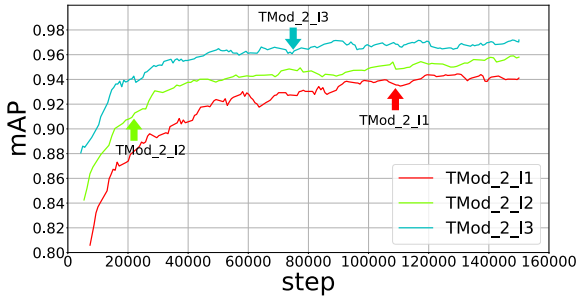

Fig. 10. Comparison of the model without iterative transfer training(TMod_2_I1) and the model with one iteration of transfer training(TMod_2_I2) on Data_3.

The three models compared are TMod_2_I1, TMod_2_I2 and TMod_2_I3. TMod_2_I1 has undergone one iteration of training, TMod_2_I2 has undergone two iterations of training and TMod_2_I3 has undergone three iterations of training.

The comparison chart is shown in Fig. 10.

The data show that model after iterative transfer learning training performs better than the model trained with the same amount of data but without iterative transfer learning.

The reason is determined by the network structure of resnet [46]. In resnet, the learning task changed from learning the basic mapping H(x) to learning the difference between x and H(x), i.e., the residual. To obtain H(x), we only need to add this residual to the output. Suppose the residual is F(x) = H(x) -x, then our network does not learn H(x) directly, but instead learn F(x) + x. Its implementation is to propose a method called residual structure block.

If we represent the output of the residual block as a^l, and w^l represents the weight value in the l-th residual block, then the convolutional layer and activation layer are represented by $F(a^l, w^l)$. We use a^{l+1} to represent the output of the l-th residual block, and a^{l+2} to represent the output of the (l+1)-th residual block. When $a^{l+1} = x^l + F(a^l, w^l)$, a^{l+2} can be expressed as follows:

$$a^{l+2} = a^l + F(a^l, w^l) + F(a^{l+1}, w^{l+1}). \tag{6}$$

When the network depth reaches the last layer L, a^L can be expressed as follows:

$$a^L = a^l + \sum_{i=l}^{L-1} F(a^i, W^i). \tag{7}$$

where the error is represented by J. In the network backpropagation, the gradient can be expressed as follows:

$$\frac{\alpha J}{\alpha a^l} = \frac{\alpha J}{\alpha a^L}\frac{\alpha a^L}{\alpha a^l} = \frac{\alpha J}{\alpha a^L}(1 + \frac{\alpha}{\alpha a^l}\sum_{i=l}^{L-1} F(a^i, w^i)). \tag{8}$$

The formula shows that in the process of the network's backpropagation to find the gradient, the weight and bias gradient of a certain layer need to be multiplied by partial derivatives. However, due to the existence of the residual parameter network, "+1" appears on the right side of the equation. In general, the latter term cannot always be "-1" for all a^l; Therefore, even if the weight is arbitrarily small, the gradient will not disappear. The characteristics of our dataset is that the background is complex and the target is single, and the size of the target is similar, and the differences are small. The experiments show that iterative transfer learning of the structure of resnet improves the accuracy of the model.

V. Conclusion

Based on the analysis of the motion characteristics of the industrial production line, we found a method suitable for video sequence analysis with temporal characteristics and normative definitions. We propose an ITL framework, which greatly improves the training performance of deep learning models. During the experiment, we mixed the newly generated extended dataset with the original small sample data to form the training dataset. The experimental results show that the ITL framework has a good effect on the recognition of motion sequences. With ITL, deep learning enters into the field of industrial production with less benchmark data. Furthermore, ITL has a wide range of applications in sequence recognition with data having timing and action specifications.

References

[1] X. Li, M. Ye, Y. Liu, F. Zhang, D. Liu, and S. Tang, "Accurate object detection using memory-based models in surveillance scenes," *Pattern Recognition*, vol. 67, pp. 73–84, 2017.

[2] Z. Liu, D. Luo, Y. Wang, L. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and T. Lu, "TEINet: Towards an Efficient Architecture for Video Recognition," 2019.

[3] V. Petrov, S. Andreev, M. Gerla, and Y. Koucheryavy, "Breaking the Limits in Urban Video Monitoring: Massive Crowd Sourced Surveillance over Vehicles," *IEEE Wireless Communications*, vol. 25, no. 5, pp. 104–112, 2018.

[4] M. Wang, W. Li, and X. Wang, "Transferring a generic pedestrian detector towards specific scenes," 06 2012, pp. 3274–3281.

[5] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[6] M. Romero, Y. Interian, T. Solberg, and G. Valdes, "Training Deep Learning models with small datasets," *arXiv: Learning*, 2019.

[7] M. G. Agnieszka Mikolajczyk, "Data augmentation for improving deep learning in image classification problem," *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, pp. 117–122, 2018.

[8] J. Salamon and J. P. Bello, "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[9] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," pp. 5967–5976, 2017.

[10] T. Kim, M. Cha, H. Kim, J. Lee, and J. Kim, "Learning to Discover Cross-Domain Relations with Generative Adversarial Networks," 2017.

[11] Q. Mao, H. Lee, H. Tseng, S. Ma, and M. Yang, "Mode Seeking Generative Adversarial Networks for Diverse Image Synthesis," pp. 1429–1437, 2019.

[12] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1345–1359, 2010.

[13] R. Raina, A. Y. Ng, and D. Koller, "Constructing informative priors using transfer learning," in *Proceedings of the 23rd International Conference on Machine Learning*. ACM, 2006, pp. 713–720.

[14] J. Blitzer, M. Dredze, and F. Pereira, "Domain adaptation for sentiment classification," in *45th Annv. Meeting of the Assoc. Computational Linguistics (ACL'07)*.

[15] T. Li, Y. Zhang, and V. Sindhwani, "A Non-negative Matrix Tri-factorization Approach to Sentiment Classification with Lexical Prior Knowledge," in *ACL*, 2009.

[16] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in *Proceedings of the 19th International Conference on World Wide Web*. ACM, 2010, pp. 751–760.

[17] P. Wu and T. G. Dietterich, "Improving SVM accuracy by training on auxiliary data sources," 2004.

[18] W. Dai, Y. Chen, G.-R. Xue, Q. Yang, and Y. Yu, "Translated Learning: Transfer Learning across Different Feature Spaces," in *NIPS*, 2008.

[19] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng, "Self-taught learning: Transfer Learning from Unlabeled Data," in *ICML '07*, 2007.

[20] W. Pan, E. Xiang, and Q. Yang, "Transfer Learning in Collaborative Filtering with Uncertain Ratings," in *AAAI*, 2012.

[21] B. Cao, N. Liu, and Q. Yang, "Transfer Learning for Collective Link Prediction in Multiple Heterogenous Domains," in *ICML*, 2010.

[22] A. F. Bobick and J. W. Davis, "The Recognition of Human Movement Using Temporal Templates," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 3, pp. 257–267, 2001.

[23] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2. IEEE, 2005, pp. 1395–1402.

[24] J. Yuan, Z. Liu, and Y. Wu, "Discriminative subvolume search for efficient action detection," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 2442–2449.

[25] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden Markov models for complex action recognition," in *CVPR*, vol. 97, 1997, p. 994.

[26] V. Pavlovic, B. J. Frey, and T. S. Huang, "Time-series classification using mixed-state dynamic Bayesian networks," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, vol. 2. IEEE, 1999, pp. 609–615.

[27] M. S. Ryoo and J. K. Aggarwal, "Recognition of Composite Human Activities through Context-Free Grammar Based Representation," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1709–1718.

[28] Z. Si, M. Pei, B. Yao, and S. Zhu, "Unsupervised Learning of Event AND-OR Grammar and Semantics from Video," pp. 41–48, 2011.

[29] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in *null*. IEEE, 2003, p. 726.

[30] Z. Jiang, Z. Lin, and L. Davis, "Recognizing Human Actions by Learning and Matching Shape-Motion Prototype Trees," *IEEE_J_PAMI*, vol. 34, no. 3, pp. 533–547, 2012.

[31] Z. Jiang, Z. Lin, and L. S. Davis, "A Tree-Based Approach to Integrated Action Localization, Recognition and Segmentation," in *ECCV Workshops*. Springer, 2010, pp. 114–127.

[32] M. S. Ryoo and J. K. Aggarwal, "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities," in *ICCV*, vol. 1. Citeseer, 2009, p. 2.

[33] S. M. O'Rourke, I. Herskowitz, and E. K. O'Shea, "Yeast go the whole HOG for the hyperosmotic response," *Trends in Genetics*, vol. 18, no. 8, pp. 405–412, 2002.

[34] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[35] P. Natarajan and R. Nevatia, "View and scale invariant action recognition using multiview shape-flow models," *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[36] M. S. D. Manual, "Motorola Semiconductor Products Inc," *Phoenix, AZ*, 1989.

[37] "Synthetic structure of industrial plastics (Book style with paper title and editor), author=Young, GO and Peters, J, year=1964, publisher=Plastics."

[38] R. Nevatia, T. Zhao, and S. Hongeng, "Hierarchical Language-based Representation of Events in Video Streams," in *2003 Conference on Computer Vision and Pattern Recognition Workshop*, vol. 4. IEEE, 2003, pp. 39–39.

[39] M. Ryoo and J. Aggarwal, "Stochastic Representation and Recognition of High-Level Group Activities," *International Journal of Computer Vision*, vol. 93, no. 2, pp. 183–200, 2011.

[40] R. B. Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015.

[41] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE_J_PAMI*, vol. 39, no. 6, pp. 1137–1149, 2017.

[42] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. Berg, "SSD: Single Shot MultiBox Detector," 2016.

[43] J. Redmon, S. Divvala, R. B. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016, pp. 779–788.

[44] V. Nair and J. J. Clark, "An unsupervised, online learning framework for moving object detection," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2. IEEE, 2004, pp. II–II.

[45] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-Supervised Self-Training of Object Detection Models." *WACV/MOTION*, vol. 2, 2005.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," pp. 770–778, 2016.

Yang Wang received a B.S. degree in software engineering from Yangtze University, Jingzhou, Hubei, China, in 2016, and is currently pursuing her M.S. degree in Computer Science at Yangtze University. She is in her second year of graduate school and will graduate in 2021. Her research interests include target recognition and classification and action sequence recognition.

Cheng Chen received a B.S. degree in software engineering from Yangtze University, Hubei, China, in 2018. She is currently pursuing her M.S. degree in Computer Science at Yangtze University, Jingzhou, Hubei, China. Her current research interests are computer vision, especially image processing, object detection and recognition, and model compression.

Ke Yi received a B.S. an M.S. degree from Yangtze University, Hubei, China. She received a Bachelor of Science degree in Mathematics and Applied Mathematics and a Master of Science degree in Software Engineering at Yangtze University, China in 2015 and 2019, respectively. Her current research direction is deep learning, which mainly includes computer vision and model compression acceleration.

Tongxi Wang obtained his B.S. and M.S. degrees in science from Chengdu University of Technology in Chengdu, China, in 1994 and 2007, respectively. He is currently an associate professor in the Department of Software Engineering at Yangtze University. He is the director of the Department of Software Engineering at the School of Computer Science and a doctoral and master's thesis reviewer at the Degree Center of the Ministry of Education. He is also a director of the Big Data Alliance of Chinese Universities and a member of the China Artificial Intelligence Association (CAAI). His current research interests are big data and artificial intelligence technologies, and intelligent computing.

Yuncai Zhou received his B.S. degree from Jianghan Petroleum Institute in 1981 and his M.S. degree in basic mathematics from Yunnan University in 1986. He is a professor in the School of Computer Science at Yangtze University. His research interests are algorithm design and research, computational theory, and software implementation.

Hua Xiang received his B.S degree from Yangtze University and a Master's degree from Wuhan University, in 2002 and 2009, respectively. He is currently a lecturer in the School of Computer Science at Yangtze University. His current research interests include artificial intelligence, software engineering, and big data technologies.