ISA Transactions 125 (2022) 459–473

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Practice article

A temporal–spatial attention-based action recognition method for intelligent fault diagnosis

Wentao Luo [a], Jianfu Zhang [a,b,*], Pingfa Feng [a,b,c], Dingwen Yu [a], Zhijun Wu [a]

[a] Beijing Key Lab of Precision/Ultra-precision Manufacturing Equipments and Control, Department of Mechanical Engineering, Tsinghua University, Beijing, 100084, China
[b] State Key Laboratory of Tribology, Department of Mechanical Engineering, Tsinghua University, Beijing, 100084, China
[c] Division of Advanced Manufacturing, Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China

ARTICLE INFO

Article history:
Received 13 October 2020
Received in revised form 30 June 2021
Accepted 30 June 2021
Available online 3 July 2021

Keywords:
Temporal-attention model
Spatial-attention model
Long-short term model
Video fault diagnosis

ABSTRACT

The intelligent fault diagnosis of video data has become a demanding task in industrial applications. However, existing models require expensive computational cost and memory demand, which makes this technology applied in factories impossible. To address this problem, a temporal–spatial attention-based action recognition method (TARM) integrating TAB (temporal-attention-based frame splitting model), SAB (spatial-attention-based agent focusing mode) and LSB (long-short term feature learning mode) is proposed. TAB first extracts important frames from raw videos. Then, SAB refines video data by reinforcing their essential features and weakening unnecessary features. Furthermore, LSB monitors action type of video data by establishing recurrent convolutional architectures. Finally, the performance of TARM in terms of training time and fault diagnosis accuracy are validated by comparing with six state-of-the-art video diagnosis methods.

1. Introduction

With visual devices replacing human eyes as the main means of monitoring, video diagnosis technology plays an essential role in versatile domains, such as traffic regulation, security alert, manufacturing supervision and so on. Video diagnosis is a 3D data processing technology that determines the type of object action by analyzing video data. Unlike image diagnosis methods, video diagnosis technologies are becoming big difficulties and challenges in action diagnosis domain as video data, consisting of countless frames with temporal features, place a greater demand on computing power and storage capacity of a computer. Even so, since rich process information are included in video data, it makes the video diagnosis be closer to human diagnosis, thus its decision-making results are more convincing than signal diagnosis (1D data) or image diagnosis (2D data).

The video diagnosis technology is mainly divided into two sub-steps, appearance recognition and motion estimation. The appearance recognition focuses on learning spatial feature of one single frame, while the motion estimation focuses on learning temporal variation feature of consecutive frames. As the motion estimation method need to track the dynamic information from video data, it has become the core and challenging technology. The optical flow method is a widespread and reliable method to extract dynamic information from data [1,2], which uses the mode motion speed in the time-varying image to represent the mode motion information. Dense Trajectories (DT) method [3] uses optical flow fields to obtain video sequence trajectories on multiple spatial scales, after which it utilized the support vector machine method to achieve video classification. DT methods obtained great performance on video action recognition before the advent of deep learning methods and the Improved Dense Trajectories (IDT) [4] increased the classification accuracy of the DT method in the action recognition benchmarks UCF50 from 84.5% to 91.2%. However, the DT method and IDT method need to calculate the optical flow field, which makes the calculation process slow.

With the emergence and development of deep learning methods, an increasing number of scholars make contributions to improving the classification accuracy in video diagnosis tasks. The Two Stream Network method is the first application of deep learning methods on video diagnosis technology, which used a pair of parallel convolutional networks (ConvNets) to extract the temporal and spatial features of the video data respectively and fused the two-flow direction results to obtain the final classification result. It is noted that the ConvNets used are 2D shallow convolutional networks. To further enhance the classification accuracy of video diagnosis technology, the deep convolutional neural network was used as the two-stream architecture of the Two

* Corresponding author at: Beijing Key Lab of Precision/Ultra-precision Manufacturing Equipments and Control, Department of Mechanical Engineering, Tsinghua University, Beijing, 100084, China.
E-mail address: zhjf@tsinghua.edu.cn (J. Zhang).

https://doi.org/10.1016/j.isatra.2021.06.041
0019-0578/

Stream Network, and the classification accuracy was increased to 92.5% [5] in the benchmark dataset UCF101.

The above-mentioned deep learning video diagnosis technology used the mainstream 2D convolutional layer as the basis of the model structure, which only focused on appearances and short-term motions of video data, thus lacking the ability to capture long-range temporal features. To address this problem, Temporal Segment Network (TSN) [6] method was proposed, which was designed for capturing long-range temporal features of video representation by adopting sparse temporal sampling strategy. The TSN increased the classification accuracy of the benchmark data set UCF101 to 94.2%. Since then, researches focusing on learning long-term features of data has been deeply explored in the field of video diagnostic analysis.

It can be seen that the above methods all use two-stream structure to analyze the temporal and spatial flow respectively. To simplify the model structure, 3D ConvNets (C3D) [7], consisting of $3 \times 3 \times 3$ convolution kernels, are built to extract the temporal and spatial features of multi-images simultaneously. However, as the C3D model expands the dimensionality of the convolution kernel, the number of parameters to be trained is also expanded several times, resulting in reduced training efficiency. Inspired by the state-of-art method Resnet in image recognition, some scholars proposed Pseudo-3D Residual Networks (P3D) [8] by combining residual block with C3D model to reduce the parameters and structure complexity of the C3D. To further reduce the parameters of a C3D model, a R(2 + 1)D model [9] mixing both 2D and 1D networks was proposed and proved to be the baseline method in video diagnosis field. Moreover, some tricks such as TSM [10] and attention pooling block [11] in C2D and C3D are also presented and show great potential on reducing compute and network complexity.

To sum up, the researches on the video diagnosis technology has been continuously updated, and it has achieved a good performance with regard to the accuracy in experiment. However, the following problems are still faced in the application of video diagnosis engineering,

(1) Large parameters of the feature learning model often lead to over-fitting problems and high training time, which hardly satisfy applied demand in industry.
(2) Low quality video data with massive invalid frame included affects the speed of model convergence and the final learning performance.
(3) Less attention is paid on fine-grained object or feature action recognition in video, thus tiny object motions are often ignored by existing models.
(4) There is a trade-off between the ability to learn the short-term and long-term temporal feature for video diagnosis model.

To solve above-mentioned problems, a temporal–spatial attention-based action recognition method (TARM) is proposed, which consists of the temporal-attention-based frame splitting mode (TAB), the spatial-attention-based agent focusing mode (SAB) and the long-short term feature learning mode (LSB). The TAB mode is used to extract important frames from long and raw video data, which provides an efficient way to form high quality video data with less invalid frames from a raw video. The SAB mode is built to map one single frame into one feature space in which fine-grained object features can be strengthened, thus avoiding the negative impact of noise or unimportant objects in a frame on the video diagnosis. The LSB mode is put forward to avoid the overfitting and the trade-off problems by combining an improved C3D model to learn the short-term feature and a recurrent neural network to learn the long-term feature in video feature extraction process.

2. Related works

Top performance of methods for video diagnosis use rich tricks to capture spatial–temporal features in video input, such as two-stream networks, 3D convolution networks and recurrent networks.

2.1. Two stream networks

Two stream architectures are typically consisted of two parallel networks named spatial stream and temporal stream respectively. In detail, spatial stream takes as input images to learn their appearance features, and temporal stream takes as input optical flow to learn their motion features. Then, the output from two streams should be fused to best discriminate spatial–temporal feature of each video input comprehensively [5]. Although many different variants of two stream models have been proposed and studied, the optical flow estimation could not be ignored as its significance on improving classification accuracy in two stream networks for video diagnosis domain. However, the optical flow estimation task is extremely difficult as predicting generic optical flow requires a sufficiently large training set, thus leading to long running time for video diagnosis. Several machine learning methods have been applied to optical flow, such as Gaussian scale mixture [12], restricted Boltzmann machines [13] and synchrony autoencoder [14], which improved the calculation efficiency of optical flow to a certain extent. To sum up, optical flow could highly increase the performance of video diagnosis, but it also increases the running time and decreases the algorithm efficiency of two stream networks [15].

2.2. 3D convolution networks

3D convolution neural networks (3D CNNs) is proposed to add temporal feature extraction function to 2D CNNs for video analysis. As spatiotemporal feature can be learned by 3D CNNs, it can be seen 3D CNNs lead to strong video diagnosis and action recognition performance while training on a large-scale video data [7]. Although 3D CNNs show great potential on video diagnosis and other extended domains such as action detection, video captioning [16] and medical image segmentation [17], it is still computationally intensive at training and not easy to scale up for testing on big data. Therefore, how to reduce the large parameters of 3D CNNs is a hot topic for researchers. At present, the P3D model [8] and the R3D model [9] are two light weight and advanced models. In these new models, a $3 \times 3 \times 3$ CNN kernel is replaced with several $1 \times 3 \times 3$ CNN kernel and $3 \times 1 \times 1$ CNN kernel, and the new kernels were proved to have a better performance and efficiency than the previous one. Besides, 3D Channel-Separated-Network (CSN) [18] is also a method to reduce the complexity of C3D, where all convolution blocks are factorized into either depth-wise $3 \times 3 \times 3$ and pointwise $1 \times 1 \times 1$ convolutions. To further decrease the need of C3D, a lightweight Gated-Shift Module (GSM) [19] is proposed to only apply a 2D-CNN with gated networks (simplified C3D) into a highly efficient feature extractor. By reviewing this state-of-the-art C3D variants, it can be concluded that scholars are trying best to reduce the expensive computation of traditional C3D by redesigning this structure. However, C3D is good at learning spatial features but has short receptive field on memorizing the temporal features due to the common size of convolutional kernel is 3. Therefore, a temporal-attention structure is indispensable for temporal feature extraction.

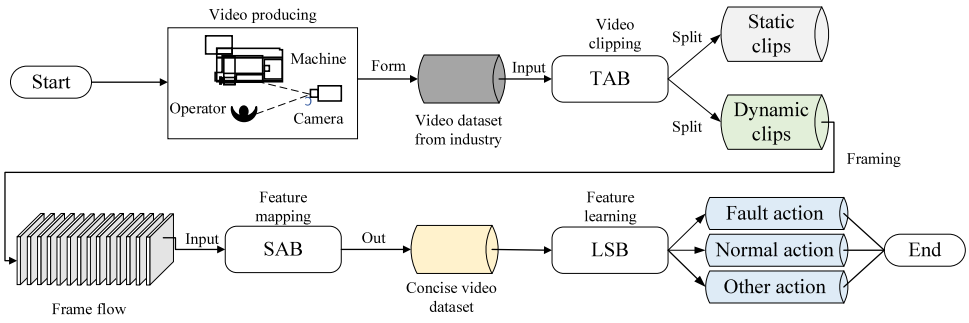

Fig. 1. The TARM flow chart.

2.3. Recurrent networks

Recurrent networks are well-known methods for extracting temporal features of long sequence data. To fully utilize the spatial feature extraction ability of CNNs and long temporal feature extraction ability of recurrent networks, a Long-term Recurrent Convolutional Networks (LRCNs) [20] is proposed for specific video activity recognition. In addition, Gated-Recurrent-Unit Recurrent Networks (GRUs), concise and high performance of recurrent network are added to the last layer of deep CNNs to reduce complexity of the model units and enforce sparse connectivity of inner structures [21]. Moreover, Long-Short Term Memory (LSTM) networks, classic recurrent networks are combined with auto-encoder methods to complete several unsupervised tasks such as classification or prediction tasks. Although LTSM has good capability on learning temporal feature of data, the low training efficiency is still its main drawback. Therefore, some attention methods are used in LSTM to address this problem. Convolutional LSTM (ConvLTSM) [22] and VideoLSTM [23] both extend the fully connected LSTM (FC-LSTM) with convolutional structures to maintain the spatial feature as much as possible, which are able to capture spatiotemporal correlations better and consistently outperforms FC-LSTM in tests. Moreover, VideoLSTM uses the flow images as motion-attention guides to help generate the attention maps while using RGB appearance input. The above LSTM variants perform well on several open-source datasets. Long Short-Term Attention (LSTA) [24] extends LSTM with pooling RNN cell and output gating modules to realize smooth attention tracking and high-capacity information storing. Moreover, a Global Context-Aware Attention LSTM (GCA-LSTM) [25] shows robust performance on selectively focusing on the informative joints in each frame of the skeleton sequence by feeding the global contextual information into all steps. In the complicate video captioning task, Spatio-Temporal and Temporo-Spatial (STaTS) attention-based language model [26] was proposed to build reliable relationships between captions and videos, which inspires me to propose a light-weight, attention-based and industry-oriented model in the intelligent diagnosis task.

To sum up, each tool method of video diagnosis has its own advantages and is suitable for different types of input video data. Competing other methods in benchmark dataset may not mean it can behave high performance on another dataset with diverse types. In this paper, engineering operating video data was collected and analyzed, and a comprehensive temporal–spatial feature extraction model was proposed to diagnose engineering data autonomously with high efficiency and effect.

3. Model establishing

3.1. Architecture of TARM

TARM, a temporal–spatial attention-based action recognition method, consists of TAB, SAB and LSB models. The main purpose of TARM is to improve the efficiency and performance of video diagnosis methods so that it can be applied to real industry and factories where near real-time and high accuracy diagnosis need to be satisfied. The dataflow of TARM as shown in Fig. 1. The three sub-models achieve this goal from vary perspective. More specifically, TAB aims at filtering out static and redundant video contents so as to form a high-quality video dataset for training in the next step, which reduces the data size from temporal perspective. Then SAB targets at mapping each frame of the video in the dataset to a feature space where desired features are reinforced, and other irrelevant features are weakened, which reduces the data size from spatial perspective. Finally, LSB focuses on learning the long and short-term features of video from the dataset to complete video diagnosis task, which improves the model's ability to sufficiently extract features and enhance the performance of final model by both paying attention to long-term and short term of data features.

3.2. Temporal-attention-based frame splitting mode (TAB)

The TAB aims at extracting frames that are helpful for feature learning from original video data, reducing the negative impact of long invalid segments of original video on learning performance. If a video clip without motion for a relatively long time, it can be defined as an invalid segment. This part focuses on the temporal dimension of video data and calculates the optical flow matrix for each frame to act as an indicator to determine whether the frame has motion or not.

To visualize the optical flow, the optical flow matrix is mapped from Cartesian coordinates to polar coordinates to form an HSV image. As ranges of the three channels of HSV images are vary ($h \in [0, 180]$; $s \in [0, 255]$; $v \in [0, 255]$), the HSV color space is converted to BGR color space to conveniently calculate three-channel optical flow matrix value [27]. The ranges of each value of BGR are all in the interval of [0, 255]. Then, the BGR image is converted to the grayscale image, then the average optical flow of each frame can be computed [28].

Due to the interference of the motion noise, the average value of the optical flow directly solved by the above method cannot accurately reflect the motion feature of the frame. More specifically, Fig. 2 shows gray images of the optical flow result from a static image and a dynamic image respectively. Fig. 2(a) is a static frame. Fig. 2(b) is the optical flow image of the static frame, and the random noise are filled in this image. (The average optical flow value is around 3.751.) Fig. 2(c) is a dynamic frame, and an operator is fastening bolt into a part. Fig. 2(d) is the optical flow image of the dynamic frame, in which no obvious noise is shown, and fingers motion is clear to see. (The average optical flow value is about 3.896.) It can be seen in the figure that the average optical flow of the static frame is similar to the dynamic one, which is not true in reality. Therefore, even though the optical flow method

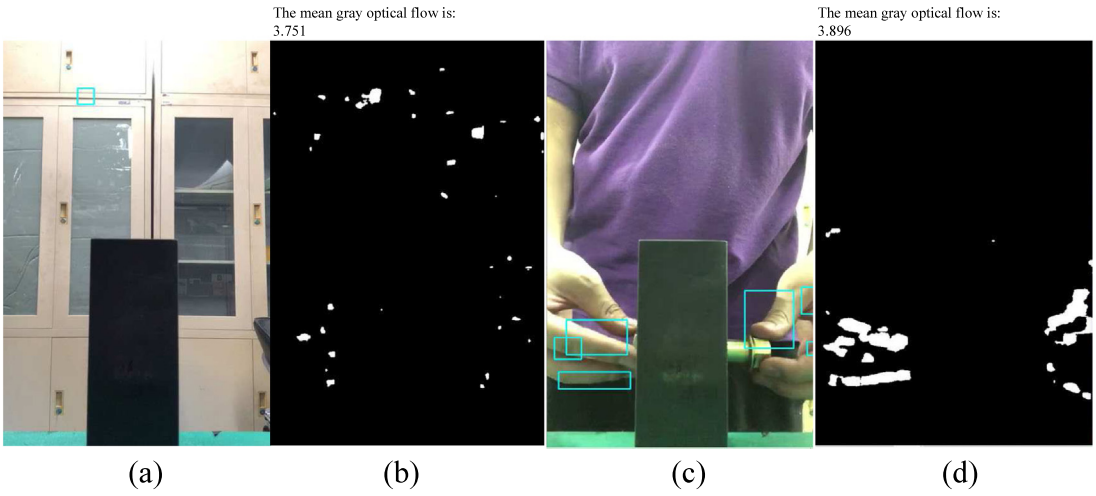

Fig. 2. The optical flow images in static and dynamic images.

can reflect the motion feature of a frame, static frames are easily to be considered as dynamic frames due to the motion noise.

In order to improve the frame motion discrimination capability by the average optical flow, this paper introduces the noise judgment method based on the partition idea to reduce the noise effect on average optical flow calculation.

The noise judgment method based on the partition idea uses the random and scattered characteristics of the noise distribution to identify the noisy optical flow frame. The noise is usually caused by the instability of the photographic equipment and uncontrollable ground vibration, because ground vibration will cause the dynamic vibration of the photographic equipment, which could make photographic equipment mistake static light and shadow for noise. Considering the randomness of ground vibration frequency and the difference of resonance frequency of each element, we assume that the appearance of noise obeys Gaussian distribution. Based on this assumption, we divide the optical flow image into five blocks named I–V respectively, as shown in Fig. 3(a). After dividing blocks, it can be seen that each partition of I–V of the image with the obvious noise feature are covered with optical flows regularly, as shown in Fig. 3(b). It can also be discovered that the optical flow of image with less noise covers only 1–2 blocks and the optical flows are concentrated in a specific area which can be defined as the valid optical flow, as shown in Fig. 3(c).

Specifically, the probability of a noise point appearing in a close-to-center area (block V) is slightly greater than far-to-center area (block I, II, III, IV). Moreover, the probability of noise points appearing in the five divided regions is not 0 and the difference is small. To distinguish a frame with the random noise, noise judgment constraints are defined as follow.

White noise if

$$\begin{cases} \overline{F}_I, \overline{F}_{II}, \overline{F}_{III}, \overline{F}_{VI}, \overline{F}_V > 0 \\ \max(\overline{F}_I, \overline{F}_{II}, \overline{F}_{III}, \overline{F}_{VI}, \overline{F}_V) - \min(\overline{F}_I, \overline{F}_{II}, \overline{F}_{III}, \overline{F}_{VI}, \overline{F}_V) < \varepsilon \\ \overline{F}_V \geq \max(\overline{F}_I, \overline{F}_{II}, \overline{F}_{III}, \overline{F}_{VI}) \end{cases} \quad (1)$$

where, $\overline{F}_I, \overline{F}_{II}, \overline{F}_{III}, \overline{F}_{VI}, \overline{F}_V$ are the average optical flow value in five blocks respectively (I, II, III, IV, V), ε represents the upper limit of the difference between the maximum average optical flow and the minimum average optical flow of the five partitions of the white noise image. It should be noted that all three inequalities in formula (1) must be satisfied before the image can be judged as a noise image.

By doing experiments, we found this method can effectively recognize noisy frame. It can also be used to other tasks and videos which have the Gaussian noise interference. The fine-tuning work is simple. In the extreme condition, the maximum gap between the average optical flow of each block is 255 (one block is completely filled with optical flow and the other block has no optical flow at all), so we only need to adjust at most 255 times in the worst case. In fact, we only need to fine-tune in the 0–20 range, because the larger the threshold ε, the looser the conditions for identifying noise, and the smaller the threshold ε, the more stringent the conditions for identifying noise.

While noise images are determined, the median filter is used to reduce the interference of white noise on the average optical flow calculation, as the nonlinear median filter is proved to be effective in removing speckle noise.

After filtering optical noises, an automatic editing method for video based on double pointers is proposed in this section, which aims to find and save motion video frames and cut off static and redundant video clips. The basic step is as follows (the pseudo code is shown in Appendix):

(1) Create a start pointer S and an end pointer E and put the two pointers in the head of the video frames.
(2) Calculate the average optical flow of each frame.
(3) Set the upper threshold $\mu_{\max}$ and lower threshold $\mu_{\min}$ of the optical flow average. When the average optical flow of a frame is lower than the lower threshold, it means that the frame has little change compared to the frame at the next time step, so this frame is labeled as p; when the average optical flow is higher than the lower threshold and lower than the upper threshold, it means that the frame has a relatively big change compared to the frame at the next time step, so this frame is labeled as m; when the average optical flow is higher than the upper threshold, it means that the frame has a significant change compared with the frame of the next time step, so this frame is labeled as r.
(4) The E pointer scans each frame in chronological order. When the E pointer encounters the frame marked as m and r, the E pointer moves to the next frame without any operation. When the E pointer encounters the frame marked as p, relabel this frame as E. Repeat the step (5) until the whole frames are scanned.
(5) The S pointer scans each frame in chronological order. When the S pointer encounters the frame marked p and r, the S pointer moves to the next frame without any operation. When the S pointer encounters the frame marked m,

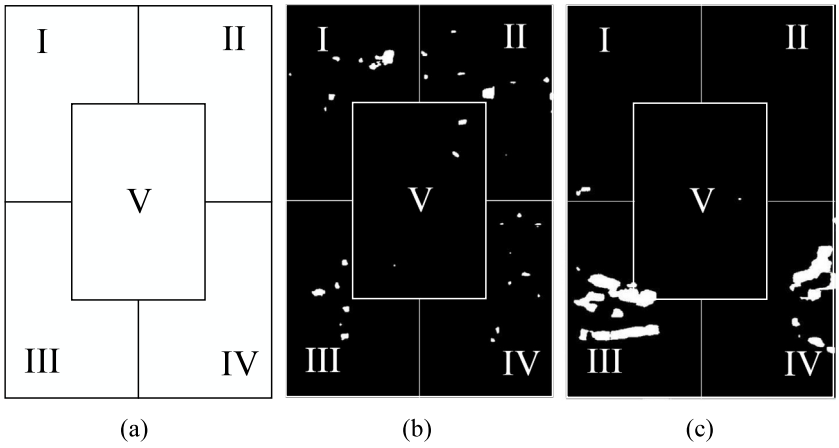

Fig. 3. Demonstration diagram of optical flow image blocks.

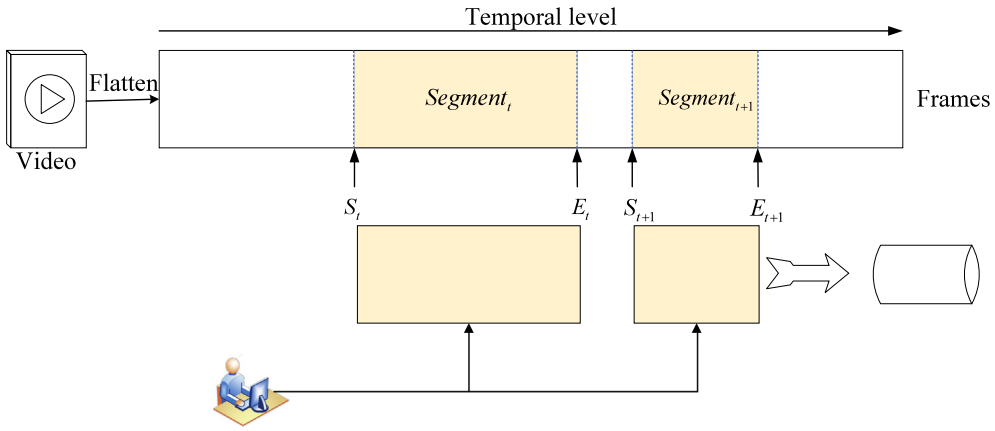

Fig. 4. Demonstration of the automatic editing method.

relabel this frame as *S*. Repeat the step (5) until the whole frames are scanned.

(6) Search frames sequentially. When the frame marked *S* is encountered for the first time, find the frame marked *E* for the first time after the frame. Then, record the corresponding time steps of the two pointers as a segment pair (t_{S1}, t_{E1}) and stored in the segment frames database.

(7) Search for the frame with mark *S* after $t_{E1} + 1$ time step, and look for the nearest *E* pointer after the *S* pointer to form a new segment pair and store it in the segment frames database.

(8) Repeat the step (7) until no new segment pair can be stored in the segment frames database. The segment pairs stored in the database are the effective start and end editing points of valid video frames.

The automatic editing method can be depicted in Fig. 4.

Through the above method, a large number of subdivided small videos can be obtained from the original video and invalid clips with no obvious motion changes can be deleted (usually such invalid clip occupies a large part of the video). Moreover, this algorithm guarantees that different motions in the video will be separated to isolated clips as still frames are existed between two motion frames and can be recognized by the algorithm.

After automatic editing the frames, each clip is labeled manually with motion type to form an engineering dataset that can be used for feature learning.

3.3. The spatial-attention-based agent focusing mode (SAB)

There are many features in one frame. Focusing on different feature would lead to different motion recognition results. The feature recognition process is shown in Fig. 5. The last feature layer (convolution layer) is the activation mapping. Fig. 5(a) is the original tightening operation frame, and Fig. 5(b) and (c) are the heat maps representing the feature parts that the learning model focuses on. It can be seen that the learning model focuses on the hand feature in Fig. 5(b) and focuses on the bolt feature in Fig. 5(c), which could cause different classification results.

Therefore, before entering to feature learning mode, guiding the learning model to pay more attention on important features could get better benefits and achieve higher diagnosis accuracy.

The SAB mode is proposed to map the frame into feature space where important features are emphasized. The flowchart of the SAB as shown in Fig. 6. Firstly, a deep pretrained networks (Googlenet) is used as basic structure of the learning model from [29] due to this model achieves high performance on feature extraction and image classification at present. Secondly, the top 10 layers of the Googlenet, owing the ability to extract general features from any data, are frozen (the parameters of these layers cannot be altered while training) so that the feature extraction ability from the original Googlenet can be utilized and transferred to new tasks. Then, the next 15 layers of the model, owing the ability to extract specific features from data, are not frozen so as to be updated by training with new image data. Thirdly, the last classification layer of the model is replaced by fully-connected layers, so that the Googlenet model can be trained to extract

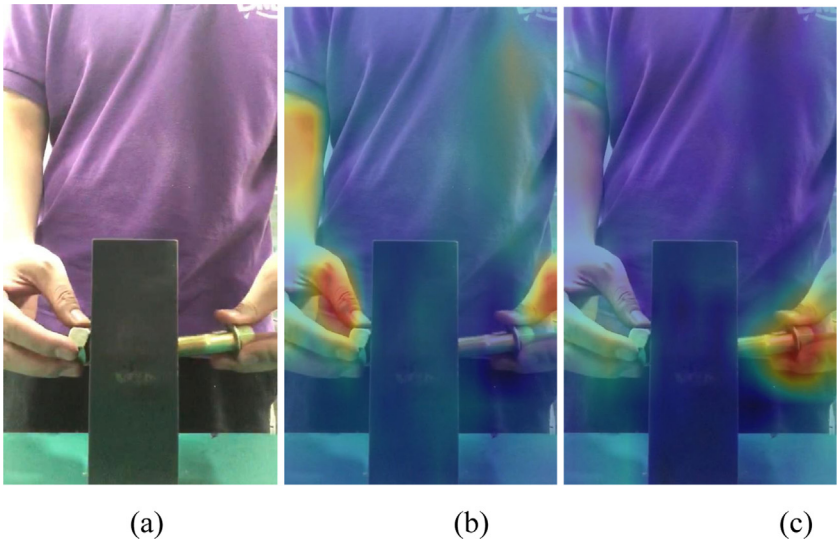

Fig. 5. Example of feature learning process in visualize way.

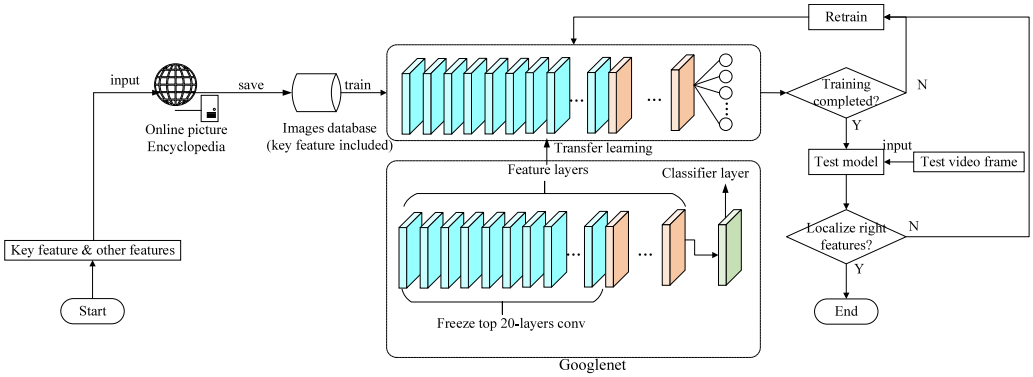

Fig. 6. The flowchart of the SAB mode.

important feature based on new data source. Finally, data with desired key features are collected to train the built model, thus making the model be able to recognize important features.

In detail, search and collect image data with desired features in the picture encyclopedia, and then train the improved model with the collected data. The training objective is to increase the specific feature extraction ability of the SAB model. The classes we choose are bolts, hands, clothes, mechanical parts, desks and machines. The loss function is the cross entropy of the predict output and the real label (one-hot encoding). The input sizes of the fully-connected layers are 1000, 512 and 128, respectively (each output layer is followed by a dropout method to reduce overfitting effects). After training process, the improved model gets the capability to map images to feature space where desired features we need are strengthened.

3.4. Long-short term memory-based feature learning mode (LSB)

Through the above method, the video data are edited into a collection of multiple video clips with important motion features, which reduce the complexity of video diagnosis methods. In this section, an LSB model integrating P3D blocks (as shown in Fig. 7) and recurrent model is proposed as a novel video diagnosis method to achieve high performance on video diagnosis in terms of diagnosis accuracy and training time cost. As shown in Fig. 8, the LSB model combines the feature learning capabilities of the C3D model for short-term clips and the GRU for long-term clips to improve the accuracy of the model. Moreover, an improvement is made in this model to increase its efficiency, that is, the C2D+C1D model is proposed based on the C3D model to reduce the training cost used in video diagnosis, where the C2D model is able to extract spatial feature, and the C1D model is able to extract temporal model.

4. Experiments

4.1. Datasets

The purpose of the engineering application in this paper is to propose an intelligent method to automatically monitor and diagnose the worker's operation process, replacing manual video diagnosis methods that is energy and time consuming for human. Therefore, a large number of video data recorded during the operation of workers are collected as the experimental data to demonstrate and verify the reliability and robustness of the proposed model in this paper. The video data consist of three actions: a worker tightens the bolts correctly, a worker tightens the bolts incorrectly, and a worker unloads the bolts. Due to the above-mentioned actions are frequently occurred in industry, proposing a model that is able to discriminate these actions in video clips can realize automatic diagnosis of worker action and make intelligent manufacturing technology come true in engineering.

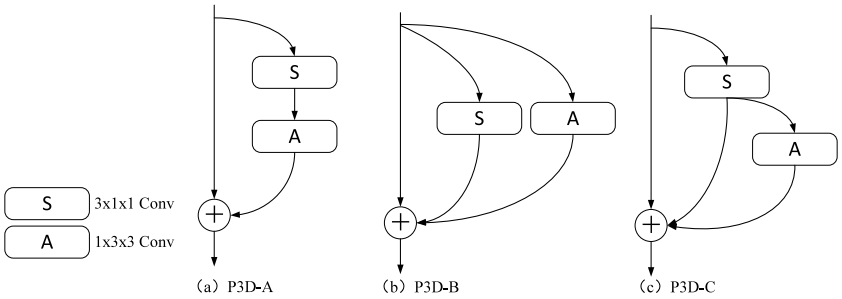

Fig. 7. Three structures of P3D blocks.

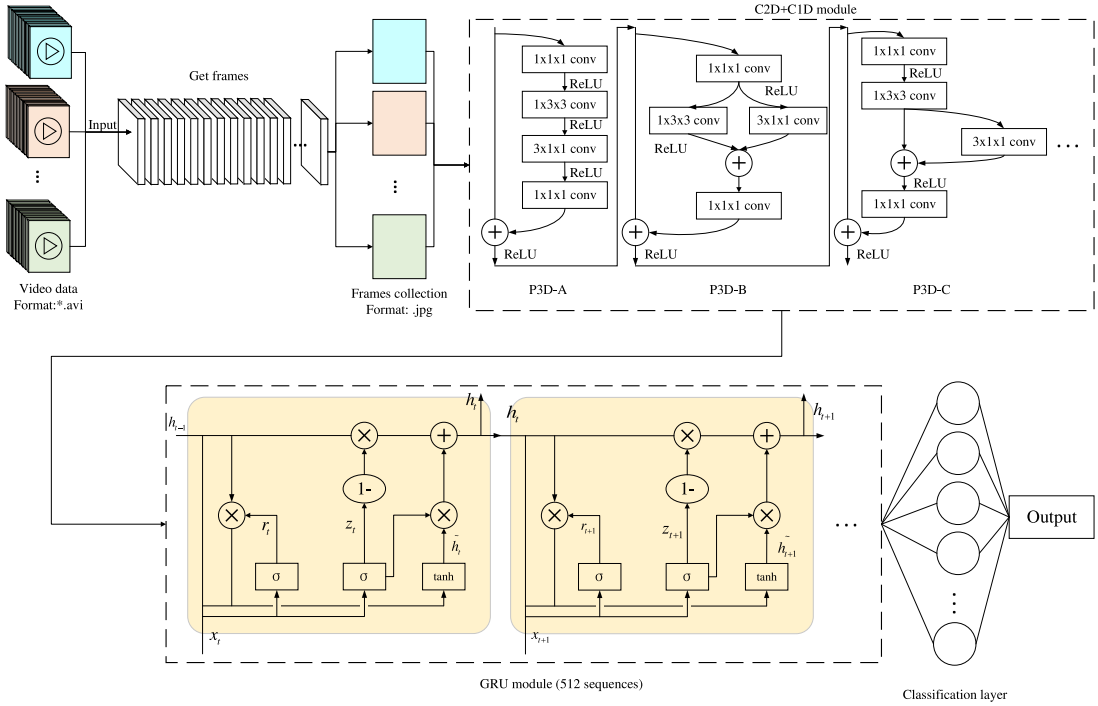

Fig. 8. The process of the LSB model.

4.2. Temporal-attention-based frame splitting mode (TAB)

4.2.1. Noise judgment method based on the partition idea

To verify the effectiveness of the noise judgment method on filtering the optical flow noise, it compares the optical flow image with and without the processing of the noise filtering method. The comparison results as shown in Fig. 9. Fig. 9(a) and (b) are the comparison of the noise optical flow without and with the noise filtering method respectively. It can be seen in Fig. 9(a)–(b) that most of noise are filtered by using the noise filtering method proposed in this paper and the average optical flow of the noise image is reduced from 3.751 to 1.310, which decreases the possibility of misclassification. Fig. 9(c) and (d) are the comparison of the clean optical flow without and with the noise filtering method respectively. Due to no noise are emerged in the original optical flow frame, the filtering method regards it as a clean optical flow frame.

From this experiment, it can be concluded that the noise judgment method based on the partition idea show great performance on discriminating noise optical flow frames.

4.2.2. Sensitivity analysis for automatic editing method

The automatic editing method we proposed has two hyper parameters μ_{min}, μ_{max}. Due to the influence of noise, the choice of μ_{min} is highly sensitive to the performance, so we fix μ_{max} to 10 (insensitive to results) and perform sensitivity analysis on μ_{min}. In this experiment, a one-hour worker operation video is input to this automatic editing method (20 min static frames and 70 assembly actions are ground truth (GT)). μ_{min} is selected in the open interval of 0–10. The sensitivity analysis results can be shown in Figs. 10 and 11.

It can be summarized that $\mu_{min} = 5$ is a better choice, because all the prediction metrics in this value are close to the ground truth. Therefore, the hyper parameters we adopted in this video task are $\mu_{\max} = 10$, $\mu_{\min} = 5$ (hyper parameters can be adjusted according to real situation), then algorithm 1 is performed to automatically split the video data. The algorithm predicts 18.0 min static frames and split 90 actions clips in the video. From the results, it can be known that the algorithm has a strong ability to split video clips, thus some videos containing only one action are still segmented to several clips. By analyzing the segmentation results, we find that some operator assembly actions are slow, so many meaningless movements that are unrelated to the assembly appear in one action, and these movements are segmented by the automatic editing method. As over-segmented video clips are eventually labeled by human, the irrelevant actions can be examined and deleted, so they have no effect on the following training results. Moreover, this over-segmentation ability can avoid missing some small actions related to assembly process.

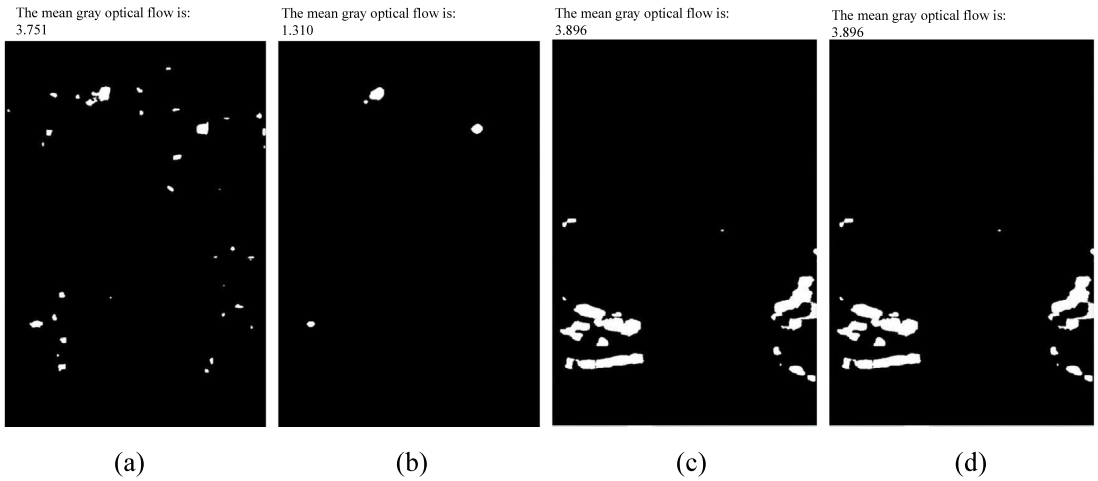

Fig. 9. The experiment results of the noise judgment method based on the partition idea.

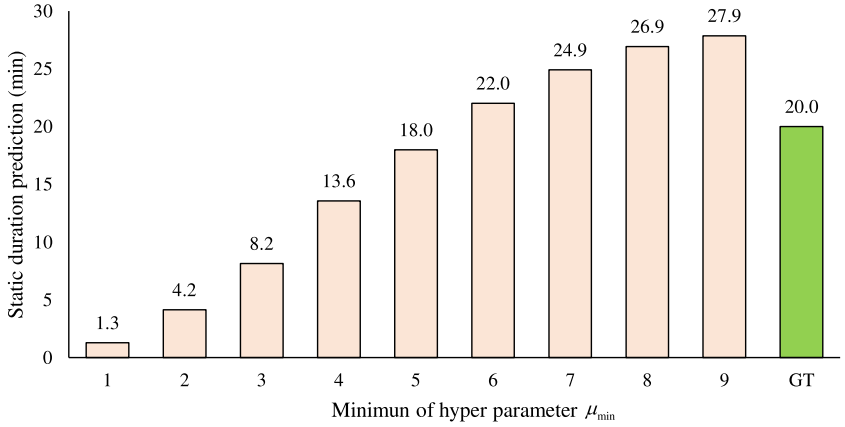

Fig. 10. Sensitivity analysis for static duration prediction.

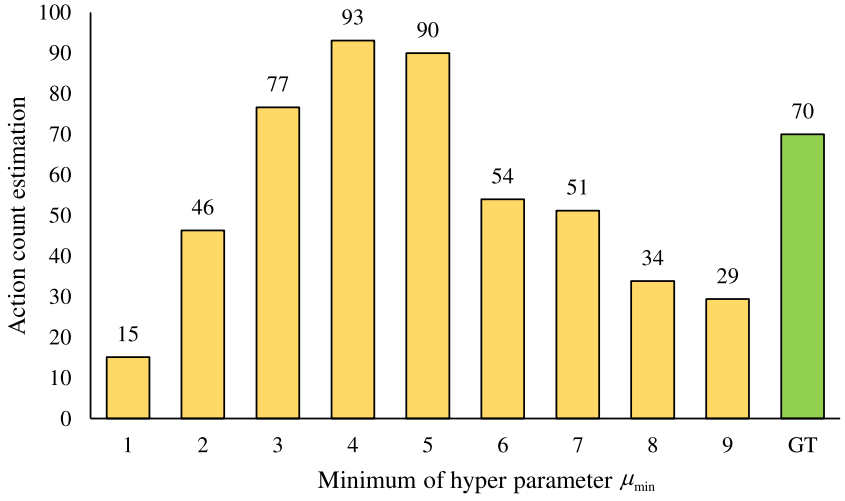

Fig. 11. Sensitivity analysis for action count estimation.

In this experiment, a total of 100 videos of three action categories are prepared by this model, with video duration ranging from 20 s to 30 s, which are divided into the training and test set based on the proportion of 7:3.

4.3. The spatial-attention-based agent focusing mode (SAB)

To identify the types of operator actions, and at the same time distinguish whether the bolts are qualified to tighten after the operator completes the operation, the worker's hands and

Table 1
The architecture of state-of-the-art models.

C3D	P3D/R3D	LRCN	LSB
Conv1 $[3 \times 3 \times 3, 64]$	Conv1 $[3 \times 7 \times 7, 64]$	TimeDistributed Conv1 $[7 \times 7, 32]$	Conv1 $[3 \times 7 \times 7, 64]$
Conv2 $[3 \times 3 \times 3, 128]$	Conv2 $\begin{bmatrix} 1 \times 3 \times 3, 32 \\ 3 \times 1 \times 1, 32 \end{bmatrix} \times 3$	TimeDistributed Conv2 $[3 \times 3, 32]$	Conv2 $\begin{bmatrix} 1 \times 3 \times 3, 32 \\ 3 \times 1 \times 1, 32 \end{bmatrix} \times 3$
Conv3 $\begin{bmatrix} 3 \times 3 \times 3, 256 \\ 3 \times 3 \times 3, 256 \end{bmatrix}$	Conv3 $\begin{bmatrix} 1 \times 3 \times 3, 64 \\ 3 \times 1 \times 1, 64 \end{bmatrix} \times 3$	TimeDistributed Conv3 $[3 \times 3, 64] \times 2$	Conv3 $\begin{bmatrix} 1 \times 3 \times 3, 64 \\ 3 \times 1 \times 1, 64 \end{bmatrix} \times 3$
Conv4 $\begin{bmatrix} 3 \times 3 \times 3, 512 \\ 3 \times 3 \times 3, 512 \end{bmatrix} \times 2$	Conv4 $\begin{bmatrix} 1 \times 3 \times 3, 128 \\ 3 \times 1 \times 1, 128 \end{bmatrix} \times 3$	TimeDistributed Conv4 $[3 \times 3, 128] \times 2$	Conv4 $\begin{bmatrix} 1 \times 3 \times 3, 128 \\ 3 \times 1 \times 1, 128 \end{bmatrix} \times 3$
Dense $[4096] \times 2$	Conv5 $\begin{bmatrix} 1 \times 3 \times 3, 256 \\ 3 \times 1 \times 1, 256 \end{bmatrix} \times 3$	TimeDistributed Conv5 $[3 \times 3, 256] \times 2$	Conv5 $\begin{bmatrix} 1 \times 3 \times 3, 256 \\ 3 \times 1 \times 1, 256 \end{bmatrix} \times 3$
	Dense $[4096] \times 2$	TimeDistributed Conv6 $[3 \times 3, 512] \times 2$	GRU [2048]
		LSTM [2048]	
Classification layer			

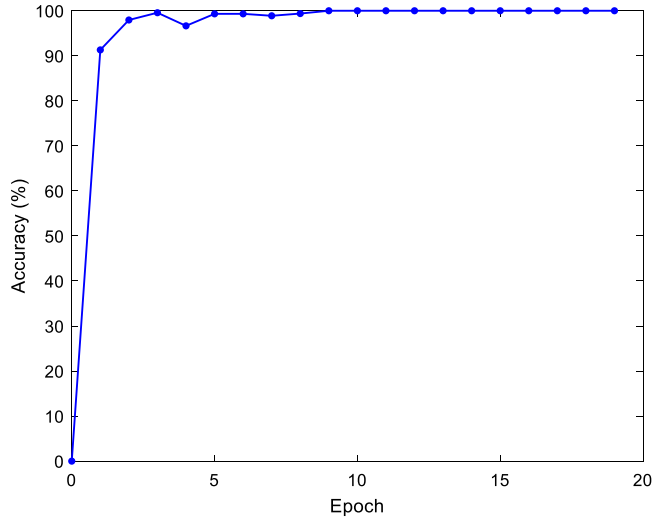

Fig. 12. The accuracy curve of the SAB.

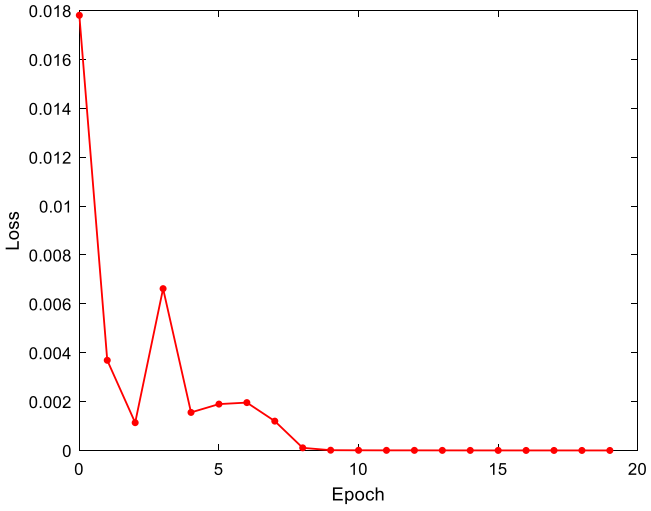

Fig. 13. The loss curve of the SAB.

bolts are two features that need to be focused on during feature extraction. Firstly, tons of images with hand and bolt features are collected to form the image dataset. Then the image dataset is divided into the training and test set based on the proportion of 7:3. Next, the training and test dataset are input to train the SAB, where the number of training epochs is set to 20, the batch size is set to 12, the optimizer we adopted is ADAM algorithm [30] and the learning rate is set to 0.0002. The training accuracy curve and loss curve are shown in Figs. 12 and 13.

It can be seen from Figs. 12 and 13 that the SAB model has good learning performance on the collected dataset. In order to verify that the model has the ability to learn the characteristics of hands and bolts, multiple operation diagrams are input into the trained model, and the visualized heat map of feature learning are produced to reflect the training effect. The result is shown in Fig. 14.

It can be seen from the heat map that the well-trained SAB model has the ability to capture the features of the operator's hand and bolt. Afterwards, the data processed by the SAB form the attention-based dataset for the LSB model training.

4.4. Long-short term memory-based feature learning mode (LSB)

The LSB model is used to complete the video diagnosis task. The processor of the computer used in this experiment is Intel(R) Core(TM) i7-8565U CPU. To validate the state-of-art of the proposed model, the C3D model, P3D model, R3D model and LRCN model, which do not contain attention mechanism, are compared with the LSB model by training them with the same videos collected in this paper. Moreover, attention-based models such as GSM, LSTA and SAB-LSB (ours) model are also added to the comparison task to validate the effectiveness of the SAB model in video diagnosis task. As varies in architecture of state-of-the-art models, it is not appropriate to directly use these models from other papers without any adjustment. In order to ensure the fairness of the comparative test, the number of training parameters of each model is guaranteed to be in the same order of magnitude by adding some dense layers at the end of some models. The above compared model architectures can be seen in Table 1.

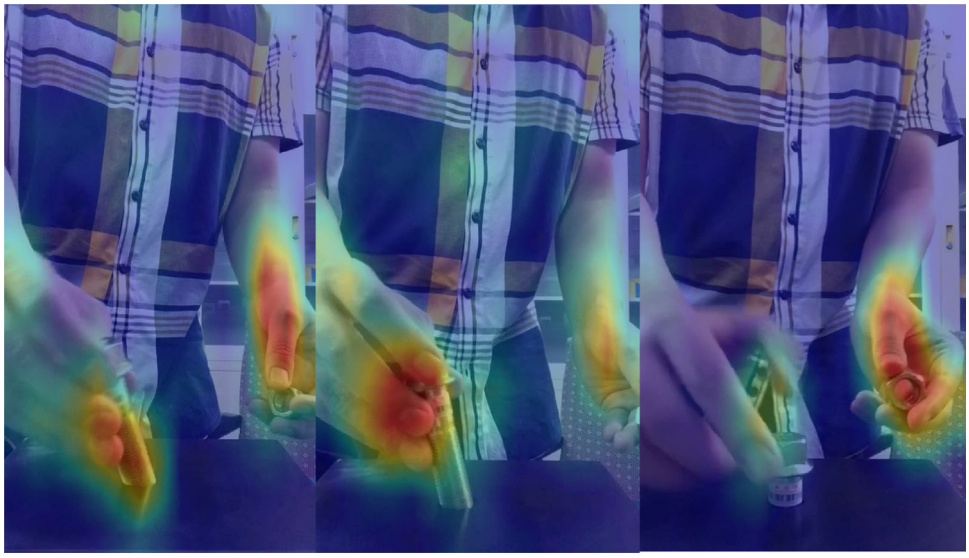

Fig. 14. The visualization heat map of the training process in the SAB model.

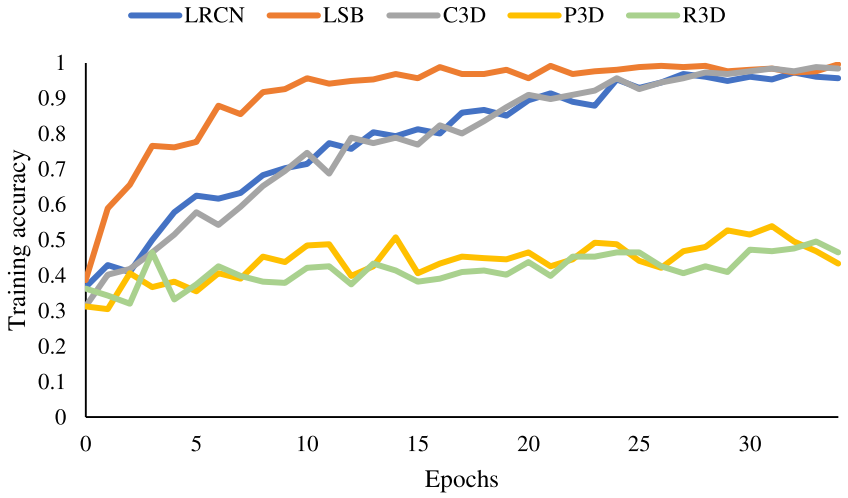

Fig. 15. The training accuracy of the models.

The training and validating results as shown in Figs. 15–18. It should be noted that the curves of each model are smoothed to reflect the trend of each curve more clearly.

To verify the efficiency of those models, the parameters number (PN), the total training time (TT), the final validation accuracy rate (FR) and the average inference time (IT) are compared in the Table 2. By testing the inference time, the hyper parameters are set to the same value. In detail, the batch size of the input frames is 16, the input frame size is (80,80,3) and each action duration is 20 s. The inference time of each video is computed by setting time functions before and after the main test function respectively and calculating the difference between the two time-value. 35 action videos are input to calculate the average inference time. It should be noted that only CPU is adopted to do the inference task.

It can be seen that the LSB model outperforms all baseline models without attention mechanism (C3D, P3D, R3D, LRCN) in terms of the training accuracy and training loss, while the performance of the LSB and C3D model are similar and both outperform other models with regard to the validation accuracy and validation loss. As the total training time and inference time of C3D is bigger than the LSB, it can be concluded that the LSB model has a comprehensive performance on video diagnosis tasks.

Table 2
Performance comparisons in terms of parameters number, total training time and final validation accuracy rate.

Models	PN	TT	VR	IT
C3D	82,202,371	65.14 h	60.48%	79.48 s
P3D	53,150,403	17.07 h	51.41%	45.42 s
R3D	53,150,403	17.85 h	49.06%	48.88 s
LRCN	25,709,859	7.84 h	43.91%	20.89 s
GSM	–	7.42 h	84.18%	17.29 s
LSTA	–	8.50 h	67.19%	20.23 s
LSB (ours)	63,755,523	9.27 h	64.38%	18.43 s
SAB-LSB (ours)	63,755,523	5.22 h	90.07%	15.45 s

To further do comparative experiments in attention-based models, the GSM, LSTA and SAB-LAB model are also compared, and the training and validation results are shown in Figs. 19–22. The LSB without SAB model is also added to do the ablation experiment.

It can be seen that the SAB-LSB model outperforms the LSB model at every aspect, which could prove that the preprocessed dataset by the SAB model can boost the training efficiency and increase effect of the LSB model in video feature learning. Moreover, GSM and LSTA also show good performance on training time

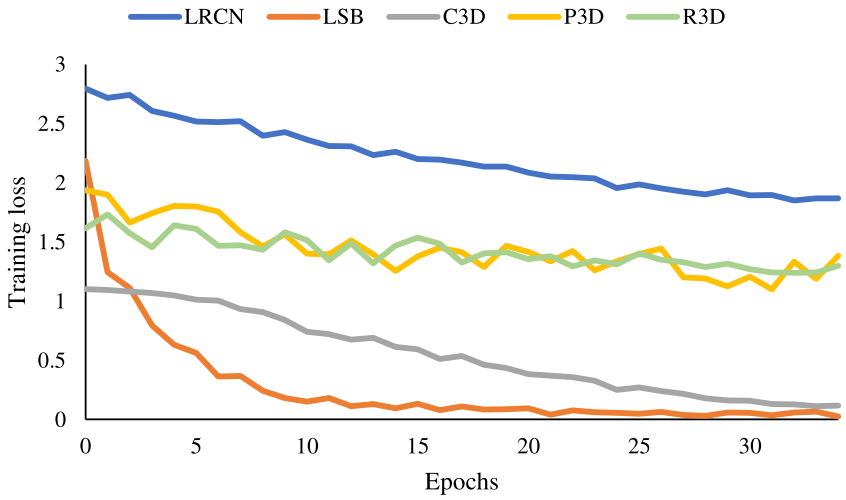

Fig. 16. The training loss of the models.

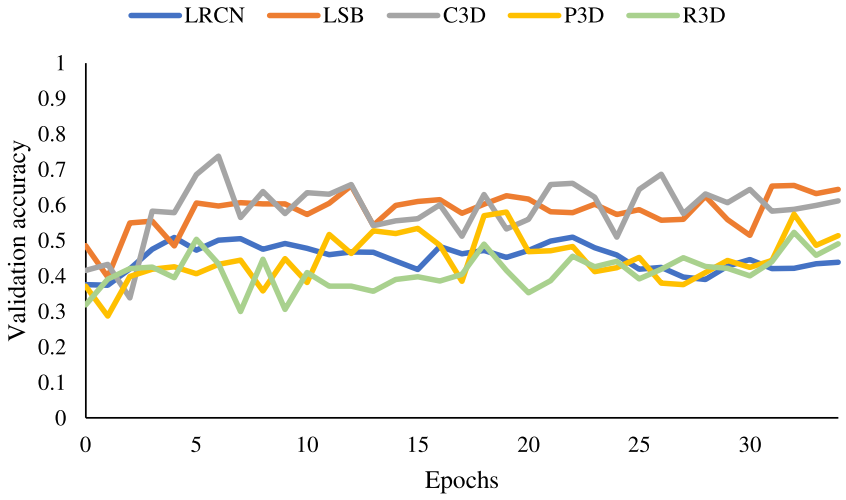

Fig. 17. The validation accuracy of the models.

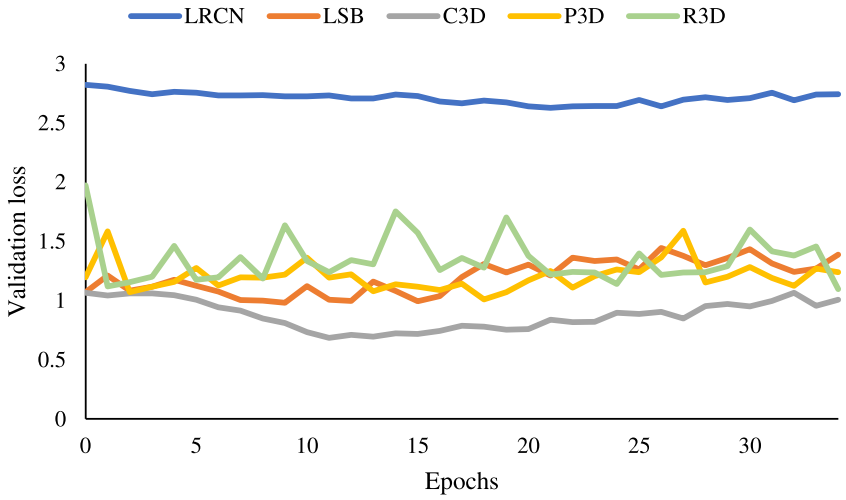

Fig. 18. The validation loss of the models.

and inference time than baseline models. We can conclude that attention tricks in action recognition models can accelerate the training time of video diagnosis tasks and shorten the inference time of video test. It should be mentioned that the evaluation scenario is not very complex, but it is a realistic scenario for the application. By doing this evaluation test, the robust and good performance of our model are validated. At present, this method is put into practice and show a reliable result.

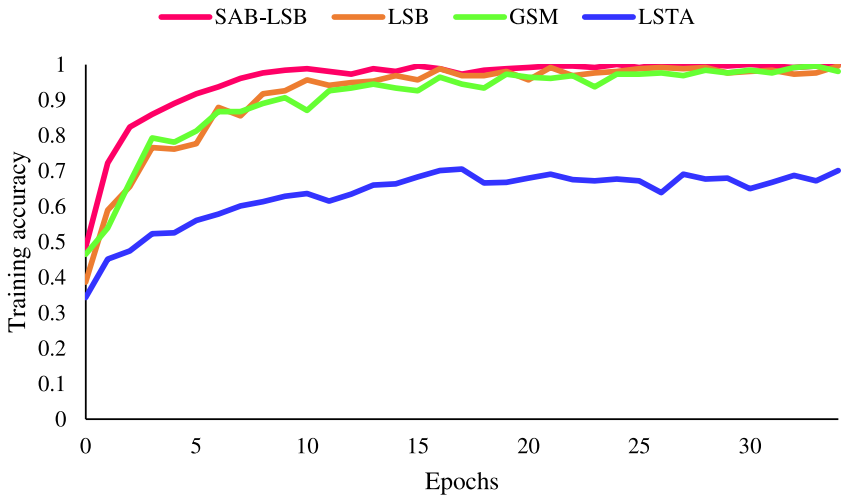

Fig. 19. The training accuracy of attention-based models.

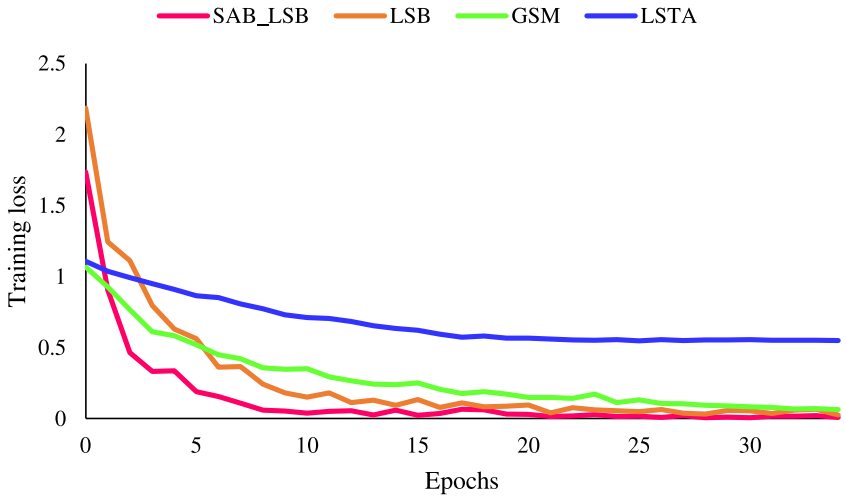

Fig. 20. The training loss of attention-based models.

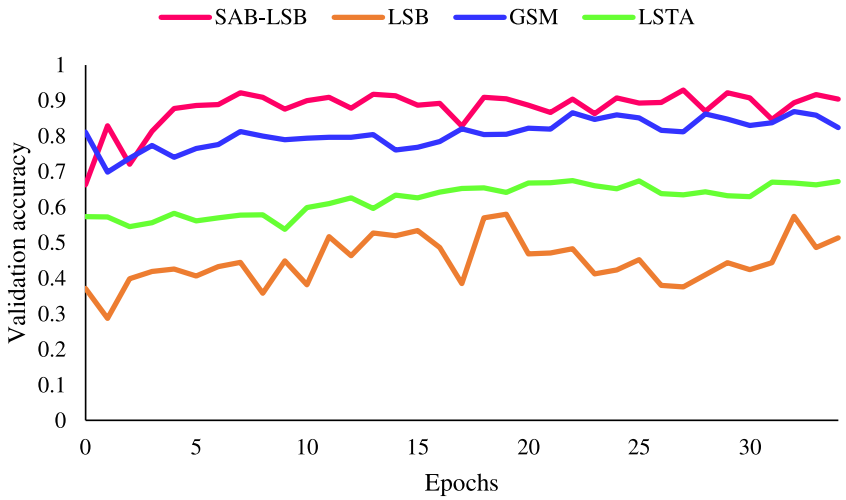

Fig. 21. The validation accuracy of attention-based models.

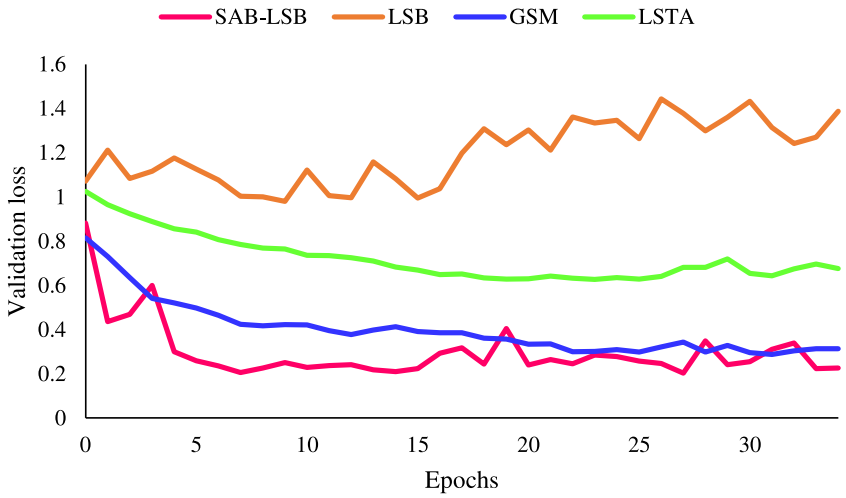

Fig. 22. The validation loss of attention-based models.

5. Discussion

1. Training time. It can be seen that 3D convolutional kernels lead to longer training time, although validation accuracy performance is relatively high (at 60.48%). The C3D model mainly consists of 3D convolutional kernels, so it spends the longest time finishing the training task among whole above models (65.14 h). The P3D and R3D models replace C3D model with C1D kernel plus C2D kernels in different ways, reducing the training time and complexity of the model. The LRCN model consists of the 2D CNN kernels and the long-term feature learning model LSTM, which shows higher training efficiency (7.84 h) than those models without long-term feature learning module, such as the C3D model (65.14 h), P3Dmodel (17.07 h) and R3D model. The LSB model takes advantage of the good points of the P3D model and LRCN model by combining P3D model with a concise long-term feature learning model GRU. Therefore, even the LSB model has larger size of parameters than the P3D model, the training time of the LSB is shorter than the P3D model. The SAB-LSB model has the best performance on training efficiency (5.22 h) and inference time (15.45 s per video) among all models, as the SAB model extracts the key features of training dataset to the LSB model before, saving the time for the LSB model to learn data features. Attention-based model such as GSM and LSTA also show good performance on training time and inference time in experiments.
2. Over-fitting problem. Over-fitting problem is the main cause to a low validate accuracy of video diagnosis models. The LRCN model has a serious over-fitting problem as there is a significant gap between the training and validation accuracy rate. It may contribute to the fact that some hyper-parameters of the LRCN model are adjusted to make sure the model can be compared in a relatively same level (same multitude of parameters number), which increases the training parameters number. The over-fitting problem in the C3D model, P3D model, R3D model and LSB model are reduced relatively, but the validation accuracy of each model is still around 65%. The SAB-LSB model has the best validation accuracy, which proves it reduces the over-fitting problem very well. It also shows that data preprocessing by the SAB can decrease the effect of over-fitting problem on model training.
3. Induction discussion. Since the LSB model has a better performance than other state-of-the-art models such as the C3D model, P3D model, R3D model and LRCN model, it can be easily concluded that the SAB-LSB model has a better performance than the SAB-C3D, SAB-P3D model, SAB-R3D model and SAB-LRCN model. Also noteworthy is that every model has its application condition, choosing the appropriate state-of-the-art model for specific application situation would make the best result. For example, if video clips are short enough, long-term feature learning model is not necessary applied to video diagnosis task because short-term feature learning model can satisfy the need for feature learning.
4. There is no standard way to build attention tricks, all tricks should be built according to the project demand. For example, if the complexity of C3D is the main bottleneck of the whole model, designing a light-weight gated-shift module to replace C3D is a good option (i.e. GSM); if the base model is weak in spatial feature extraction, then a spatial attention trick could be added to the model (i.e. LSTA); if the important features have been clarified based on engineering experience, information-based attention model can be designed to avoid models learning knowledge from scratch (i.e. TARM).
5. It should be noted that hyper-parameters and structure of the state-of-the-art models are altered to make sure the comparative experiments be fair. Therefore, the results of the experiment are relative rather than absolute. If each model is adjusted to the best state, the results of each model are not comparable due to the different size of model parameters. In addition, the LSB model is inspired by the LRCN model and the P3D model, so it has both advantages of these two models and overcome the limits of each model in video diagnosis process. Moreover, the SAB model provides the possibility to greatly simplify the complexity and improve the efficiency and accuracy of video diagnosis, as it extracts important features from the video data in advance to avoid the model training large amount of redundant video data.
6. As a high computing cost is demanded in the video data training, it is impossible to train benchmark datasets such as UCF101 and HMDB51 by using a CPU. Therefore, TARM has its restriction that the size of video dataset should not be too big.

7. It can be seen that attention-based tricks play an essential role in accelerating the training time and inference time of video recognition models, so attention tricks can be further explored and designed to reduce the complexity of these models in the future works. It is a potential research area that can be beneficial to both academics and industry.

6. Conclusion

There are three contributions of the model presented in this paper, which are detailed as follow:

1. The TARM model is proposed to improve video diagnosis performance on video dataset with long-term and fine-grained features to learn. Attention-based mechanisms and long-short term feature learning model are main ideas of TARM. In detail, temporal attention-based mechanism focuses on temporal dimension of the video data to filter out motionless video contents in raw industrial video data. Spatial attention-based mechanism concentrates on spatial dimension of each frame of video to reinforce its key features and weaken its other features. Long-short-term feature learning model focuses on extracting both long-term and short-term feature of video data sufficiently to better learn fine-grained, minor changed and long correlation video data features.
2. To verify the high accuracy and efficiency in training and inference phases of TRAM, baseline models and attention-based models are cited, compared and discussed in this paper. To validate robust of TRAM on the application of industrial video diagnosis, manual assembly action videos, consisting of the normal assembly operation, wrong assembly operation and prepared action, are recorded to be used as the experimental dataset in this paper. The number of each model's training parameters and structure is tuned to ensure a fair comparison. The total training time, training accuracy and loss, validation accuracy and loss of each model are compared.
3. In results, TARM outperforms the state-of-the-art models and decreases the over-fitting problem while training video data. Moreover, TARM shows potential in intelligent video diagnosis in industry, because the training efficiency and accuracy are highest in the comparative experiments. It can be look forward that TARM will probably play an essential role in the industrial video fault diagnosis and monitoring.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We gratefully acknowledge the financial support from the National Basic Scientific Research Program of China (JCKY2018208A001), and Tsinghua University-Weichai Power Joint Institute of Intelligent Manufacturing (JIIM02).

Appendix

Algorithm A

Sampling frames from video to build frames set at f frequency
Create start pointer S and end pointer E, put S and E in the head of frames set
Create stack D to save segment pairs
While True:
    Calculate $\overline{F}$ for corresponding frame of pointers S and E
    if $\overline{F} < \mu_{\min}$:
        Label frame as p
    else if $\mu_{\min} \le \overline{F} < \mu_{\max}$:
        Label frame as m
    else:
        Label frame as r
    if S is not None:
        S=S.next
    if E is not None:
        E=E.next
    if S is None and E is None:
        break
Initialize S and E in the head of frames set
While True:
    if label(S)==m:
        Re-label frame as S
    if label(E)==p:
        Re-label frame as E
    if S is not None:
        S=S.next
    if E is not None:
        E=E.next
    if S is None and E is None:
        break
Initialize S in the head of frames set
T=None
While True:
    if label(S)==S and T is None:
        T=position(S)
    if label(S)==E and T is not None:
        D.append((T, position(E)))
        T=None
    if S is None:
        break
    S=S.next
Output each pair from D

References

[1] Baker S, Roth S, Scharstein D, Black MJ, Lewis JP, Szeliski R. A database and evaluation methodology for optical flow. p. 1-8.
[2] Barron JL, Fleet DJ, Beauchemin SS, Burkitt TA. Performance of optical flow techniques. p. 236-42.
[3] Wang H, Kläser A, Schmid C, Liu C-L. Dense trajectories and motion boundary descriptors for action recognition. Int J Comput Vis 2013;103(1):60–79, 2013/05/01.
[4] Wang H, Schmid C. Action recognition with improved trajectories. p. 3551-8.
[5] Feichtenhofer C, Pinz A, Zisserman A. Convolutional two-stream network fusion for video action recognition. p. 1933-41.
[6] Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Van Gool L. Temporal segment networks: Towards good practices for deep action recognition. p. 20-36.
[7] Tran D, Bourdev L, Fergus R, Torresani L, Paluri M. Learning spatiotemporal features with 3D convolutional networks. p. 4489-97.
[8] Qiu Z, Yao T, Mei T. Learning spatio-temporal representation with pseudo-3D residual networks. p. 5534-42.

[9] Tran D, Wang H, Torresani L, Ray J, LeCun Y, Paluri M. A closer look at spatiotemporal convolutions for action recognition. 2017, https://ui.adsabs.harvard.edu/abs/2017arXiv171111248T, [November 01, 2017, 2017].
[10] Lin J, Gan C, Han S. TSM: Temporal shift module for efficient video understanding. 2018, arXiv; https://ui.adsabs.harvard.edu/abs/2018arXiv181108383L, [November 01, 2018, 2018].
[11] Girdhar R, Ramanan D. Attentional pooling for action recognition. 2017, arXiv; https://ui.adsabs.harvard.edu/abs/2017arXiv171101467G, [November 01, 2017, 2017].
[12] Sun D, Roth S, Lewis JP, Black MJ. Learning optical flow. p. 83-97.
[13] Taylor GW, Fergus R, LeCun Y, Bregler C. Convolutional learning of spatio-temporal features. p. 140-53.
[14] Konda K, Memisevic R. Unsupervised learning of depth and motion. Comput Sci 2013.
[15] Butler DJ, Wulff J, Stanley GB, Black MJ. A naturalistic open source movie for optical flow evaluation. p. 611-25.
[16] Pan Y, Mei T, Yao T, Li H, Rui Y. Jointly modeling embedding and translation to bridge video and language. p. 4594-602.
[17] Jain V, Bollmann B, Richardson M, Berger DR, Helmstaedter MN, Briggman KL, Denk W, Bowden JB, Mendenhall JM, Abraham WC, Harris KM, Kasthuri N, Hayworth KJ, Schalek R, Tapia JC, Lichtman JW, Seung HS. Boundary learning by optimization with topological constraints. p. 2488-95.
[18] Tran D, Wang H, Torresani L, Feiszli MJae-p. Video classification with channel-separated convolutional networks. 2019, https://ui.adsabs.harvard.edu/abs/2019arXiv190402811T, [April 01, 2019, 2019].
[19] Sudhakaran S, Escalera S, Lanz OJae-p. Gate-shift networks for video action recognition. 2019, https://ui.adsabs.harvard.edu/abs/2019arXiv191200381S, [December 01, 2019, 2019].
[20] Donahue J, Hendricks LA, Rohrbach M, Venugopalan S, Guadarrama S, Saenko K, Darrell T. Long-term recurrent convolutional networks for visual recognition and description. IEEE Trans Pattern Anal Mach Intell 2017;39(4):677–91.
[21] Ballas N, Yao L, Pal C, Courville A. Delving deeper into convolutional networks for learning video representations. Comput Sci 2015.
[22] Shi X, Chen Z, Wang H, Yeung D-Y, Wong W-k, et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. 2015, arXiv; https://ui.adsabs.harvard.edu/abs/2015arXiv150604214S, [June 01, 2015, 2015].
[23] Li Z, Gavrilyuk K, Gavves E, Jain M, Snoek CGM. VideoLSTM convolves, attends and flows for action recognition. Comput Vis Image Underst 2018;166:41–50, 2018/01/01/.
[24] Sudhakaran S, Escalera S, Lanz O. LSTA: Long short-term attention for egocentric action recognition. 2019, 2019 IEEE/CVF Conf. Computer Vis Pattern Recognit (CVPR); https://ui.adsabs.harvard.edu/abs/2018arXiv181110698S, [November 01, 2018, 2019].
[25] Liu J, Wang G, Hu P, Duan L, Kot AC. Global context-aware attention LSTM networks for 3D action recognition. p. 3671-80.
[26] Cherian A, Wang J, Hori C, Marks TK. Spatio-temporal ranked-attention networks for video captioning. 2020, arXiv; https://ui.adsabs.harvard.edu/abs/2020arXiv200106127C, [January 01, 2020, 2020].
[27] Andreadis I. A real-time color space converter for the measurement of appearance. Pattern Recognit 2001;34(6):1181–7, 2001/06/01/.
[28] Mizukami Y, Tadamura K. Optical flow computation on compute unified device architecture. p. 179-84.
[29] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. 2014, https://ui.adsabs.harvard.edu/abs/2014arXiv1409.4842S, [September 01, 2014, 2014].
[30] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, Computer Sci; https://ui.adsabs.harvard.edu/abs/2014arXiv1412.6980K, [December 01, 2014, 2014].

Wentao Luo is a Ph.D. student from the Department of Mechanical Engineering at Tsinghua University, China. He was born in Hunan province, China in 1994. He received the B.S. degree in industrial engineering from the Jilin University, Changchun, in 2017. He is mainly engaged in the research of armored wheel assembly. His research interests include intelligent assembly and industrial intelligence. He also published related papers on rational decision-making and information fusion.

Jianfu Zhang is an associate professor and doctoral supervisor in the Department of Mechanical Engineering, Tsinghua University, China. He graduated from Tsinghua University in 2009 with a doctorate in mechanical engineering. He is mainly engaged in research work in the fields of precision processing technology and equipment, and intelligent manufacturing technology. He has undertaken more than 20 projects such as the National Natural Science Foundation, the National Science and Technology Major Special Project, the Beijing Natural Science Foundation. He has won 8 provincial and ministerial awards, published more than 150 papers, invented more than 30 patents and achieved 20 software copyrights.

Pingfa Feng is a professor and doctoral supervisor in the Department of Mechanical Engineering, Tsinghua University, China. In 2003, he received a doctorate in mechanical engineering from the Department of Transportation and Mechanical Systems at the Technical University of Berlin, Germany. His main research interests include intelligent manufacturing and manufacturing equipment performance analysis and optimization. He has completed or is in charge of more than 20 national 973 programs, 863 programs, major projects, natural science funds, international cooperation and corporate cooperation projects, published 1 monograph, published more than 200 papers, and invented more than 30 patents.

Dingwen Yu is a professor and doctoral supervisor in the Department of Mechanical Engineering, Tsinghua University, China. He has been a senior member of the China Mechanical Engineering Society. His main research fields are information integration of manufacturing systems, etc. He participated in and undertook more than 15 projects such as national "973" basic research projects, 863 national projects, major national science and technology projects, national defense pre-research projects, and projects of the National Natural Science Foundation. He has published more than 130 papers, invented more than 30 patents.

Zhijun Wu is a professor in the Department of Mechanical Engineering, Tsinghua University, China. His main research fields are precision processing technology and equipment, and manufacturing information and system integration technology. He participated in and undertook 18 projects such as the National Natural Science Foundation, the National Science and Technology Major Special Project, the Beijing Natural Science Foundation, the Beijing Science and Technology Program. He has published more than 120 papers, invented more than 30 patents.