Semantic Analysis of Action with Spatio-Temporal Features Based on Object Detection

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

Abstract—**Vast studies on action analysis with spatio-temporal features and deep learning algorithms have proposed countless improvements over the years. This paper presents a domain model with continuous and spatio-temporal features based on object detection for the industrial pipeline scenario. In this way, we can conduct a comprehensive evaluation: action pose accuracy in the spatial dimension and action efficiency in the temporal dimension. The method is applicable to scenarios that require semantic recognition of action in a short time, without any dedicated action capture devices. Firstly, the discretized images are combined into spatio-temporal, structured and continuous action sequences. Then we apply the model to the sequences to get the spatial action information through video streams with only two-dimensional information, and then complete the analysis of the action specification based on these action streams with temporal features. Furthermore, this paper performs several ablation experiments on training strategies and hyperparameters to improve accuracy. Experimental performances show that it achieves an average recognition accuracy of about 96.45%.**

Index Terms—**semantic analysis of action, action specification, spatio-temporal features, deep learning, object detection.**

I. Introduction

THE research of human behavior recognition and analysis plays a significant role at present. Aaron F. Bobick and James W. Davis studied simple human behavior based on Temporal Templates and Dynamic Time Warping [1]. The method requires the temporal and spatial action features simultaneously and extracts the whole video information for recognition. However, when the movement interval of the same movement is inconsistent, its recognition accuracy varies greatly. Then, the variable sequences produced by human behavior began to become the hinge of research. Hidden Markov Model (HMM) and Bayes Model are most representative. B. Matthew, O. Nuria and P. Alex used the coupled hidden Markov model to simulate the interaction process [2]. It proves its superiority over traditional HMM in the task of bimanual action classification. N. Oliver, E. Horvitz and A. Garg applied HMM to implement a hierarchical probability model for perception and learning temporal inference at multiple levels [3]. C. Sminchisescu, A. Kanaujia and D. Metaxas studied human action recognition algorithms based on Conditional Random Field (CRF) and Maximum Entropy Markov (EM) in video sequences [4]. But it no longer meets the demand for long-scale action scenarios.

The continuous study of human brain nerves inspired deep learning, and it has aroused an upsurge [5]. The research of human behavior analysis based on deep learning method has received increasingly attention. S. Lin *et al.* provided an algorithm, two-layered SFA learning structure with 3D convolution and maximum pool layer, to capture abstract and structural features from video and build action recognition models [6]. In 2018, A. Mathis *et al.* proposed an efficient method for markerless pose estimation based on deep neural network, which achieved excellent results with the least training data [7]. There are many difficulties to capture human behavior with large-scale and low-cost equipment, which also lead to huge problems in the study of human behavior analysis, including industrial scenarios. So far, there have been few public available datasets from industrial field.

In this paper, the front-end video capture device is easy to deploy, and is not a dedicated action capture device, which provides the possibility to deploy standard action work at large scale and low cost. We collect the dataset from the real workshops. All images are manually labelled, which makes it a good benchmark to evaluate object algorithms. The difficulty of the research is to extract the region of interest and build an appropriate and accurate recognition model. Therefore, the focus shifts to the recognition and real-time tracking of the hand gesture of the actors in the video.

Here is a list of our main contributions series:

1) We present a domain model with continuous and spatio-temporal features based on object detection for the industrial pipeline scenario. It can also be applied to other analogous scenarios that require semantic recognition of the action in a short time, e.g. sign language recognition scenarios. The results and analysis are given for a guideline.
2) Without the special camera device, we extract the two-dimensional information from the collected video stream to get the spatial action information, and then complete semantic analysis of action based on the action information with time flow. In this case, it provides a possibility to deploy standard action work at large scale and low cost.

II. Related Work

A. Datasets

Numerous datasets and benchmarks for object detection have been released in the past 10 years. For instance, the datasets of PASCAL VOC Challenges(e.g., VOC2007, VOC2012)([8], [9]), ImageNet Large Scale Visual Recognition Challenge (e.g., ILSVRC2014 [10]), MS-COCO Detection Challenge [11], etc. The datasets of PASCAL VOC, and the two versions of Pascal-VOC are mostly used in object detection: VOC07 and VOC12, where the former consists of 5k images and 12k annotated objects, and the latter consists

This work is supported by the National Natural Science Foundation of China under Grant (No. 61703278).

C. Chen, Y. Wang, K. Yi, T. Wang and H. Xiang are with College of Computer Science, Yangtze University, Jingzhou, Hubei, 434023 China.

T. Wang and H. Xiang are corresponding authors: phone: 0086-139-9760-1255; e-mail: txwang@yangtzeu.edu.cn.

of 11k images and 27k annotated objects. These large action datasets helped the development of recognition algorithms on action analysis. Thus, it is necessary to collect and produce datasets in our scenario.

We first record and collect video data using Hikvision DS-2CD2051-1 network camera. The uniform frame rate is 25fps. The video data size is 1280×720. Without affecting the normal operation of the camera, we require adjusting and fixing the camera's angle, height and distance, and maximizing the distance between the camera and the actor to ensure an effective maximum ratio of information in the video screen. For each action type, there are no erroneous actions, only single object information and no other interference items as much as possible. Then each video is divided into several small clips based on basic actions type. In this way, each clip is a single basic unit. If there exist complex basic actions or subset of the basic action class, we will consider dividing them into subclasses or cutting. Then we extract one frame per second from the episodes and save the images extracted by appropriate skipping frames.

B. Object Detection Algorithms

DPM [12], the peak of the traditional object detection methods, was originally proposed by P. Felzenszwalb, D. McAllester and D. Ramanan as an extension of the HOG detector [13] in 2008, and then a variety of improvements have been made by Ross B. Girshick, Pedro F. Felzenszwalb and D. McAllester [14]. But the traditional method encountered bottlenecks soon. Later, the rapid development of deep learning technology injected new blood into the researches of object detection, leading to remarkable break throughs and pushing it forward to a search hot-spot with unprecedented attention. After 2014, there are usually two popular algorithms based on object detection. One is named one-stage detector algorithms that directly detects and recognizes objects in images in a single run, typically including SSD [15], YOLO(v1/2/3)([16]–[18]), RetinaNet [19], CornerNet [20] and CenterNet [21], *et al.* They have a faster detection speed. The other is called a two-stage detector algorithm, which first uses the network to extract the objects features in the image that are bounded by the bounding box, and then performs learning based on these features, and finally recognizes each bounding box and obtains the category of each object. Typical algorithms include R-CNN [22], SPPNet [23], Faster R-CNN [24], R-FCN [25], Mask R-CNN [26], TridentNet [27], NAS-FPN [28] and Cascade R-CNN [29], *et al.* Compared with the former, these methods have a higher recognition accuracy.

However, not all of the above algorithms are suitable for action detection and recognition in industrial pipeline operation scenarios. The differences in time and complexity of each action can cause large intra-class distances, whereas several actions with similar processes can result in small inter-class distances. Thus, how to construct a model by selecting some baselines to accurately recognize the workers actions in real time is a challenging problem.

III. Methods

To complete the entire detection and recognition task for this scenario, this section mainly contributes to a method for processing raw dataset, baselines based on object detection algorithms, and whole scheme for action recognition.

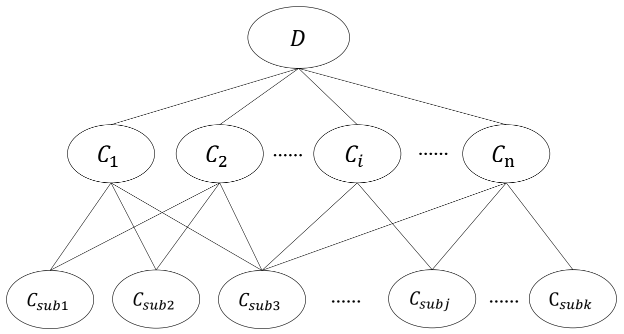


Fig. 1. The Key idea of Action Decomposition Strategy.

A. Training and Test Samples

For action detection tasks, we need to introduce labelled training and test samples for evaluation. Before training, these images need to be preprocessed and labelled. We chose the standardized annotation rules of PASCAL VOC to ensure consistency of reuse rules. The object detection and annotation tool, Visual Object Tagging Tool (VoTT), released by Microsoft is adopted to label the dataset. Due to the high similarity of adjacent video frames and the heavy workload of frame-by-frame labelling, our solution is to select keyframes for labelling. In this way, it can reduce the number of markers and grab more effective ranges. All images are manually labelled, which make it a good benchmark for evaluating object algorithms.

Unstructured raw data are difficult to train in experiments. It is a huge challenge in this paper. Fortunately, we find that there are always some basic actions in every action flow, while the order of basic action combination is different. Besides, the action in industrial scenarios has repeatability, continuity and spatio-temporal features. Thus, these raw data are transformed into structured recognizable sequences that can be combined and analyzed over time by action decomposition. Fig. 1 shows the action decomposition strategy. It refers to an action dataset D consisting of many basic units. Let $C = \{C_1, C_2, \cdots\cdots C_n\}$ be basic unit with n instances. $C_{sub} = \{C_{sub1}, C_{sub2}, \cdots\cdots, C_{subj}, \cdots\cdots C_{subk}\}\,(0 < j \leq k, j \in N^*, k \in N^*)$ is the j-th subunit of i-th basic unit. More specifically, it can be abstractly described as a tree-like structure $Tree =< V, E >$, which is composed of n-order τ edges. The V is a set of action vectors consisting of D. When $n > 1$, the remaining nodes can be divided into m disjoint finite action sets $Tree_1, Tree_2, Tree_3, \cdots\cdots, Tree_m$. By the strategy, we can get the representative subset C_{sub}, which is the smallest recognition unit in the action recognition task. In our dataset, the total number of labelled C_{sub} is 50. The total number of pictures is 27,328, 80% of which are training samples and 20% of which are testing samples. Fig. 2 visualizes the scatter of the dataset.

B. Baselines Based on Object Detection

There are many convolutional neural networks as image classifier backbones for object detection task. The Inception-v2 network proposed by C. Szegedy *et al.* had joined the

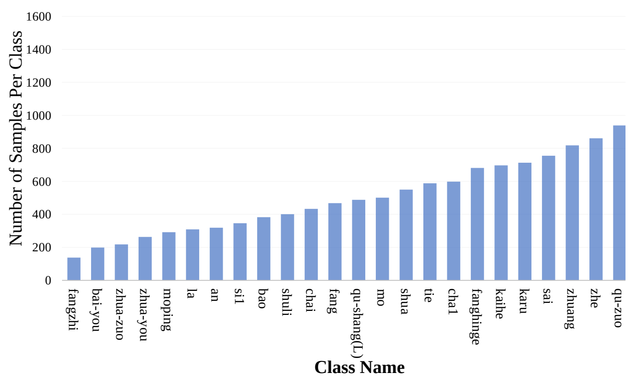


Fig. 2. The Scatter of action dataset D. The D includes a total of 50 action categories, and only partial action category information is shown here.

batch normalization layer and replaced one 5×5 convolution layer with two 3×3 convolution layers [30]. The improvement reduced the internal covariate shift and increased the robustness of the model. But it also brought about an increase, 25% of the weights and 30% of the computational cost. In 2017, MobileNet used the Depthwise Separable Convolutions to compress the model [31]. In this method, its speed has been improved a lot. However, the decrease of accuracy may be a problem. Additionally, the Deep Residual Networks (ResNet), proposed by Kaiming He's research team in 2015, is the champion of the ImageNet challenge [32]. ResNet with the design of "bottleneck" block can have up to 1000 or more network layer. The deeper the residual network is and the more complex the computation, the degradation of network performance will be. Thus, here only consider the 50-player and 100-player ResNet. Compared with the former two, ResNet has state of the art performance, not only feature extraction.

Detector algorithms have evolved for nearly 20 years, except for feature extraction networks. In 2014, G. Gkioxari, R. Girshick and J. Malik took the lead to break the deadlocks and proposed the Regions with CNN features (R-CNN) for object detection [22]. In 2015, R. Girshick proposed Fast R-CNN detector, which simultaneously trained a detector and a bounding box regressor under the same network congurations [33]. Then Kaiming He's research team presented Faster R-CNN detector shortly after the Fast R-CNN [24]. Their speed was the one-stage detecor inferior to themselves. SSD with high detection speed and simplicity, the second one-stage detector, was proposed by W. Liu *et al.* in 2015 [15]. Meanwhile, there were still some achievements in some respects. For example, T. Y. Linetal *et al.* proposed RetinaNet in 2017 to solve class imbalance problem [19], while Z. C. Cai and N. Vasconcelos designed Cascade R-CNN in 2019 to resolve IOU thresholds selection which is difficult to match in train and inference stage [29].

C. *Whole Scheme for Action Recognition*

Besides datasets and training models, we need to design a scheme to complete the entire detection and recognition task for this scenario. Fig. 3 is the whole scheme for action detection and recognition algorithm in a video frame. In this way, the algorithm can be run on the input video frame to detect all actions in the image.

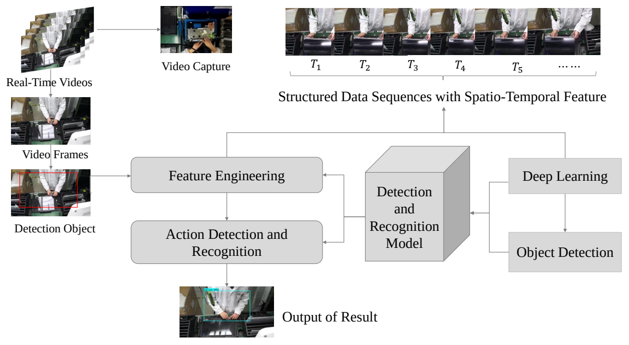

Real-Time Videos
Video Capture
T_1 T_2 T_3 T_4 T_5 ……
Structured Data Sequences with Spatio-Temporal Feature
Video Frames
Feature Engineering
Detection and Recognition Model
Deep Learning
Detection Object
Action Detection and Recognition
Object Detection
Output of Result

Fig. 3. Whole Scheme for Action Detection and Recognition.

Firstly, video datasets are collected and processed after capturing real-time video. Then we convert video data into frame-by-frame image data, which are required to run the detection model after being discretized into sequences with spatio-temporal features. Based on the action association model of standard specifications in industrial scenario, we make relevant detection and recognition decision-making schemes relying on the features of time series. Eventually, we form a model with spatio-temporal feature for action semantic recognition by integrating big data technology and deep learning methods in the whole scheme for detection and recognition. In the way, we can achieve analysis of action specification, i.e., the normalized semantics of the action.

IV. Training Refinements

In this section, we mainly provide various ablation experiments to improve the impact of models. As empirically investigated in this work, we can note some strategies for model. Unless explicitly mentioned otherwise, mAP@.5IOU and AR(average recall) are used as evaluation metrics.

A. *Basic Parameters Refinements*

Anchors play a prominent role in the extraction of action semantics. A reasonable value enables the model to help learn more detailed features. Here the value is usually set to 9, 12, 15. The relatively subtle action translation of the actors and the strong correlation between frames in this scenario, thus the regression of the detection frame is easy to learn. The threshold of NMS also has a non-trivial impact to the accuracy. Generally, the threshold of positive samples fluctuates between 0.6 and 1.0. We take 0.6, 0.7 and 0.8 to observe the change of accuracy.

Ablation experiments are conducted on nine benchmark architecture, namely M_1-M_9 in Table. I. The total number of iterations is 100,000. Setting Th_{NMS} too low will give rise to noisy detections, while letting Th_{NMS} be too high will produce too few positive samples and over-fitting easily. Increasing the value of anchors with too small or too large size also leads to a descendent accuracy. As empirically investigated in this work, the $M_5(N_{an} = 12, Th_{NMS} = 0.7)$ achieves better or comparable performance under evaluation metrics.

B. *Learning Rate Refinements*

Learning rate adjustment is crucial to training. Considering the complexity of the model and the training cost, we mainly

TABLE I
THE PERFORMANCE OF MODEL ARCHITECTURES M_1-M_9 ON THE DATASET

Models	Basic Parameters	mAP@.5IOU	AR
M_1	$N_{an} = 9, Th_{NMS} = 0.6$	96.13%	87.65%
M_2	$N_{an} = 9, Th_{NMS} = 0.7$	96.35%	87.43%
M_3	$N_{an} = 9, Th_{NMS} = 0.8$	94.27%	86.19%
M_4	$N_{an} = 12, Th_{NMS} = 0.6$	96.48%	86.55%
M_5	$N_{an} = 12, Th_{NMS} = 0.7$	**96.63%**	**86.67%**
M_6	$N_{an} = 12, Th_{NMS} = 0.8$	93.08%	87.79%
M_7	$N_{an} = 15, Th_{NMS} = 0.6$	95.88%	87.51%
M_8	$N_{an} = 15, Th_{NMS} = 0.7$	95.84%	87.95%
M_9	$N_{an} = 15, Th_{NMS} = 0.8$	94.31%	87.81%

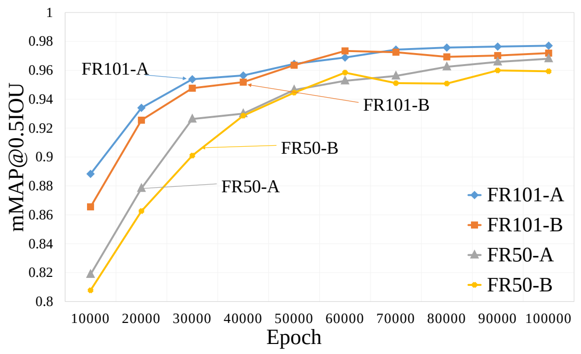


Fig. 4. Performance of two model algorithms on step decay and without step decay. FR101-A, FR101-B, FR50-A and FR50-B represent whether Faster R-CNN + ResNet-101 and Faster R-CNN + ResNet-50 models use step decay respectively, where A means use step decay, B Indicates that it is not used.

optimize the learning rates of piecewise constant decay and cosine learning rate decay.

The first type of learning rate, piecewise constant decay learning rate, is to set different learning constants in a pre-defined training interval. The interval setting needs to be adjusted according to the sample size. The larger the sample size, the smaller the interval. We mainly use Faster R-CNN+ResNet-50, and Faster R-CNN+ResNet-101 for tuning experiments. The best value is usually 0.0003. Fig. 4 shows the validation accuracy curve with regard to the four schedules of Faster R-CNN+ResNet-50 and huge advantage in terms of average accuracy and average recognition time. In most cases, the performance of the two model algorithms is better in the way of step decay. The comparison of learning rate between the way of step decay and without step decay are illustrated in Fig. 5. At the beginning, them adopt a larger learning rate and keep a constant value. But then the former starts to decay the learning rate as the epoch in the process of approaching the optimal solution. In this case, the former can continue to reduce the loss with finer iterations and potentially improve the training convergence.

Then cosine learning rate is the second object we optimize. To improve the convergence speed of gradient schemes and deal with gradient-free optimization of multimodal functions, I. Loshchilov and F. Hutter proposed a cosine annealing strategy in 2016 [34]. As mentioned in [35], within the i-th run at batch $m(1 \leq i \leq m)$, the learning rate η_m^i of each batch is decayed by cosine annealing, as shown in Eq. 1.

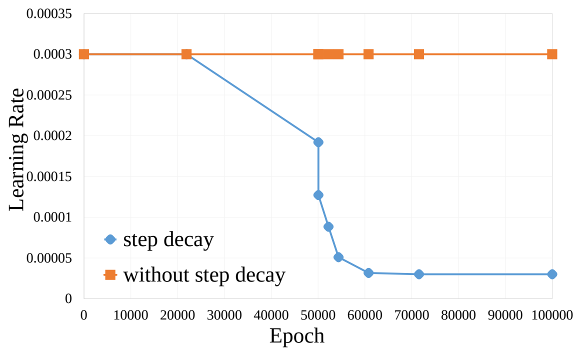


Fig. 5. Learning Rate Schedule.

$$\eta_m^i = \eta_{min}^i + \frac{1}{2}(\eta_{max}^i - \eta_{min}^i)\left(1 + cos\left(\frac{T_0}{T_i}\ \pi\right)\right), \quad (1)$$

where η_{min}^i and η_{max}^i are the maximum and minimum learning rates, and T_0 indicates the number of epochs that has been executed since the last restart. We take some random values to T_0 in each batch of iteration updates. The [36] proposed to a simplified expression in the case of ignoring the preheating stage, as the following:

$$\eta_m = \frac{1}{2}\left(1 + cos\left(\frac{m}{T}\pi\right)\right)\eta. \quad (2)$$

Cosine decay slowly reduces the learning rate at the beginning, then almost linearly decreases in the middle, and again slows down at the end. The stage simulates the process of rapid convergence of potential learning rate, and the cycle effect brought by restart may lead to the improvement of accuracy.

In this way, the basic learning rate is recorded as $base_{lr}$, the warmup learning rate is recorded as $warmup_{lr}$. The total epoch is fixed to 100000, and the warm up steps is 2000. Generally, we set $base_{lr}$ to 0.04, 0.004, 0.0004, $warmup_{lr}$ to 0.013333, 0.0013333 and 0.00013333 respectively. It is worth noting that the actual training requires that $base_{lr}$ be greater than or equal to $warmup_{lr}$. In other words, the value of $warmup_{lr}$ is only set to 0.0001333 when $base_{lr} = 0.004$. Different basic learning rates correspond to the same warm up learning rate. Finally, we get three different SSD+ResNet-50 model architectures with different learning rates. Fig. 6 and Fig. 7 show the trend of performance and learning rate in three methods, LR1-LR3: $base_{lr} = 0.04, warmup_{lr} = 0.00013333$; $base_{lr} = 0.004, warmup_{lr} = 0.00013333$ and $base_{lr} = 0.0004, warmup_{lr} = 0.00013333$. From these graphs above, we can conclude that the model with $warmup_{lr} = 0.004$ converges faster and performs better in this scene.

The impact of $warmup_{lr}$ is non-trivial. Then we need to obtain a suitable value of $warmup_{lr}$. We fixed $base_{lr} = 0.004$, $warmup_{lr}$=0.0013333 or 0.00013333 respectively to further the experiment. In Fig. 8 and Fig. 9, it is seen intuitively the performance of model and the learning rate with two case: $warmup_{lr} = 0.004$ and $warmup_{lr} = 0.0004$. In the same epoch, the learning speed decreases faster in $warmup_{lr} = 0.004$, the convergence speed of the model is larger, and the precision is higher.

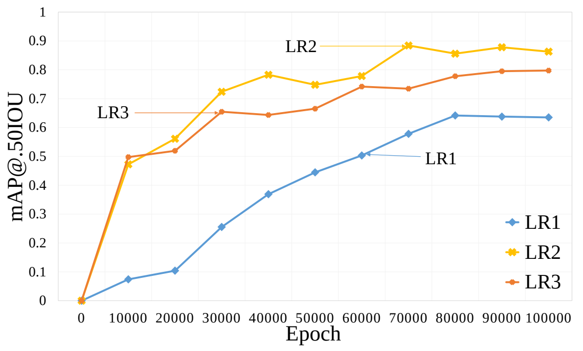


Fig. 6. Performance of SSD+ResNet-50 Model on Different Base Learning Rates.

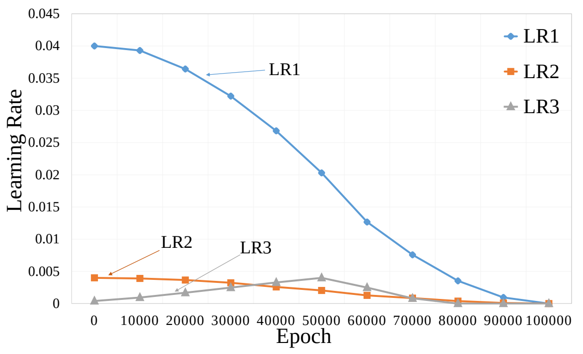


Fig. 7. Visualization of Cosine Decay Learning Rate with Warm-up on Validation.

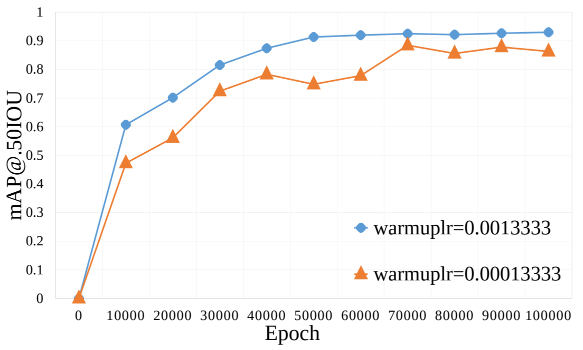


Fig. 8. Performance of SSD+ResNet-50 Model on Different Warmup Learning Rates.

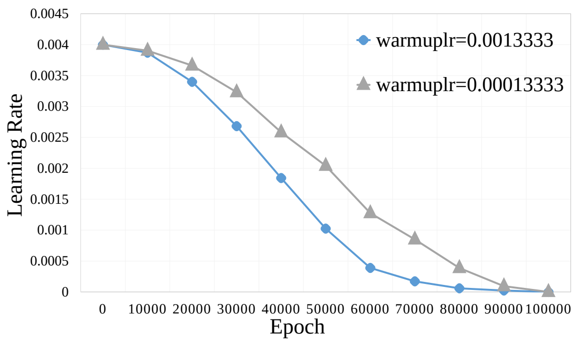


Fig. 9. Visualization of Cosine Decay Learning Rate with Warm-up on Validation.

V. Experiments and Results

In this section, it mainly involves the relevant empirical process and results. Tensorflow is a deep learning framework to help us complete experiments. The physical environment is three Alienware servers with 32G memory, Winow10 operating system and GTX 1070i dual-core GPU.

TABLE II
Details of Data Sample Distribution

Data	Total pictures	Total categories	Number of training sets
A	3611	12	2889
B	6369	22	5040
C	9050	30	7229
D	11770	40	9407
E	27328	50	21785

A. Training and Test Results

There are 21785 pictures in the training samples and 5543 pictures in the test samples in our data. Fig. 2 shows us the imbalance of datasets. Therefore, the simple random sampling might not be representative. This paper divides all into four subsamples by stratified sampling according to the category of actions, which can ensure that each class is represented with approximately equal proportions in both the training set and test set.

To verify and enhance the generalization ability of the model, we divide the dataset into 50 subsets in the context. The first subset is then selected based on the video of different lengths until there are a total of 10 subsets. Finally, we merge the 10 subsets and get the first novel subsample, which is cast to A. The rest is step-wise done in the same manner so that we can get the second subsample. We record the subsample merged by the second subsample and the first subsample as data B, which exist overlap. Each subsample obtained in each round needs to be merged with the previous subsample. In this manner, we can get data C and D separately. The whole dataset is recorded as E in Table. II. In addition, we also made some minor adjustments existed overlaps to the dataset to ensure the purity of each basic action type. The training strategy can also improve the accuracy of our model and reduce the training time.

In Table. III, SSD+MobileNet-v1, SSD+Inception-v2, SSD+ResNet-50, Faster R-CNN+Inception, Faster R-CNN+ResNet-50, Faster R-CNN+ResNet-101 and Faster R-CNN+ResNet-101 are abbreviated respectively: A1, A2, A3, A4, A5, A6 and A7. From A to E, the inadaptability of SSD+Inception-v2 model is particularly prominent with the increase of action categories, while Faster R-CNN+ResNet-50, Faster R-CNN+ResNet-101 and R-FCN+ResNet-101 have high accuracy comparing to others.

B. Verification Results and Result Analysis

Furthermore, verifying these models in real job scenario is essential for selecting the best model. Here we preform the verification experiment based on the whole scheme for detection and recognition described in Sec. III-C. Table. IV shows that R-FCN+ResNet-101 is the state-of-the-art model under the trade-off between recognition time and accuracy. Compared with Faster R-CNN+ResNet-50 and Faster R-CNN+ResNet-101, R-FCN+ResNet-101 model' accuracy is

TABLE III
THE PERFORMANCE OF EACH MODEL ON DATA A TO E

Model \ Data		A	B	C	D	E
A1	mAP	0.9770	0.9611	0.8981	0.8911	0.8793
	AR	0.8687	0.8595	0.8248	0.8245	0.8149
A2	mAP	0.9348	0.8888	0.8851	0.8387	0.7261
	AR	0.7236	0.6767	0.6808	0.6691	0.5711
A3	mAP	0.9806	0.9712	0.9246	0.9143	0.9110
	AR	0.7236	0.8429	0.8252	0.8168	0.8038
A4	mAP	0.9995	0.9958	0.9829	0.9626	0.9454
	AR	0.9073	0.9002	0.8760	0.8608	0.8410
A5	mAP	**0.9988**	**0.9978**	**0.9824**	**0.9792**	**0.9648**
	AR	**0.9256**	**0.9049**	**0.8861**	**0.8759**	**0.8670**
A6	mAP	**0.9992**	**0.9977**	**0.9925**	**0.9823**	**0.9769**
	AR	**0.9381**	**0.9297**	**0.9276**	**0.9158**	**0.9099**
A7	mAP	**0.9883**	**0.9945**	**0.9857**	**0.9709**	**0.9645**
	AR	**0.9285**	**0.9254**	**0.9088**	**0.8943**	**0.8860**

TABLE IV
THE AVERAGE RECOGNITION ACCURACY AND TIME OF EACH MODEL

Model	Average accuracy	Average time(s)
SSD+MobileNet-v1	52.30%	0.11
SSD+Inception-v2	72.00%	0.01
SSD+ResNet-50	83.30%	0.19
Faster RCNN+Incption-v2	89.90%	0.18
Faster RCNN+ResNet-50	96.48%	0.24
Faster RCNN+ResNet-101	96.67%	0.40
R-FCN+ResNet-101	**96.45%**	**0.23**

almost at the same level as they are. But its detection speed is superior to the formers, which is thanks to the design of position-sensitive network. Thus R-FCN+ResNet-101 is preferable for industrial pipeline operation scenario.

As empirically investigated results above, a holistic analysis can be done. The research methods in this paper can be applied to other analogous scenarios that require semantic recognition of the action in a short time, e.g. sign language recognition scenario. In addition, the front-end video capture device is easy to deploy, eliminating these dedicated action capture devices, which provides the possibility to deploy standard action work at large scale and low cost. The scene has a single background and continuous, repetitive, decomposable action flow. The actor's single semantic action time is extremely short (less than 2s), requiring fast positioning and extraction of action semantics. The main contribution is getting a domain model with continuous and spatio-temporal features based on object detection algorithms. Then, here we extract the two-dimensional features of the video stream and obtain continuous action information to be recognized through it. Finally, these recognition actions are stitched back to semantic information. The solution can also be migrated to analogous industrial scenarios with these features.

For these similar scenarios, given a video sequence $V \in S^{n_x \times n_y}$, we confirm that these postures at the video level with these features firstly. Then, according to the calculation of Eq. 3, the V is cut into several clips $V_{clip} \in S^{k_x \times k_y}(0 < k_x < n_x,\ \ 0 < k_y < n_y)$. Such a sampled V_{clip} approximates actions contained in a 2D cube of size $k_x \times k_y$. For each frame $V(:,:,T_i)$ of V with $T = (T_1, T_2, T_3, ...T_t)$, the way we compute it is like this:

$$V_{result}(:,:,T_i) = |V(:,:,T_i) - V(:,:,T_i + \Delta t)|, \quad (3)$$

where $V_{result}(:,:,T_i)$ denotes the resulting sequence and Δt is the distance of the two-time frame to compute the difference.

Based on these features, the idea of divide and conquer can simplify and decompose complex action objects into a tree-like structure as in Sec. III-A mentioned, rather than directly detecting all at one run. Then extracting the two-dimensional information from V_{result} can obtain the input $Input = (data, label)$,where $data = \{D_1, D_2, D_3 \ldots\ldots, Dn\}$ with n instances and $label = \{L_1, L_2, L_3 \cdots\cdots, L_n\}$ with n labels. In this case, we can get discretized and recognizable action sequences C. The sequence C exhibits a certain continuity in space while having a certain degree of dispersion in time. In the learning and inference stage, feeding the input set Input into our model, we get the spatial action information $A^S = \{A_1^S, A_2^S, A_3^S......A_n^S\}$ with n instances. Then $A^{S\times T} = \{A_1^{S\times T}, A_2^{S\times T}, A_3^{S\times T}......A_n^{S\times T}\}$ is obtained on the time stream T from A^S. In this way, we can further understand the semantic action and complete the analysis of action specification through inference.

In Fig. 11, it is not surprising to expect that exploiting the domain model as well could further advance the state of the art. Here we use two sets to calculate the union operation to visualize the model:

$$STM = FE_{feature} \cup D_{detector}. \quad (4)$$

STM is the union result of feature extraction networks $FE_{feature}$ and detectors $D_{detector}$, i.e., a domain model with continuous and spatio-temporal features for these application scenarios. $FE_{feature}$ represents convolutional neural networks with deep mining feature information capabilities, such as ResNet. $D_{detector}$ involves detectors that satisfy the following core features:

1) It has a fully convolutional image classifier backbone that can extract end-to-end training and can extract object semantic information, which is mainly used to extract candidate regions in this paper. It can ensure translation-invariance in image classification. Image classification requires the network to have translation-invariance. Most of the current convolutional neural networks can do very well in classification. e.g., Faster R-CNN and F-RCN. Faster R-CNN used RPN network to select the proposal region, and then used Fast R-CNN to classify. To ensure translation-variance in object detection, Kaiming He's research team extracted about 2000 candidate regions according to RPN and calculated the loss function using the full connection layer [24]. There are some convolutions in front of

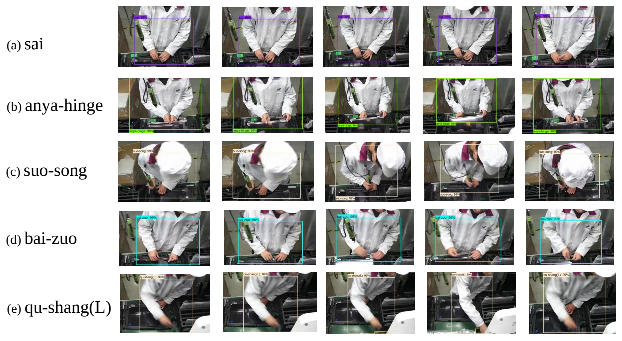


Fig. 10. Detection and Recognition Results. (a)-(e) are the detection effects of sai, anya hinge, suo-song, bai-zuo and qu-shang(L) respectively.

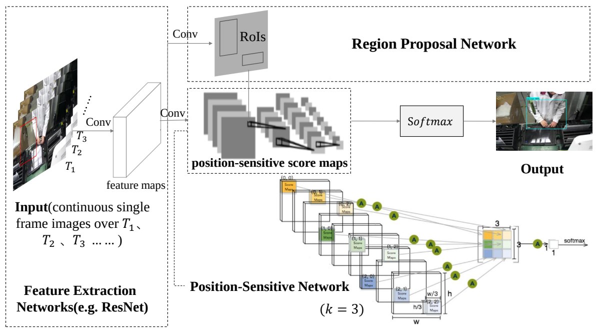

Conv
RoIs
Region Proposal Network
Conv
Conv
feature maps
Softmax
position-sensitive score maps
Output
Input(continuous single frame images over T_1、T_2、T_3 ……)
Feature Extraction Networks(e.g. ResNet)
Position-Sensitive Network
(k = 3)
softmax

Fig. 11. Overall architecture of the STM Model.

the RoI pooling layer for Faster R-CNN, which are translationally invariant. Faster R-CNN and F-RCN behave in this aspect are outstanding.

2) For translation-variance in object detection, it is necessary to have a structure with a strong position-sensitive feature that is fast in calculation. e.g., R-FCN. In such a scenario, the similarity between adjacent frames and each class is extremely high, the degree of translation between frames is also extremely small. Thus, the detector requires an accurate response to the target's translation, i.e., a structure position-sensitive feature, and it does not incur additional computational overhead. Although Faster R-CNN also maintains stability as much as possible in translation-invariance. But once the RoI pooling in Faster R-CNN is inserted, the latter network structure loses the feature. Then these candidate regions have a lot of feature redundancy, resulting a costly per-region subnetwork.In this regard, J. F. Dai, Y. Li, K. M. He and J. Sun proposed a position-sensitive network to generate detection boxes [25]. It can address a dilemma between translation-invariance in image classification and translation-variance in object detection. There are no parameters to learn after the RoI layer, thereby increasing the speed of the model by 2.5 to 20 times.

In R-FCN, a RoI region is divided into a $K \times K$ ($K = 3$ in Fig.11) grid in equal proportion, as in [25]. Each position(x, y, W, H) is recorded as a bin, which notes the coding of the position-sensitive object corresponding to the grid. For a RoI region with a size of $W \times H$, the size of each bin is about $\frac{W}{K} \times \frac{H}{K}$. In the (i, j)-th $bin(i \geq 0, j < K)$, a position-sensitivity RoI pooling is defined, which is to compute the mean value of each bin in the position-sensitive score map:

$$r_c\,(i,j|\theta) = \frac{1}{n}\sum Z_{i,j,c}(x + x_0,\ y + y_0 \mid \theta). \qquad (5)$$

where θ is all the parameters to be learned for the whole network. $r_c\,(i,j|\theta)$ represents the response value of the object of the label at (i,j)-th bin. $Z_{i,j,c}(x + x_0,\ y + y_0 \mid \theta)$, in position-sensitivity, refers the partial features of $\left\lfloor i\frac{W}{K} \right\rfloor \leq x < \left\lceil (i+1)\,\frac{W}{K} \right\rceil$ and $\left\lfloor i\frac{H}{K} \right\rfloor \leq y < \left\lceil (i+1)\,\frac{H}{K} \right\rceil$ for each bin. The network fuses the position information of the object to the RoI Pooling layer, reducing the additional learning parameters and thus improving the detection speed extremely.

VI. Conclusion

We present a domain model with continuous and spatio-temporal features based on object detection algorithm for this scene, which is helpful in worker assembly line operation scene detection, recognition and analyzing. Through it, we get the spatial action information through video streams with only two-dimensional information, and then complete the action semantic recognition based on these temporal action streams. The method is applicable to scenarios that require semantic recognition of the action in a short time, without any dedicated action capture devices. In the future, we will explore more valuable information in the analysis of action standardization, such as the recognition of workers' equipment and type classification. Besides, the lack of abundant action types in our dataset leads to the poor generalization ability of our recognition algorithm model. Thus, we will extend our work to other industrial fields, which will make it more generalized.

References

[1] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, 1997, pp. 928–934.

[2] B. Matthew, O. Nuria, and P. Alex, "Coupled hidden markov models for complex action recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997, pp. 994–999.

[3] N. Oliver, E. Horvitz, and A. Garg, "Layered representations for human activity recognition," in *Fourth IEEE International Conference on Multimodal Interfaces*, 2002, pp. 3–8.

[4] C. Sminchisescu, A. Kanaujia, and D. Metaxas, "Conditional models for contextual human motion recognition," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 210–220, 2006.

[5] L. Yann, B. Yoshua, and H. Geoffrey, "Deep learning," *Nature*, vol. 521, no. 2, pp. 436–444, 2015.

[6] S. Lin, J. Kui, C. Tsung-Han, F. Yuqiang, W. Gang, and yan shuicheng, "DL-SFA: deeply-learned slow feature analysis for action recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 2625–2632.

[7] A. Mathis, P. Mamidanna, K. M. Cury, and et al., "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning," *Nature Neuroscience*, vol. 21, pp. 1281–1289, 2018.

[8] E. Mark, V. G. Luc, W. C. K. I., W. John, and Z. Andrew, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[9] E. Mark, E. S. M. Ali, V. G. Luc, W. C. K. I., W. John, and Z. Andrew, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

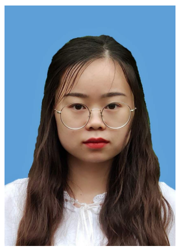
[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[11] L. Tsung-Yi, M. Michael, B. Serge, H. James, P. Pietro, R. Deva, D. Piotr, and Z. Larry, "Microsoft COCO: common objects in context," in *ECCV*, 2014, pp. 740–755.

[12] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
[13] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, 2005, pp. 886–893.
[14] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester, “Object detection with grammar models,” in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, ser. NIPS’11, 2011, pp. 442–450.
[15] L. Wei, A. Dragomir, E. Dumitru, S. Christian, R. Scott, F. Cheng-Yang, and B. A. C., “SSD: single shot multibox detector,” in *2016 Computer Vision(ECCV)*. Springer International Publishing, 2016, pp. 21–37.
[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
[17] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” *CoRR*, vol. abs/1612.08242, 2016.
[18] R. Joseph and F. Ali, “YOLOv3: an incremental improvement,” *CoRR*, vol. abs/1804.02767, 2018.
[19] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollr, “Focal loss for dense object detection,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007.
[20] L. Hei and D. Jia, “CornerNet: detecting objects as paired keypoints,” in *2018 Computer Vision (ECCV)*. Springer International Publishing, 2018, pp. 765–781.
[21] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “CenterNet: keypoint triplets for object detection,” *CoRR*, vol. abs/1904.08189, 2019.
[22] G. Gkioxari, R. Girshick, and J. Malik, “Contextual action recognition with r*cnn,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1080–1088.
[23] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *CoRR*, vol. abs/1406.4729, pp. 346–361, 2014.
[24] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
[25] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: object detection via region-based fully convolutional networks,” *CoRR*, vol. abs/1605.06409, 2016.
[26] K. He, G. Gkioxari, P. Dollar, and R. B. Girshick, “Mask R-CNN,” *CoRR*, vol. abs/1703.06870, 2017.
[27] Y. Li, Y. Chen, N. Wang, and Z. Zhang, “Scale-aware trident networks for object detection,” *CoRR*, vol. abs/1901.01892, 2019.
[28] G. Ghiasi, T. Lin, R. Pang, and Q. V. Le, “NAS-FPN: learning scalable feature pyramid architecture for object detection,” *CoRR*, vol. abs/1904.07392, 2019.
[29] Z. Cai and N. Vasconcelos, “Cascade R-CNN: delving into high quality object detection,” *CoRR*, vol. abs/1712.00726, 2017.
[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.
[31] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *CoRR*, vol. abs/1704.04861, 2017.
[32] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
[33] R. Girshick, “Fast R-CNN,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.
[34] I. Loshchilov and F. Hutter, “SGDR: stochastic gradient descent with restarts,” *CoRR*, vol. abs/1608.03983, 2016.
[35] P. Goyal, P. Dollár, R. B. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, “Accurate, large minibatch SGD: training imagenet in 1 hour,” *CoRR*, vol. abs/1706.02677, 2017.
[36] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, “Bag of tricks for image classification with convolutional neural networks,” *CoRR*, vol. abs/1812.01187, 2018.

Cheng Chen obtained her B.S. degree in Software Engineering from Yangtze University, Jingzhou, Hubei, China, in 2018. Currently she is purchasing her M.S. degree at College of Computer Science, Yangtze University, Jingzhou, Hubei, China. Her current research interests are about Computer vision in general, with specific interests in image processing, object detection and recognition, model compression.

Yang Wang got her B.S. degree from Yangtze University, Jingzhou, Hubei, China, in 2016. Currently purchasing her M.S. degree at College of Computer Science, Yangtze University, Jingzhou, Hubei, China. Her current research interests are object recognition and classification in general, especially action sequence analysis.



Ke Yi received B.S. degree in Mathematics and Applied Mathematics and M.S. degree in Software Engineering from Yangtze University, Jingzhou, Hubei, China, in 2015 and 2019, respectively. The current research is about deep learning, which mainly includes computer vision and model compression acceleration.

Tongxi Wang received his B.S. and M.S. degrees from Chengdu University of Technology, Chengdu, China, in 1994 and 2007, respectively. He is currently an associate professor with the Department of Software Engineering, Yangtze University. He serves the Head of the Department of Software Engineering, College of Computer Science and an expert in doctoral and master thesis review of the Degree Center of the Ministry of Education. He is also a member of the Chinese Association of Artificial Intelligence (CAAI). His current research interests are about big data and artificial intelligence technology, intelligent computing.

Hua Xiang is currently a lecturer in School of Computer Science, Yangtze University. Before that, he received his B.S. degree from Yangtze University, Jingzhou, Hubei, China, in 2002, and M.S. degree from WuHan University, Hubei, Wuhan, China, in 2009. His current research interests include artificial intelligence, software engineering, and big data technologies.