Skeleton-based Action Recognition for Industrial Packing Process

Zhenhui Chen[1], Haiyang Hu[1], Zhongjin Li[1], Xingchen Qi[1],
Haiping Zhang[1], Hua Hu[1,2] and Victor Chang[3]
[1]*School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China*
[2]*School of Information Science and Engineering, Hangzhou Normal University, Hangzhou, China*
[3]*School of Computing & Digital Technologies, Teesside University, Middlesbrough, U.K.*

Keywords: Skeleton-based Action Recognition, Image Classification, Industrial Packing Process.

Abstract: The applications of action recognition in real-world scenarios are challenging. Although state-of-the-art methods have demonstrated good performance on large scale datasets, we still face complex practical problems and inappropriate models. In this work, we propose a novel local image directed graph neural network (LI-DGNN) to solve a real-world production scenario problem which is the completeness identification of accessories during the range hood packing process in a kitchen appliance manufacturing workshop. LI-DGNN integrates skeleton-based action recognition and local image classification to make good use of both human skeleton data and appearance information for action recognition. The experimental results demonstrate the high recognition accuracy and good generalization ability on the range hood packing dataset (RHPD) which is generated in the industrial packing process. The results can meet the recognition requirements in the actual industrial production process.

1 INTRODUCTION

Deep learning has been successfully applied in computer vision, including image classification, object detection, semantic segmentation and pose estimation. Action recognition, which is a fundamental task in the fields of video understanding and intelligent monitoring, has also benefited from the deep learning technique. However, it has not been solved completely. As an attempt of using deep learning in action recognition, it is the first time that the two-stream model (Simonyan et al., 2014) surpasses previous hand-crafted features methods. It decomposes videos into spatial appearance and temporal motions information by modelling individual frame and stacked optical flow maps respectively with CNNs then fusing the results of these two streams. The concept of two-stream has a profound impact on subsequent development in this domain. Another important branch of approach to CNN-based video modelling is represented by 3D convolutional networks (C3D) (Tran et al., 2015) introduced by Du Tran et al. They regarded whole video as three-dimensional data and used 3D CNNs to classify videos which is resemble 2D CNNs to classify images. Notably, Carreira and Zisserman introduced a model (I3D) (Carreira et al., 2017) that combines two-stream processing and 3D convolutions. I3D significantly improved action recognition results on UCF101 (Soomro et al., 2012) and HMDB51 (Kuehne et al., 2011). More recently, kernel factorization strategy which replaced 3D convolution with a 2D convolution (in space) followed by a 1D convolution (in time) has shown its unique advantages in reducing computation and improving accuracy, such as P3D (Qiu et al., 2017), R(2+1)D (Tran et al., 2018), S3D (Xie et al., 2018) and CSN (Tran et al., 2019). The above methods are all based on RGB frames or dense optical flow maps of videos. In addition, skeleton-based approach plays an important role in the field of action recognition. Skeleton data of human can robustly accommodate dynamic circumstances and complex backgrounds which is benefit from the highly accurate depth sensors (e.g. Kinetics) and pose estimation algorithms. Skeleton-based action recognition can pay attention to the posture and motion process more intuitively and accurately of the human body. To some extent, it is more suitable for both subtle and complex action recognition.

The applications of action recognition and detection in real-world scenarios are not as prevalent

Chen, Z., Hu, H., Li, Z., Qi, X., Zhang, H., Hu, H. and Chang, V.
Skeleton-based Action Recognition for Industrial Packing Process.
DOI: 10.5220/0009340800360045
In *Proceedings of the 5th International Conference on Internet of Things, Big Data and Security (IoTBDS 2020)*, pages 36-45
ISBN: 978-989-758-426-8

as object detection and semantic segmentation. The main reasons lie in three folds: First, existing models

Figure 1: The range hood packing scene in the workshop of kitchen appliance manufacturing enterprise Robam.

are not applicable to recognition of specific human actions in real scenarios. Second, the video processing is really computationally inefficient. Finally, the models trained on current datasets cannot be transfer directly into real-world tasks.

In this work, for solving the problem of identifying the completeness of the accessories during the range hood packing process of kitchen appliance manufacturing enterprise Robam, we investigate a novel method called Local Image Directed Graph Neural Networks (LI-DGNN) to model both human skeleton data and frames in a surveillance video. Figure 1 shows the range hood packing assembly line. Workers need to pack two kinds of parts (connecting pipe and carton containing tools) together into range hood packing box. Although the numbers of range hoods and accessories on this assembly line are recorded, there are still cases where workers forgot to put accessories into the box. In this way, the cost of finding the missing box later is extremely high. The skeleton data is collected by a real-time pose estimation algorithm AlphaPose. Specifically, a skeleton-based action recognition algorithm directed graph neural networks (DGNN) (Shi et al., 2019) is embedded to classify worker's activities, then local images of worker's hands taken from skeleton points are further analyzed to identify what is in the hand with image classification model ResNeXt (Xie et al. 2017).

In order to train our data-driven learning framework and evaluate its advantage, we create a range hood packing dataset (RHPD). The dataset contains 4 different modalities of data: RGB videos, 2D skeleton data, local images and parts' bounding boxes data. We experimentally show that our proposed LI-DGNN model outperforms other object detection and action recognition methods in both cost saving and accuracy on RHPD. The main contributions of our work lie in three folds: (1) A novel local image directed graph neural network is proposed to solve the problem of accessories deficiency recognition in a real-world industrial production scenario. (2) We make a range hood packing dataset (RHPD) to train and evaluate our model which is finally deployed online. (3) On the dataset, our model exceeds the mainstream action recognition methods in both recognition accuracy and generalization ability.

2 RELATED WORK

Before using deep learning to solve action recognition problem, improved dense trajectories (iDT) (Wang and Schmid, 2013) is universally acknowledged as the state-of-the-art approach the state-of-the-art which describes video as several hand-designed features which follow dense trajectories, computed by optical flow. The features include histogram of oriented gradients (HOG), histogram of optical flow (HOF) and motion boundary histogram (MBH). MBH is a gradient-based feature, separately computed on the horizontal and vertical components of optical flow. Then an SVM classifier is used to classify Fisher Vector encoded by those features.

Driven by the breakthrough of deep learning in still-image recognition, some active researches have been dedicated to the design of deep networks for videos. Moreover, researchers have put considerable efforts to utilize convolutional networks to model videos. Two-stream model was introduced by Simonyan and Zisserman, who proposed to extract deep features from both RGB frames and dense optical flow maps then fuse results predicted from these two modalities features. Feichtenhofer et al. enhanced the two-stream networks using the ResNet architecture (He et al., 2016) and additional features fusion between streams (Feichtenhofer et al., 2016). Temporal segment networks (Wang et al., 2016) are proposed to model long-range temporal structure over the whole video by applying two-stream framework to multiple video segments.

Another influential approach to learning spatiotemporal features of video is represented by 3D convolutional networks (C3D). Compared with 2D CNNs, C3D has a stronger learning ability benefit from its increased parameters. But it is difficult to learn good features on small scale datasets. This has been proved in the work of I3D, in which C3D was re-implemented and pretrained on a larger video

dataset Kinetics, finetuned on UCF-101 and HMDB-51, and better results have been obtained. I3D proposed by Carreira and Zisserman combined two-stream framework and 3D convolutions to model both videos and optical flow maps. It also uses successful ImageNet (Russakovsky et al., 2015) architecture designs and even their parameters by inflating 2D ConvNet to 3D. I3D raised the performance of action recognition to a new level. Recently, in order to accelerate training and convergence, kernel factorization has become a new trend. Typically, Du Tran et al. introduced R(2+1)D model which decompose 3D convolution into a 2D spatial convolution followed by 1D temporal convolution within a ResNet architecture. It compensates for the defect of 3D convolutions without losing performance.

In addition to directly model video frames and optical flow maps, extracting features of dynamic skeletons to predict human actions has also become an active research area. The dynamic skeleton data represents human action as a sequence of coordinates of the major body joints in videos. It can be easily captured by the existing pose estimation algorithms (Cao et al. 2017; He, Gkioxari et al. 2017; Fang et al. 2017) or the depth sensors. Skeleton-based action recognition can focus on human body activities without interferences of body scales changes, motion speeds, camera viewpoints and complicated background. The traditional methods of processing skeleton data represent the human pose based on hand-craft features. However due to the challenge of manually designing a good feature extractor, which impedes the way of yielding satisfying results. Methods based on deep learning has been proved to be superior to the traditional methods. There are mainly three frameworks for deep-learning-based methods: sequence-based methods, image-based methods and graph-based methods.

Sequence-based methods use RNN-based architectures (Shahroudy et al., 2016; Liu et al., 2016; Song et al., 2017; Li et al., 2018) to model skeleton data represented as a sequence of joints with designed traversal strategy. Image-based methods mainly use CNNs which are applied successfully in the area of image classification (Du, Fu et al., 2015; Kim and Reiter, 2017) to model a pseudo-image constructed from skeleton data. Instead of representing the coordinates of joints as sequences or pseudo-images, it is more intuitive to organize the data as a graph with joints as vertexes and bones as edges (Tang et al., 2018; Shi, Zhang et al. 2018). Yan et al. investigated graph convolutional networks (GCN) to model both spatial and temporal information of dynamic skeleton data which was constructed as a graph (Yan et al. 2018). The strategies of two-stream and adaptive graph construction were integrated to GCN framework for the purpose to boost the performance.

However, the existing approaches cannot perfectly solve our practical problem. Most methods based on RGB frames or optical flow are computationally costly and high time consuming on both training and inference stages. This impedes real-time recognition in production deployment. Moreover, these methods are also data-hungry. That means it is difficult for the model to obtain a good accuracy and generalization ability which was trained on a small dataset. Skeleton data is naturally robust and skeleton-based methods can capture subtle human body movement stably. But it is at the expense of discarding the important appearance contents and background information. We proposed local image directed graph neural networks (LI-DGNN) which can capture both RGB contents and human skeleton data by combining Local image classification and skeleton-based action recognition. Our experiments show that LI-DGNN framework yields best performance and generalization ability on RHPD compared with existing mainstream action recognition approaches (specifically two-stream networks, C3D, I3D, R(2+1)D, ST-GCN (Yan et al. 2018) and DGNN).

3 METHOD

In this section, we describe the LI-DGNN framework which solved our problem of lacking the integrity recognition of range hood accessories on the packing line. Our model integrates directed graph neural networks (DGNN) and local image classification model ResNeXt which can capture both dynamic skeleton data and appearance information to predict human actions efficiently. The pipeline of the proposed LI-DGNN demonstrated in the following (Figure 2). First, skeleton data is extracted from a video stream by pose estimation algorithm AlphaPose in real time, then a sliding window is introduced which will be performed on the skeleton sequence to obtain available samples. The data from samples will be fed into DGNN to predict current action. Finally using ResNeXt model to classify local images which are cropped from the area of skeletal points of the worker's hands for further identification. The final integrity result is determined by the results of the two models.

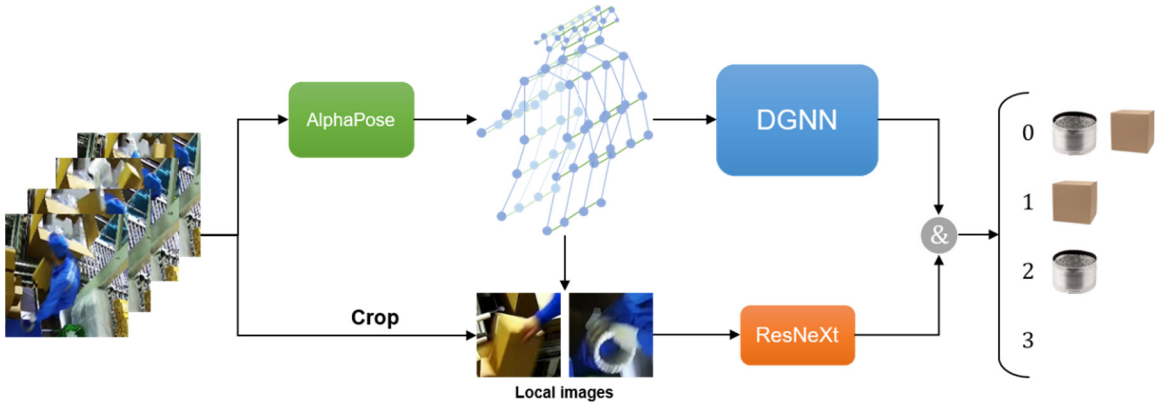

Figure 2: Pipeline of our Local Image Directed Graph Neural Networks (LI-DGNN) framework.

3.1 Regional Multi-person Pose Estimation

To extract skeleton data accurately and fast, we use AlphaPose model to detect the locations of human skeleton key points on each frame of video stream. AlphaPose is a regional multi-person pose estimation framework (RMPE) (Fang et al. 2017) which is introduced to facilitate pose estimation in the presence of inaccurate bounding boxes and redundant detections of human detectors from single-person pose estimator (SPPE). There are primarily two techniques contributing to this: symmetric spatial transformer network (SSTN) and parametric pose non-maximum-suppression (NMS).

3.1.1 Symmetric STN and Parallel SPPE

SSTN includes spatial transformer network (STN), single-person pose estimator (SPPE) and spatial de-transformer network (SDTN). SDTN is the inverse of STN. STN receives human bounding boxes from human detector, and SDTN generates pose proposals (Figure 3).

The spatial transformer network (STN) (Jaderberg et al., 2015) has demonstrated excellent performance in selecting region of interests automatically. STN embedded in networks performs a 2D affine transformation. Therefore, the networks possess the functions of cropping, translation, scaling and rotation on the input feature map. It can learn how to transform the shape of data by training. Parallel SPPE is another single-person pose estimator branch. The parameters of this branch are fixed in the training phase. The output of this SPPE branch is directly compared to labels of centre-located ground truth poses and back-propagate centre-located pose errors to the STN module. If the pose extracted by the STN is not centre-located, the parallel branch will back-propagate large errors. In this way, STN is transformed to the correct region to extract high-quality human body region images.

3.1.2 Parametric Pose NMS

For the purpose of eliminate redundant pose estimations, non-maximum suppression (NMS) method is required. Different from traditional NMS methods, parametric pose NMS which embedded in AlphaPose has four parameters in the eliminate criterion function. The parameters are optimized to achieve the maximal mAP for the validation set in training phase. The parameters will be fixed once convergence is achieved which will be used in the testing phase.

The signs $\{(k_i^1, c_i^1), \dots, (k_i^m, c_i^m)\}$ denotes the pose P_i, with m joints, where k_i^j and c_i^j are the j^{th} location and confidence score of joints separately. AlphaPose defined pose distance metric $d(P_i, P_j|\Lambda, \lambda)$ to calculate the pose similarity, and a threshold η is introduced as elimination criterion, where Λ is a parameter set of function $d(\cdot)$. The elimination criterion is demonstrated as follows

$$f(P_i, P_j|\Lambda, \eta) = 1[d(P_i, P_j|\Lambda, \lambda) \leq \eta] \quad (1)$$

The output of $f(\cdot)$ should be 1, if $d(\cdot)$ is smaller than η. This further indicates that we should eliminate pose P_i, which is because of the redundancy with reference pose P_j.

The definition of the distance function $d_{pose}(P_i, P_j)$ is the sum of two distances, soft matching of joints and spatial distance between parts. Definition of the soft matching function is written as follows

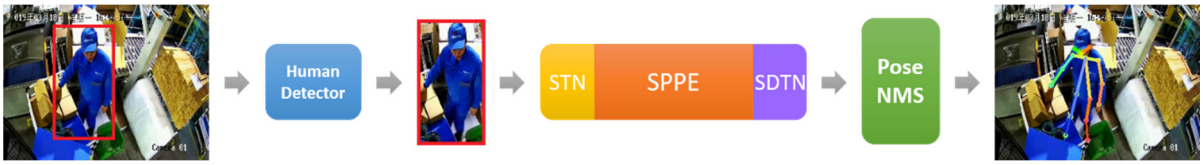

Figure 3: Structure of regional multi-person pose estimation model.

$$K_{Sim}(P_i, P_j|\sigma_1) = \begin{cases} \sum_n tanh\frac{c_i^n}{\sigma_1} \cdot tanh\frac{c_j^n}{\sigma_1}, & if\ k_j^n\ is\ within\ \mathcal{B}(k_i^n) \\ 0 & otherwise \end{cases} \quad (2)$$

where $\mathcal{B}(k_i^n)$ is a bounding box centred at k_i^n, and each dimension of $\mathcal{B}(k_i^n)$ is defined as 1/10 of the original box $\mathcal{B}_i$ for pose P_i. The $tanh$ operation deletes poses with low-confidence scores. The output will be close to 1 if the confidence scores of two corresponding joints are both fairly high. The number of joints that match between poses is counted softly by this distance. We define the spatial distance between parts is as follows

$$H_{Sim}(P_i, P_j|\sigma_2) = \sum_n exp\left[-\frac{(k_i^n - k_j^n)^2}{\sigma_2}\right] \quad (3)$$

The final distance function can be written as

$$d(P_i, P_j|\Lambda, \lambda) = K_{Sim}(P_i, P_j|\sigma_1) + \lambda H_{Sim}(P_i, P_j|\sigma_2) \quad (4)$$

where λ is a weight that balancing the two distances and $\Lambda = \{\sigma_1, \sigma_2, \lambda\}$. These parameters were determined in a data-driven manner.

3.2 Directed Graph Neural Networks

The raw skeleton data extracted from AlphaPose are a sequence of frames, each of which contains a set of joint coordinates. Directed graph neural networks (DGNN) can capture spatial and motion information of a sequence of skeleton frames to recognize human action. The spatial information (the joints and bones) are represented as the vertexes and edges within a directed acyclic graph. The difference of coordinates of joints and bones in consecutive frames are regarded as the motion information with the same graph. The spatial and motion information are fed into a two-stream DGNN framework respectively to extract features for action recognition.

3.2.1 Bone and Motion Information

The importance of combining the joint information and bone information together has been emphasized by previous works for skeleton-based action recognition. In this algorithm, the bone is represented as difference value of coordinates between two connected joints. To extract both the movements of joints and the deformations of bones, the movements of joints is calculated as the difference of coordinates along the temporal dimension and the deformation of bones is represented as the difference of the vectors for the same bone in consecutive frames. Mathematically, the joint in raw data is represented as a vector with two elements, i.e., its x-coordinate and y-coordinate. There are two joints $j_1 = (x_1, y_1)$ and $j_2 = (y_2, y_2)$, so the bone linked from j_1 to j_2 is formulated as the difference of the two joint vectors, i.e., $b_{j_1,j_2} = (x_1 - x_2, y_1 - y_2)$. The movement of joint j in time t is calculated as $m_{j_t} = j_{t+1} - j_t$. The deformation of bone is defined similarly as $m_{b_t} = b_{t+1} - b_t$.

3.2.2 Graph Construction

DGNN needs a directed acyclic graph (DAG) depicted by the skeleton data with the joints as vertexes and bones as edges. The direction of each edge is determined by the distance between the vertex and the root vertex, where the vertex closer to the root vertex points to the vertex farther from the root vertex. This representation is intuitive since the human body is naturally an articulated structure. The joints closer to the center of the human body always control adjacent joints which are farther from the center. As with the spatial information modeling, the motion information is represented as directed acyclic graphs similarly to the movements of the joints as vertexes instead of simply joints and the deformation of bones as edges instead of the bones.

Formally, for each vertex j_p, defines the edge heading to it as the incoming b_p^- and the edge emitting from it as the outgoing edge b_p^+. For each directed edge b_q, b_q^s and b_q^t denote the source vertex and the target vertex of the edge respectively. In this

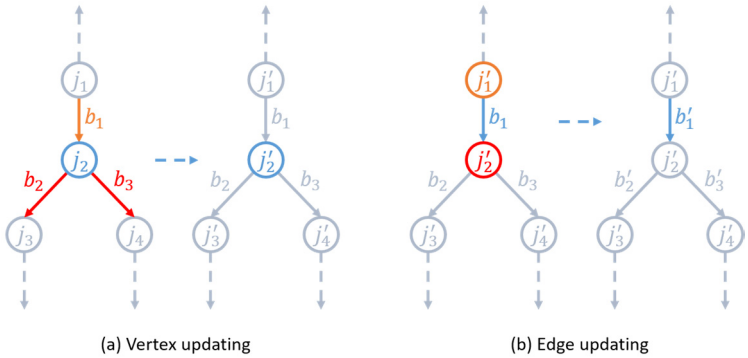

Figure 4: Directed graph network block, vertex and edge updating.

way, a skeleton-based frame can be formulated as a directed graph $G = (J, B)$, where J is a set of vertexes (joints) and B is a set of directed edges (bones). A skeleton-based video is a sequence of frames that can be formulated as $S = \{G_1, G_2, \dots, G_T\}$, where T means the length of the video. Similarly, the motion information of the skeleton-based video is represented as a sequence of directed acyclic graphs $S = \{G_1^m, G_2^m, \dots, G_T^m\}$, where $G^m = (J^m, B^m)$, $J^m = \{m_{J_q}\}_{q=0,\dots,N_j}$ and $B^m = \{m_{b_p}\}_{p=0,\dots,N_b}$. Then, the spatial and motion graphs are fed into two-stream DGNN respectively to make the prediction for the action label. Finally, two networks are fused by adding the output scores of the softmax layer.

3.2.3 Directed Graph Network Block

The directed graph network (DGN) block is the basic block for a directed graph neural network; it includes two updating functions, h^j and h^b, and two aggregation functions, g^{b^-} and g^{b^+}. The updating function is used to update the attributes of vertexes and edges which is according to their adjacent edges and vertexes (Figure 4). We use the aggregation function to make aggregation of the attributes which are contained in multiple incoming (outgoing) edges connected to one vertex. The aggregation function should stay invariant to the permutation of its inputs since there are no apparent orders for these edges and is able to take variable numbers of arguments, such as the average pooling, max pooling and elementwise summation. Formally, this process is designed in details as follows:

$$\bar{b}_p^- = g^{b^-}(\mathrm{B}_i^-) \quad (5)$$

$$\bar{b}_p^+ = g^{b^+}(\mathrm{B}_p^+) \quad (6)$$

$$j_p' = h^j([j_p, \bar{b}_p^-, \bar{b}_p^+]) \quad (7)$$

$$j_p' = h^j([j_p, \bar{b}_p^-, \bar{b}_p^+]) \quad (8)$$

Where the concatenation operation is denoted as $[\cdot]$. j' and b' are the updated versions of j and b, respectively.

For temporal information modelling, inspired by the pseudo-3D and R(2+1)D CNN which model the spatial information with the 2D convolutions and then model the temporal information with the 1D convolutions, after updating the spatial information of joints and bones or the motion information of movements of joints and deformations of bones in each DGN block, we apply a 1D convolution along the temporal dimension to model the temporal information.

The whole DGNN model is similar to I3D model which is based on two-stream framework and use 3D CNNs to model RGB frames in spatial stream and optical flow maps in temporal stream. DGNN also has spatial and temporal streams. The RGB frames and optical flow maps are replaced with joints and bones data in the spatial stream and movements of joints and deformations of bones in the temporal stream respectively. Instead of 3D CNN block, DGNN use directed graph network block and 1D convolution along the temporal dimension to model spatio-temporal information. Lastly, two stream networks are fused by adding the output scores of the softmax layer.

3.3 Local Image Classification

The human action class can be recognized by DGNN with skeleton data. However, in this process, we do not utilize appearance information of video. Furthermore, even if the action class of the worker who performed a specific task are identified, we might also encounter a “feint” situation that the workers do the same body movement without holding the accessories in his hands. To further improve the correct recognition rate and make use of the content

Table 1: Object detection and action recognition methods accuracy on RHPD.

Method	Dataset	mAP / top-1 (Line-1)	mAP / top-1 (Line-2)
Faster RCNN	RHPD accessories detection	92.1	86.6
YOLO v3	RHPD accessories detection	94.7	55.9
Two-Stream	RHPD video	87.5	79.6
ST-GCN	RHPD skeleton	**96.4**	**94.3**

information efficiently, we crop local images according to coordinates of wrists in skeleton joints estimated by AlphaPose, then classify the images to identify whether there are accessories in his hands and whether the worker put them into packing box. The problem cannot be solved completely if only the images of hands area of the worker are tracked and recognized, which is due to various factors like motion-induced blur, camera viewpoints, object occlusion and interference of backgrounds. Only combining action recognition and the local image classification can identify that the accessories are packed into the box, which has been proved in our experiments.

There are many methods for image classification. ResNeXt which was introduced by Xie et al. provides the advantages of high accuracy and fast speed. ResNeXt enhanced ResNet by substituting residual block with grouped convolutions block and introduced a super parameter "cardinality". This structure is similar to the inception block (Szegedy et al., 2015) but there are only 1×1 and 3×3 convolutional kernels on the paths of the grouped convolutions block. The value of cardinality represents the number of paths in the block. The input feature map is divided into cardinality paths according to the channels, then the outputs of all paths are aggregate by summation. This process is called aggregated residual transformations. Specifically, ResNeXt-50 (32×4d) is embedded into our LI-DGNN framework. The value of cardinality is 32, and 4d represents each path receives 4 channels feature maps in first grouped convolutions block.

4 EXPERIMENTS

4.1 The Range Hood Packing Dataset

The range hood packing dataset (RHPD) is made for solving the problem of lacking integrity in the recognition of accessories during range hood packing process of kitchen appliance manufacturing enterprise Robam. We recorded videos of the process on two production lines by several cameras set up with different viewpoints, then labelled them manually, extracted human skeletons and perform data augmentation. The dataset contains 4 different modalities of data: RGB videos, 2D skeleton data, local images and parts' bounding boxes data. The data of two assembly lines are separated and noted as line-1 and line-2. The line-1 is used for training and validation. The line-2 is only used in the test phase to evaluate the generalization ability of model on the other production lines.

Video Dataset. Video data contains 3000 clips of worker's actions during packing process. The actions are divided into three classes (grab, feint and other) which represent packing accessories, doing packing action without accessories in hands and other actions during packing respectively. Each clip is 25 FPS and lasts about 5 seconds. There are 2000 clips in line-1 and 1000 clips in line-2.

Skeleton Dataset. Skeleton data is extracted by pose estimation algorithm AlphaPose. The classes and numbers are the same as video dataset. The length of skeleton sequence detected in the whole clip is the product of video time and frame rate. AlphaPose predicts 18 joints for each person as labelled. Each joint is represented as 2D coordinate (X, Y) and corresponding confidence score C.

Local Image Dataset. There are 6000 images cropped from local areas according to coordinates of wrists in skeleton joints. The dataset contains 3 classes (connecting pipe, tool carton and other). Each class has 2000 images and the size of them are all 350×350.

Accessories Detection Dataset. This is a dataset that we originally created which uses object detection methods to detect and track accessories in the whole packing process. It contains 4000 images and bounding boxes data of accessories in the images. There two classes of objects we annotated (connecting pipes and tool boxes). The images are also captured from two production lines by several cameras from different viewpoints.

Table 2: RGB-based and skeleton-based methods action recognition performance on RHPD.

Method	Dataset	Line-1	Line-2	Line-1(no feint)	Line-2(no feint)
Two-Stream	RHPD video	87.5	79.6	91.3	82.6
C3D	RHPD video	73.7	63.8	85.2	77.4
I3D	RHPD video	84.1	79.5	97.5	89.6
R(2+1)D	RHPD video	**89.9**	**81.2**	**98.9**	85.7
ST-GCN	RHPD skeleton	-	-	96.4	94.3
DGNN	RHPD skeleton	-	-	98.5	**96.2**

4.2 Implementation Details

Data Augmentation. We use both spatial and temporal jittering for augmentation. Specifically, we apply random cropping, horizontal flipping, rotation and colour jittering on videos and images. We augment skeleton data with translation and rotation. Temporal jittering is also applied during training by randomly selecting a starting frame and decoding T frames.

Training. All experiments are conducted on the PyTorch deep learning framework (Paszke et al., 2019). We train directed graph neural networks (DGNN) and image classification model ResNeXt on RHPD skeleton and local image dataset respectively. For DGNN, we initialize the network weights from the model pretrained on Kinetics-Skeleton dataset. The batch size is 64. Stochastic gradient descent (SGD) with momentum (0.9) is applied as the optimization strategy. The learning rate is initialized as 0.1 and reduced by a factor of 10 in epoch 30 and 60. The model is trained with 150 epochs. For ResNeXt, we use ImageNet-pretrained model as base networks. Then, we finetune it with the batch size (32), learning schedule (Adam with an initial learning rate 3e-4 and reduce by a factor 0.9) and training 1000 epochs.

4.3 Exploration Study

In this section, we examine the effectiveness of object detection approach, action recognition method and proposed LI-DGNN framework. The recognition accuracy is used as the evaluation indicator.

4.3.1 Object Detection vs. Action Recognition

For the problem of accessories deficiency recognition, a direct way is to detect and track the accessories in video stream. In this part of experiments, we investigate and compare object detection methods Faster RCNN (Ren et al., 2017), YOLO v3 (Joseph and Ali, 2018) and action recognition methods two-stream model, ST-GCN on our RHPD. We report mAP for object detection methods and Top-1 accuracy for action recognition methods (Table 1). Since ST-GCN is skeleton-based and cannot capture appearance information of accessories, we merge grab and feint data into one class. ST-GCN classifies only two classes of actions.

Table 1 shows the methods generally provide good validation performance on line-1. But on the line-2, YOLO v3 yields a poor generalization ability. Faster RCNN and two-stream model yield 5.5% and 7.9% precision gaps between line-1 and line-2. Notably, skeleton-based action recognition approach ST-GCN has a best validation and test accuracy on line-1 and line-2. The test accuracy is only 2.1% lower than the validation accuracy. This shows ST-GCN has a good generalization on other production lines. In the following sections, we will focus on investigating action recognition approaches.

4.3.2 RGB-based vs. Skeleton-based

In this section, we implement several mainstream action recognition methods (two-stream, C3D, I3D, R(2+1)D, ST-GCN and DGNN) and evaluate on the RHPD video and skeleton datasets. These methods are divided into two camps, RGB-based and skeleton-based. Since the feint and grab actions are very similar, we not only evaluate on the original datasets (three classes), but also remove the feint class to evaluate the model in order to eliminate the interference of fake actions. The results are summarized in Table 2.

On the original sets, we test the RGB-based methods, but most of them do not perform very well. R(2+1)D model yields best top-1 accuracy, which is only 81.2% on line-2, though. After eliminating the interference of fake actions, the overall accuracy has been improved significantly. This indicates that RGB-based methods are hard to capture small differences of actions. Compared with RGB-based methods, skeleton-based methods obtain more stable recognition accuracy on both line-1 and line-2 sets

Table 3: Comparisons with mainstream action recognition methods on RHPD.

method	dataset	Line-1	Line-2
Two-Stream	RHPD video	87.5	79.6
C3D	RHPD video	73.7	63.8
I3D	RHPD video	84.1	79.5
R(2+1)D	RHPD video	89.9	81.2
Local Image	RHPD local image	78.1	64.2
LI-DGNN	RHPD skeleton	**98.0**	**96.7**

without feint actions. DGNN obtains best test performance and 1.9% higher than ST-GCN on line-2.

4.4 Local Image Is Necessary

Our experiments have proved that skeleton-based action recognition methods have better recognition performance and generalization ability to human body actions in similar scenes. In this part, we evaluate the benefits of local image strategy to skeleton-based action recognition method DGNN, and compare our LI-DGNN with other approaches on the complete RHPD set. The results are shown in Table 3. We report top-1 accuracy as the evaluation indicator of the local image strategy. We crop local images of the area of worker's hands for each frame of the video clip, then if more than 10 images in 25 consecutive frames are identified accessory class and the wrist joint coordinate move towards the packing box of the range hood, it is judged that the accessories are successfully packed. Two accessories are tested separately and the final result is average of the two accessories.

There are several noteworthy observations. First, using the local image strategy solely does not work well, probably because of object occlusion, viewpoint changes and the harsh condition that accessories must be tracked continuously to determine packing successfully. Second, our LI-DGNN model has a surprisingly good effect in further verifying the recognized human actions with local image strategy. It raises the performance to 98.0% on line-1 and 96.7% on line-2. This result is similar to DGNN perform on no feint set, which indicates local image strategy can identify fake actions well in our model. Third, LI-DGNN also has a good generalization ability on line-2 set which the model does not seen before. Our model can be easily migrated to other new production lines for deployment. Finally, compared with other mainstream RGB-based action recognition methods, our LI-DGNN performs best on the complete RHPD dataset and this performance can meet the actual production requirements.

5 CONCLUSIONS

In this work, we proposed a novel Local Image Directed Graph Neural Network (LI-DGNN) which integrates Directed Graph Neural Networks and local image classification model ResNeXt to make good use of skeleton data and local images for action recognition. In view of the actual production scenario problem of identifying the completeness of the accessories during the range hood packing process in a kitchen appliance manufacturing workshop, we made an RHPD dataset. We implement the mainstream RGB-based and skeleton-based action recognition methods, and train them on RHPD. Compared through results by experiments, our LI-DGNN framework shows the best recognition accuracy on RHPD dataset and has good generalization ability, which can meet the recognition requirements in the actual production process.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (No. 61802095, 61572162), the Zhejiang Provincial Key Science and Technology Project Foundation (No. 2018C01012). Zhongjin Li is the corresponding author.

REFERENCES

Cao, Z., Simon, T., Wei, S, Sheikh, Y., 2017. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. *IEEE Conference on Computer Vision and Pattern Recognition,* 1302-1310.

Carreira, J., Zisserman, A., 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition,* 4724-4733.

Du, Y., Fu, Y., Wang, L., 2015. Skeleton Based Action Recognition with Convolutional Neural Network. *2015 3rd IAPR Asian Conference on Pattern Recognition,* 579–583.

Fang, H., Xie, S., Tai, Y., Lu, C., 2017. RMPE: Regional Multi-person Pose Estimation. *IEEE International Conference on Computer Vision,* 2353-2362.

Feichtenhofer, C., Pinz, A., Zisserman, A., 2016. Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition,* 1933-1941.

He, K., Gkioxari, G., Dollár, P., Girshick, R. B., 2017. Mask R-CNN. *IEEE International Conference on Computer Vision,* 2980-2988.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition,* 770-778.

Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K., 2015. Spatial Transformer Networks. *Conference on Neural Information Processing Systems*, 2017–2025.

Joseph, R., Ali, F., 2018. YOLOv3: An Incremental Improvement. *arXiv preprint arXiv: 1804.02767*.

Kim T. S., Reiter, A., 2017. Interpretable 3d Human Action Analysis with Temporal Convolutional Networks. *Computer Vision and Pattern Recognition Workshops,* 1623–1631.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T., 2011. HMDB: A Large Video Database for Human Motion Recognition. *IEEE International Conference on Computer Vision,* 2556-2563.

Li, L., Zheng, W., Zhang, Z., Huang, Y., Wang, L., 2018. Skeleton-Based Relational Modeling for Action Recognition. *arXiv preprint arXiv:1805.02556*.

Liu, J., Shahroudy, A., Xu, D., Wang, G., 2016. Spatio-Temporal LSTM with Trust Gates for 3d Human Action Recognition. *European Conference on Computer Vision,* 816-833.

Paszke, A., Gross, S., Massa, F. et al., 2015. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, 8024-8035.

Qiu, Z., Yao, T., Mei, T., 2017. Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition,* 5534-5542.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., Li, F., 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision,* 211-252.

Shahroudy, A., Liu, J., Ng, T., Wang, G., 2016. NTU RGB+D: A Large Scale Dataset for 3d Human Activity Analysis. *IEEE Conference on Computer Vision and Pattern Recognition,* 1010-1019.

Shi, L., Zhang, Y., Cheng, J., Lu, H., 2018. Non-Local Graph Convolutional Networks for Skeleton-Based Action Recognition. *arXiv preprint arXiv:1805.07694*.

Shi, L., Zhang, Y., Cheng, J., Lu, H., 2019. Skeleton-Based Action Recognition with Directed Graph Neural Networks. *IEEE Conference on Computer Vision and Pattern Recognition,* 7912-7921.

Simonyan, K., Zisserman, A., 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. *Annual Conference on Neural Information Processing Systems 2014,* 568-576.

Song, S., Lan, C., Xing, J., Zeng, W., Liu, J., 2017. An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. *AAAI Conference on Artificial Intelligence,* 4263-4270.

Soomro, K., Zamir, A. R., Shah, M., 2012. A Dataset of 101 Human Actions Classes from Videos in The Wild. *arXiv preprint arXiv:1212.0402*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, 1-9.

Tang, Y., Tian, Y., Lu, J., Li, P., Zhou, J., 2018. Deep Progressive Reinforcement Learning for Skeleton-Based Action Recognition. *IEEE Conference on Computer Vision and Pattern Recognition,* 5323-5332.

Tran, D., Bourdev, L. D., Fergus, R., Torresani, L., Paluri, M., 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. *2015 IEEE International Conference on Computer Vision,* 4489-4497.

Tran, D., Wang, H., Torresani, L., Feiszli, M., 2019. Video Classification with Channel-Separated Convolutional Networks. *arXiv preprint arXiv: 1904.02811*.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M., 2018. A Closer Look at Spatiotemporal Convolutions for Action Recognition. *2018 IEEE Conference on Computer Vision and Pattern Recognition,* 6450-6459.

Wang, H., Schmid, C., 2013. Action Recognition with Improved Trajectories. *IEEE International Conference on Computer Vision,* 3551-3558.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Gool, L. V., 2016. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. *European Conference on Computer Vision,* 20-36.

Xie, S., Girshick, R. B., Dollár, P., Tu, Z., He, K., 2017. Aggregated Residual Transformations for Deep Neural Networks. *IEEE Conference on Computer Vision and Pattern Recognition,* 5987-5995.

Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K., 2018. Rethinking Spatiotemporal Feature Learning for Video Understanding. *European Conference on Computer Vision,* 318-335.

Yan, S., Xiong, Y., Lin, D., 2018. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. *AAAI Conference on Artificial Intelligence,* 7444-7452.