

# Skeleton-based Action Recognition of People Handling Objects

Sunoh Kim<sup>1</sup>Kimin Yun<sup>2</sup>Jongyoul Park<sup>2</sup>Jin Young Choi<sup>1</sup><sup>1</sup>ASRI, Dept. of Electrical and Computer Eng., Seoul National University, South Korea<sup>2</sup>Electronics and Telecommunications Research Institute (ETRI), South Korea

{suno8386, jychoi}@snu.ac.kr, {kimin.yun, jongyoul}@etri.re.kr

## Abstract

In visual surveillance systems, it is necessary to recognize the behavior of people handling objects such as a phone, a cup, or a plastic bag. In this paper, to address this problem, we propose a new framework for recognizing object-related human actions by graph convolutional networks using human and object poses. In this framework, we construct skeletal graphs of reliable human poses by selectively sampling the informative frames in a video, which include human joints with high confidence scores obtained in pose estimation. The skeletal graphs generated from the sampled frames represent human poses related to the object position in both the spatial and temporal domains, and these graphs are used as inputs to the graph convolutional networks. Through experiments over an open benchmark and our own data sets, we verify the validity of our framework in that our method outperforms the state-of-the-art method for skeleton-based action recognition.

## 1. Introduction

Human action recognition has attracted extensive attention in recent years, due to its many potential applications in video surveillance, human-robot interaction, and so on. Recognizing actions, however, is difficult, not only because of the common challenges of the domain, such as viewpoint changes and occlusions, but also because of the ambiguities of some actions. Therefore, human action recognition is still one of the most challenging tasks for computer vision. To tackle these challenges in human action recognition, many approaches using various modalities like appearance, optical-flows, depth, and skeleton have been proposed. Among them, pose-based representation using the skeleton simplifies learning of the actions by extracting the relevant high-level features. Moreover, the pose-based representation suffers relatively little from the intra-class variances, as the action from actor to actor varies less than in other representations [40]. The development of low-cost depth sensors such as Microsoft Kinect and pose estima-

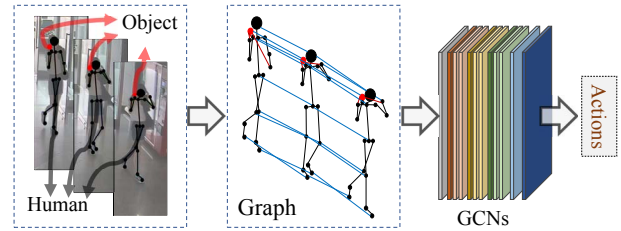


Figure 1. Action recognition via object-related human poses. The pose graph is constructed by using spatial human poses (black dots and lines), spatial object poses (red dots and lines), and temporal connections (blue lines). In spatial and temporal domains, the graph is used as the input to GCNs.

tion algorithms [3, 25, 27, 29] provides an easier way to obtain skeletal information. It also contributes to the success of data-driven methods of skeleton-based action recognition.

The applications of skeleton-based action recognition in real-world scenarios are difficult due to two major problems. First, conventional studies [18, 23, 28, 39] for skeleton-based action recognition usually use constrained datasets, assuming that skeletal data are provided. Visual information in real-world scenarios, however, is captured in an unconstrained environment where human skeletons are not given. Thus, body movement should be interpreted through pose estimation algorithms to obtain skeletal data. Pose estimation is a challenging task, however, due to the rarity of appearance or missing or overlapping body parts. Given that pose estimation results are not always accurate, it becomes crucial to develop approaches dealing with inaccurate poses. Second, most approaches focusing only on human skeletons ignore contextual information like objects and scenes. Most meaningful actions in the real world, though, involve more than one object or person [24]. Therefore, previous methods cannot fully understand the real-world actions.

Motivated by the aforementioned problems, we develop a framework for object-related human action recognition based on graph convolutional networks (OHA-GCN) as de-

picted in Figure 1. To understand the object-related human actions, our method builds the graph structure of both human and object poses. Human poses are extracted using the pose estimation method [3], and object poses are obtained by the pose heatmap and background subtraction. Using the obtained human and object poses, we model: 1) the human pose graph, where the movement of human body is considered; 2) the object-related human pose graph, where the relationship between human and object poses is represented. Our framework employs the two graphs of human poses and object-related human poses, generated in both the spatial and temporal domains.

To make the graphs more structurally complete, we explore a strategy of selecting informative frames, which discards ambiguous frames. After dividing the entire video into segments of equal length, we make a sequence of informative frames by sampling one informative frame from each segment. Using the confidence scores calculated by the pose estimation algorithm, we can decide which human poses have a more complete structure than the others. Then, the two types of the graphs are constructed from the sequence of the informative frames for pose-based representation. Using the human pose graph and object-related human pose graph, OHA-GCN runs in the architecture of the two stream to boost action recognition. The GCNs in each stream of our framework apply convolution filters to each graph. The nodes and their neighbors in each graph are applied to the convolutional filters. In experiments, our framework is validated using an open benchmark and our own data sets that includes an illegal rubbish dumping (IRD) dataset, which compiles realistic data in unconstrained environments.

## 2. Related works

### 2.1. Graph convolutional neural networks

Recent advancements in deep neural networks have led to the development of graph convolutional networks (GCNs) to understand the form of graph structures [2, 6, 7, 19, 22, 26]. GCNs generalize convolutional neural networks (CNNs) from low-dimensional grids of images to high-dimensional domains represented by arbitrarily structured graphs. These tasks are categorized into two main categories: spectral perspective and spatial perspective methods. Spectral perspective methods convert graph data into a spectrum and apply CNNs to the spectral domain [6, 7, 22]. Different from the spectral perspective methods, spatial perspective methods directly use graph convolutions to define parameterized filters [2, 26]. The convolution operation in the spatial perspective resembles the convolution operation on images. We propose a model using GCNs for action recognition following the concepts of the spatial perspective.

### 2.2. Action recognition

Video-based action recognition is a challenging task because the concepts of the actions are highly abstract and consider both spatial and temporal dimensions. Conventional deep-learning-based approaches for action recognition rely mainly on 3D CNNs and two-stream CNNs. The methods based on the 3D CNNs propose a model that directly applies 3D convolution filters to RGB video sequences [4, 14, 33]. 3D CNNs, however, have problems in training due to the explosion of parameters, and their performance is only marginally improved compared to the traditional method [35]. Methods based on the two-stream architecture of CNNs are proposed to solve these problems [8, 30, 36]. In these methods, two CNNs are applied to process appearance and motion (optical flows) independently. These methods outperform traditional methods using hand-crafted features [5, 20, 35]. But, the heavy computation requirements for optical flows become a main limitation of the methods. Our approach is based only on pose information that can be extracted from RGB images. The pose-based representation can reduce computation time and simplify training for the recognition.

### 2.3. Skeleton-based action recognition

Some methods for action recognition are based on skeletal data because human poses are highly relevant to human action. The methods can be categorized mainly into those based on hand-crafted features and those based on deep-learning features. The former methods design several features for understanding human joint motions [9, 34, 37]. For example, the human skeleton is represented as a point in the Lie group [34], encompassing rotations and translations between body parts. With the recent success of deep learning, many methods for skeleton-based action recognition have been proposed. Deep-learning-based methods usually use well-established neural models like recurrent neural networks (RNNs) [23, 28, 44] or CNNs [16, 18]. RNNs use their internal memory to process sequences of input data, making them applicable to skeleton-based action recognition. Zhang *et al.* [44] select geometric features based on distances between human joints and use an RNN-based model. CNN-based methods propose applying the CNN directly by considering the skeleton sequence as an image array. QiuHong *et al.* [16] transform the skeleton sequence into three clips and learn the spatio-temporal feature based on deep CNNs. Still, generalizing the CNNs and RNNs to work on graph structures is challenging because they cannot fully understand the graphs. Since human poses are represented by a graph structure physically, the methods based on the CNNs and RNNs are limited in their ability to use the poses. Recently, the GCN-based methods have been suggested to understand the human poses without loss of graph information [32, 39]. Yan *et al.* [39] propose

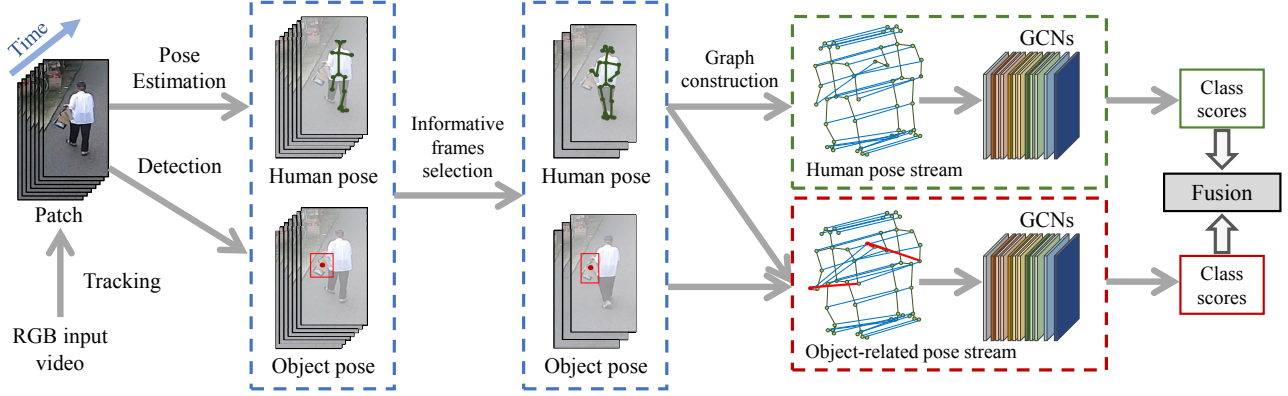


Figure 2. The overall scheme of the proposed framework that includes pipeline modules of tracking and detection for extracting informative human and object poses, pose estimation, graph construction of human and object-related human poses, and GCNs for recognition of object-related human actions.

a ST-GCN that applies the GCNs for skeleton-based action recognition. This method can capture the motion information in dynamic skeleton sequences by constructing spatio-temporal graphs. However, ST-GCN does not consider how to cope with inaccurate poses and cannot understand contextual information. By contrast, our method can build the pose graphs with the contextual information using the reliable poses through the informative joint selection.

### 3. Object-related human action recognition

#### 3.1. Overall scheme

The overall scheme of the proposed framework is depicted in Figure 2. Pose estimation and detection algorithm are applied to patches extracted from RGB video through a tracking algorithm. Then, we make a sequence of informative frames by selecting frames containing relatively accurate poses in the patches. Using a spatial-temporal graph of human poses and object-related human poses from the informative frames, the GCNs perform action recognition in the architecture of the two stream. We revisit the ST-GCN [39] in Section 3.2. We then present how our method detects the objects handled by humans in Section 3.3 and give detailed descriptions of the object-related human action recognition (OHA-GCN) framework. Section 3.4 and Section 3.5 explain how to construct graphs of object-related human poses and apply the graphs on the GCNs, respectively. Finally, we present the strategy of selecting informative frames in Section 3.6.

#### 3.2. ST-GCN [39]

ST-GCN uses a spatial temporal graph to form hierarchical representation of the human skeleton. The spatial temporal graph consists of nodes and edges where each node corresponds to a joint of the human body, and each edge

corresponds to the spatio-temporal connection of the node. In one single frame, they formulate a spatial graph which is represented by human body joints, and the spatial edge of the graph becomes natural connection between human body joints. Then, temporal edges are made by connecting the corresponding joints in consecutive frames.

The skeletal graph with  $N$  joints in one single frame and  $T$  frames of a video is represented by  $G = (V, E)$ , where  $V = \{v_{ti} | t = 1, \dots, T, i = 1, \dots, N\}$  is the node set. In terms of spatial domain, the graph convolution is written as

$$f_{out}(v_{ti}) = \sum_{v_{tj} \in B(v_{ti})} \frac{1}{Z_{ti}(v_{tj})} f_{in}(v_{tj}) \cdot \mathbf{w}(l_{ti}(v_{tj})), \quad (1)$$

where  $f_{in}$  and  $f_{out}$  are the input and output feature map, respectively.  $B(v_{ti})$  is the neighbor set of a node  $v_{ti}$ , where the 1-distance neighbors of the target node  $v_{ti}$  are considered.  $\mathbf{w}$  is the weight function which provides a weight vector to compute inner product with the input feature map. Because the number of the weight vectors is fixed, strategies to partition  $B(v_{ti})$  into a fixed number of subsets have to be designed.  $l_{ti}$  is a function mapping each node in neighborhood of  $v_{ti}$  to its subset label. To prevent the different subsets from unbalancing the output, the cardinality of the subset  $Z_{ti}(v_{tj})$  is used as the normalizing term. In ST-GCN, many partitioning strategies are introduced, such as uni-labeling, distance partitioning, and spatial configuration. Among them, spatial configuration has achieved best performance, as it consider the gravity center of the human body. This strategy divide the neighboring nodes of target node into three subsets: 1) the target node itself; 2) centripetal subset: the nodes that are closer to the gravity center; 3) otherwise centrifugal subset.

The implementation of the graph convolution [19] is adopted for ST-GCN.  $\mathbf{A}$  is the  $N \times N$  adjacency matrix where a graph structure is represented as matrix form by

indicating connections between nodes. To implement ST-GCN with the spatial configuration partitioning, the Eq.( 1) is transformed into

$$\mathbf{f}_{out} = \sum_j \Lambda_j^{-\frac{1}{2}} \mathbf{A}_j \Lambda_j^{-\frac{1}{2}} \mathbf{f}_{in} \mathbf{W}_j, \quad (2)$$

where the input feature map  $\mathbf{f}_{in}$  is represented as a tensor  $(N, T, C)$  dimensions,  $C$  being the number of input channels. Then, the adjacency matrix is divided into three matrices: 1)  $\mathbf{A}_0$ , the self-connection of each node; 2)  $\mathbf{A}_1$ , the connections of the centripetal subset; 3)  $\mathbf{A}_2$ , the connections of the centrifugal subset.  $\mathbf{W}_j$  denotes the weight matrix, where weight vectors of multiple output channels are stacked.  $\Lambda_j$  whose element can be formulated as  $\Lambda_j^{ii} = \sum_k (\mathbf{A}_j^{ik}) + \alpha$  is the diagonal node degree matrix of  $\mathbf{A}_j$ .  $\alpha$  is set to 0.001 to avoid empty rows in  $\mathbf{A}_j$ .

### 3.3. Detection of objects handled by humans

ST-GCN can capture dynamics of human poses by learning both the spatial and temporal patterns of skeletal data. ST-GCN has limitations, however, in understanding object-related human actions because it focuses on only the human poses. For example, the object-related actions, such as dumping, phoning, texting, and smoking require a specific object that acts as a main cue for recognition. Inspired by the issue, we create a new graph structure to form a hierarchical representation of the human and object poses. Before constructing the graphs, it is essential to detect the objects a human is handling.

An easy approach is to find objects that overlap with humans through the object detector, but this approach has several problems. First, objects in the background that are not related to humans, such as parked cars, are regarded as related objects. To solve this problem, we define a class of objects handled by humans and develop a new detector, but the shapes of the objects handled by humans vary widely. The objects handled by humans have a variety of shapes and are small compared to humans. Hence, conventional detectors cannot easily detect the objects handled by humans. Several methods have been presented to use detection networks to detect humans, objects, and their interaction [12, 10]. Still, a pair consisting of a human and an object is required to train the model on each action, and the training operates independently for each frame, not for the whole video.

In this paper, we use a simple and efficient method to find an object handled by a human through the temporal property of a video. Since CCTV cameras are generally fixed and limited to the motions of Pan-Tilt-Zoom, it is possible to find a moving area effectively through background modeling and camera motion compensation [17, 41, 43]. This moving area includes the areas of both the human and the object carried by the humans. Then, object area can then

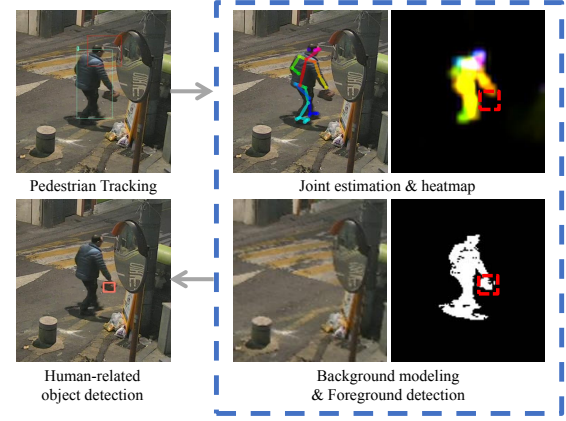


Figure 3. The procedure for detecting the object handled by a human. A moving area is obtained by background subtraction with camera motion compensation and the human area is found by estimated joint and heatmap. Then, the object area is detected by subtracting the human area from the moving area.

be detected by subtracting the human area from the moving area regardless of the shape of the object. Fortunately, a human area can be obtained by using Openpose, a 2D pose estimation algorithm providing the location of each joint. The algorithm provides not only the location but also the confidence scores of the joints and affinities between joints as a heatmap, which can then be used as a human area.

Figure 3 shows the procedure for detecting an object handled by a human. In order to detect the moving region, we use the ViBE algorithm [1] which efficiently builds the background model by storing a set of pixel values from the past. Then, through the *Openpose* algorithm, we get each coordinate of the joint and the joint heatmaps. We aggregate the heatmap for all joints and all joint association into a human region map, which is subtracted from the moving area. The regions around each human hand are selected. Finally, we can construct a graph structure by connecting the center position of this region with the joints of the closest person.

### 3.4. Graph for object-related human poses

We use our detection algorithm and *Openpose* to obtain the coordinates of the human joints and objects for creating a graph structure of the human and object poses. For human skeletal data, the pose estimation algorithm produces 25 joints in a single frame. We then formulate a human pose graph using all the human joints in a video sequence. In addition, the object position obtained in Section 3.3 is added to the human pose graph for modeling our object-related human pose graph. As a result, the object-related human pose graph consists of 26 nodes that represent human and object poses. We can model a spatial graph of the physical connection between joints in the human body and position of the

object in a frame. The corresponding nodes in consecutive frames are then connected to formulate a temporal graph.

To avoid confusion, we use notation similar to that used in the ST-GCN. In our skeletal graph  $G = (V, E)$  for object-related human poses, the node set is denoted as  $V = \{v_{ti} | t = 1, \dots, T, i = 1, \dots, N\}$  with  $N$  joints in one single frame.  $T$  is the number of informative frames sampled from the video. The details of the informative frames are explained in Section 3.6. The edge set is  $E = E_S \cup E_T$ , which depicts two subsets for spatial and temporal connection. The subset for spatial connection is

$$E_S = \{e_{tij} | v_{ti}, v_{tj} \text{ are connected}, (i, j) \in O\}, \quad (3)$$

where  $O$  is the set of the connections for natural human body and the connections between an object and the human body joints. The subset for temporal connection

$$E_T = \{e_{ti} | v_{ti}, v_{(t+1)i} \text{ are connected}\} \quad (4)$$

connects the same nodes which is temporally consecutive. An example of our spatial-temporal skeleton graph is shown in Figure 4.

We consider how to connect the object with the human body. Many strategies of the connection are explored empirically, but we discuss three strategies which are representative of many cases in our view. First, we use the relationship between the whole joint and the object. This method, however, makes the graph-based network inefficient because the graph includes redundant information. Thus, we build a graph structure that is denoted as limbs, focusing on the specific body parts used for interactions. We also introduce a strategy using only human hands, where most object-related human actions occur. The three strategies are denoted as follows: 1) Body: connections between all the human joints and the object; 2) Limbs: connections between the human joints belonging to arms or legs and the object; 3) Hands: connections between the human joints of both hands and the object. Detailed experiments with each strategy are described in Section 4.2.1.

### 3.5. Two stream GCNs with human and object poses

Using only the human pose graph is limited to express only human body motion, ignoring human-object interactions. Based on the graph structure of both human and object-related human poses, we propose a framework of two-stream GCNs called OHA-GCN for object-related human action recognition. The OHA-GCN can boost the recognition for object-related actions, as the human poses and object-related human poses are jointly used. Human-related objects do not, however, appear in every scene. In this regard, the two-stream GCN is designed to use the object-related human pose stream only if human-related objects are found. In the absence of the objects, OHA-GCN

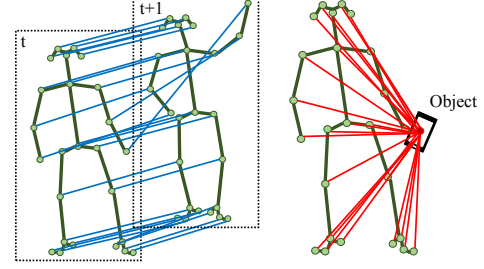


Figure 4. For the construction of our skeletal graph, all of the dots are used as nodes and all of the lines as edges. In each frame, nodes and edges based on the natural human body are denoted as green. **Left:** The graph structure of human poses in terms of both the spatial and temporal domains. Blue lines represent the temporal connections of corresponding nodes in consecutive frames. **Right:** The example of the graph structure for object-related human poses in each frame. A red dot denotes the position of a human-related object. Red lines are the connections between all human body joints and the object.

uses only the human pose stream. If the object is properly detected, OHA-GCN recognizes actions through both the human pose stream and the object-related human pose stream.

The GCN in each stream takes the graph constructed in Section 3.4 as input and produces a video-level prediction for action recognition. The model of GCN is composed of 9 GCN layers. Each layer has 64, 64, 64, 128, 128, 128, 256, 256, and 256 channels, respectively. The global average pooling layer then follows and produces the feature vector. The final outputs are produced by Softmax for action prediction. Categorical cross-entropy loss is adopted for training of the GCN. We adopt evenly averaging Softmax scores for final predictions through empirical evaluation of different aggregation methods such as averaging and maximum.

### 3.6. Informative frame selection strategy

The conventional skeleton-based action recognition methods assume that accurate human skeletons are provided by datasets. Real-world scenarios, however, do not provide skeletal data, and the estimated skeletal data might be inaccurate due to an unusual appearance or missing or overlapping body parts. Likewise, since our methods are based only on RGB sequences, the pose estimation results may be inaccurate. This issue motivates us to introduce a new sampling strategy that selects informative frames. Our informative frame selection strategy extracts a short sequence of sampled frames from a video sequence. The sampled frames preserve the joints enough to understand human poses with high confidence score from the *Openpose* algorithm. Based on confidence scores, we selectively discard frames containing ambiguous human poses. Then our method can select



only those sequences with reliable human pose information.

Initially, the entire video is divided into  $T$  segments  $\{S_1, S_2, \dots, S_T\}$  at equal interval. We construct sequences of  $T$  frames by sampling one frame from each segment. Since consecutive frames have little variation and provide redundant pose information, the sampled sequences can reduce the redundancy. Each segment has  $M$  frames, so  $t$ -th segment  $S_t$  has frames  $\{F_t^1, F_t^2, \dots, F_t^M\}$ , where  $F_t^m$  is  $m$ -th frame in the  $t$ -th segment. After applying the pose estimation algorithm to each frame  $F_t^m$ , we can obtain the confidence score  $c_{ti}^m$  of each human joint  $h_{ti}^m$ , where  $i = 1, \dots, N$  and  $N$  is the number of joints. Then, we select the frame with the highest sum of confidence scores in the estimated joints within a segment because the higher the sum is, the more accurately the human joints are detected. Finally, the sequence of the sampled frames is used directly to construct the human pose graph on both the spatial and temporal domains. The node set of our skeletal graph can be written as

$$V = \{v_{ti} | v_{ti} = h_{ti}^{m^*}, m^* = \arg \max_m \sum_i c_{ti}^m\}. \quad (5)$$

To demonstrate that our frame selection algorithm performs better than the existing sampling strategy, we conduct an ablation study in Section 4.2.1. As a result, OHA-GCN has a relatively complete form of graph that includes reliable joints.

## 4. Experiments

### 4.1. Datasets

The existing datasets [13, 28, 38] are not suitable for use in evaluating our framework to recognize the object-related actions in real-world scenes for two reasons. First, the existing datasets for skeleton-based action recognition were captured in constrained environments. These datasets acquired from sensors are limited to indoor environments. Therefore, the constrained datasets are insufficient for modeling realistic scenes because these datasets cover only a few variations in dynamic environments. Second, these datasets usually include one actor in each image, which is not reflective of the scenes from real-world surveillance. In our evaluation, instead of using the existing datasets, we use two realistic datasets captured in unconstrained real-world environments, which contain only raw video clips without skeleton data. The two datasets are described below.

**IRD dataset:** As for our own dataset, we have constructed an illegal rubbish dumping (IRD) dataset that contains videos from CCTV cameras. The dataset is used for the purpose of the general surveillance, especially monitoring illegal dumping. The original videos are about 10 minutes long, and the resolution of the original video is  $1280 \times 720$  pixels (HD). We made, however, several video

clips for annotations from the original videos by extracting the region of the event. Each video clip, averaging 690 frames in length, is divided into two classes: 1) a garbage dumping action and 2) a normal event. Humans tagged the ground truth, such as the person, the object the person carries, the start time of the action of dumping trash, and the end time. Unlike other existing datasets for skeleton-based action recognition, this dataset is not limited to indoor environments and captured in unconstrained environments. There is a wide variation of conditions in dynamic environments, including viewpoint changes, illumination variance, and occlusions. Most of all, this dataset is suitable for evaluating our method for recognizing object-related human actions. We made a total of 1374 video clips and use 1193 of them for training and 181 for validation.

**ICVL-4 dataset:** The ICVL dataset [15] includes dynamic sub-actions of multiple people at multiple locations in the surveillance scenes. The actions are divided into 13 categories. The dataset contains 497 original videos, and one actor who is one of the people appearing in the video is marked by three sub-actions. For example, one person in the surveillance scene is standing, walking, and smoking. The original video is divided into several video clips containing a single actor. We use only these specific classes related to object-related actions, such as phoning, smoking, texting, and dumping, which will be called ICVL-4 in this paper. The reason we choose the specific classes is that we have to verify the effectiveness of our method in recognizing object-related human actions. The evaluation using the entire class is meaningless, as the proposed strategies are the only major differences between previous skeleton-based approach and OHA-GCN. Therefore, we only conduct experiments with only the data from 4 classes, focusing on the object-related actions. A training set of 805 video clips and a validation set of 86 video clips are used for our experiments.

Both the IRD and ICVL-4 datasets provide only RGB video clips and action labels. As our approach for action recognition is based on the skeleton, we use the pose estimation algorithm [3] and our detection algorithm. Through the algorithms, 25 human joints are extracted, and the position of an object is obtained. There is a total of 26 nodes including both human joints and an object. We make each video clip as a  $(3, T, 26)$  tensor, where the first dimension of the tensor represents 3 channels for the 2D coordinates  $(x, y)$  and confidence scores. If more than two people exist in one video clip, we select the person with the highest sum of confidence scores as the actor for the recognition.  $T$  is the length of the frames, and 26 is the number of nodes here. We use the top-1 accuracy, which is a conventional evaluation method for both datasets.

Table 1. Ablation study of the proposed framework over the IRD and ICVL-4 datasets. The baseline of our approach is ST-GCN [39] which is the state of the art for the skeleton-based action recognition using GCN.

| Method   | IRD (%)      | ICVL-4 (%)   |
|--|--------------|--------------|
| ST-GCN [39] (HP stream + uniform samples)                  | 74.03        | 80.23        |
| OHA-GCN (OHP-body stream + informative samples)            | 71.27        | 72.09        |
| OHA-GCN (OHP-limbs stream + informative samples)           | 73.48        | 76.74        |
| OHA-GCN (OHP-hands stream + uniform samples)               | 74.59        | 79.06        |
| OHA-GCN (OHP-hands stream + informative samples)           | 76.24        | 81.40        |
| OHA-GCN (HP stream + informative samples)                  | 79.56        | 88.37        |
| OHA-GCN (Two stream; HP + OHP-hands + informative samples) | <b>80.11</b> | <b>91.86</b> |

HP: Human Pose, OHP: Object-related Human Pose

## 4.2. Experimental result

### 4.2.1 Ablation study

In the ablation study, we first investigate the role of the proposed strategy of selecting informative frames for skeleton-based action recognition. Second, we examine the strategy for how to connect the object to the human body in OHA-GCN in formulating object-related pose graphs. Finally, we verify the effectiveness of the object-related human pose in recognizing object-related actions. We will show improvement using the proposed components in our framework compared with the state-of-the-art method, ST-GCN [39], for skeleton-based action recognition.

**Frame selection strategies.** We compare the strategies of selecting frames on the IRD and ICVL-4 datasets: the proposed informative frame sampling and the existing uniform sampling. Informative samples are selected from the informative frame selection strategy, and uniform samples are obtained from the uniform sampling strategy used in ST-GCN [39]. The uniform sampling selects  $T$  frames at the same interval from a video. The comparisons between uniform and informative samples are performed using the human pose (HP) stream and the object-related human pose with hands connection (OHP-hands) stream. Our results, as observed in Table 1, show that the informative frame selection is a better strategy than uniform sampling. In particular, the informative frame selection in HP stream achieves a 5.5% (on IRD) and a 8% (on ICVL-4) improvement compared to ST-GCN using uniform samples. We argue that the improved performance comes from building more meaningful graphs by ignoring redundant information and focusing on accurate poses. Our methods use the informative frame selection strategy for the following experiments.

**Graph construction strategies.** Approaches to connecting the object to the human body are explored empirically. ‘OHP-body’ indicates connections between all the human joints and the object; ‘OHP-limbs’ indicates connections between the human joints belonging to arms or legs and the object; ‘OHP-hands’ denotes connections between the human joints of both hands and the object. ‘OHP-body’

shows lower performance than ST-GCN because the graph structure becomes complex and introduces redundant information. It means that the more abstracted form the object-related pose graph takes, the better its performance. As shown in Table 1, ‘OHP-hands’ shows the best performance among all the strategies. We believe this result is reasonably good because object-related human actions usually occur in human hands. Consequently, we use the connections between human hands and an object in other experiments.

**HP stream and OHP stream.** We also compare the human pose stream with the object-related human pose stream in the Table 1. The method using the object-related poses with the uniform sampling strategy (OHP-hands stream + uniform samples) does not improve performance compared to the human poses (HP stream) with the same strategy (ST-GCN; HP stream + uniform samples). While, The usage of object-related poses with the informative frame selection (OHP-hands stream + informative samples) achieves a 2.2% (on IRD) and a 1.2% (on ICVL-4) improvements compared to ST-GCN. The improvement is minimal, however, in the case of ‘HP stream + informative samples’. This difference in the performances is the result of the fact that the frame selection strategy is more informative to the human pose graph than the object-related one. By combining the HP and OHP streams, the two-stream OHA-GCN achieves the best performance of a 80.11% and a 91.86% on the IRD and ICVL-4 datasets, respectively, which is a significant improvement over the state-of-the-art, ST-GCN for skeleton-based action recognition.

### 4.2.2 Comparison with CNN methods using images

Our method is compared with other CNN-based methods, evaluated on the ICVL-4 datasets. Baseline CNNs are models of RGB modality, using only RGB images as input for the network. In fact, this is not a fair comparison because our approach is based on pose modality while CNN methods are based on RGB modality. Still, we want to show that even without the appearance information that is high dimensionality, our method can achieve a comparable perfor-

Table 2. Accuracy and speed comparisons with the CNN-based methods using RGB image inputs on the ICVL-4 dataset.

| Method               | Acc.(%)      | Speed (ms)    |
|----------------------|--------------|---------------|
| CNN (LeNet [21])     | 87.21        | 3.66          |
| CNN (VGG16 [31])     | 95.35        | 24.56         |
| CNN (ResNet50 [11])  | <b>96.51</b> | 17.26         |
| ST-GCN [39]          | 80.23        | 0.1321        |
| OHA-GCN (HP stream)  | 88.37        | <b>0.1306</b> |
| OHA-GCN (OHP stream) | 81.40        | 0.1319        |
| OHA-GCN (Two Stream) | 91.86        | 0.2447        |

HP: Human Pose, OHP: Object-related Human Pose

mance. We establish baseline models whose architectures are LeNet [21], VGG16 [31], and ResNet50 [11], respectively. As shown in Table 2, the accuracy of our method is lower than some of CNN-based methods, as expected. This result is explained by the difference in the modalities. RGB modality has more information for actions while pose-based representation is a compressed representation of the human body. Still, we argue that our pose-based approach achieves not only fairly high performance but also faster speed than RGB-based models. To examine this argument, we perform speed comparisons with the RGB-based methods. The test is performed on GTX TITAN X GPU, and other details of our implemented environments are described in Section 4.3. Using the ICVL-4 datasets, we measure the average runtime that the networks take to process one frame. Our method can operate at around 3571 frames per second (FPS), much faster speed than the CNN-based methods. The LeNet works relatively quickly with light computation, but its accuracy is inferior to that of our method. The ResNet achieves better results than our methods, but cannot operate quickly due to heavy computation. Given these speed/accuracy trade-offs, our method has more advantages than other methods because it achieves the comparable results at high speed even without the RGB modality.

### 4.3. Applications

Not only do our methods show promising results, but they also open up the possibility of many applications, building a real-world surveillance system. As a number of people appear throughout the original video in both datasets, there are many actors and actions to be recognized. To prevent multiple people from obstructing an action of one actor in a video, we apply a tracking algorithm to extract the regions where a single person exists. For the tracking, we adopt the online tracking-by-detection method based on the Hungarian matching algorithm [42] using pose information. If more than two people overlap in one patch obtained from tracking, we use the person whose joints have the highest sum of confidence scores as an actor for action recognition.

To examine the real-time performance of our proposed method in the surveillance systems, we analyze the runtime of the overall pipeline. Our method has been implemented with Pytorch deep learning framework, and the runtime is measured on GTX TITAN X GPU and Intel Core i7-6700K 4.0Ghz. For applications in real-world surveillance, we also use tracking algorithms whose runtime is 12.87 ms using only CPU. In our evaluation on the IRD dataset, foreground detection [3] and object detection operate at up to 45.78 ms and 14.93 ms using only CPU, respectively. The frame size in the original videos is 1920×1080, but we resize it to 656×368 which is the size recommended by *OpenPose* [3]. The pose estimation algorithm takes 74.62 ms using one GPU, 38.02 ms using two GPUs, and 20.92 ms using three GPUs. The overall pipeline, including the runtime of OHA-GCN, takes around 94.78 ms (10.6 FPS) on average, which can be applied to real-time monitoring. The experimental results and runtime analysis imply that our approach can be successfully implemented in real-world scenarios, achieving real-time performance.

## 5. Conclusion

Our main contributions are summarized as follows. First, we propose a new framework, object-related human action recognition (OHA-GCN), which uses the graphs of human poses and the object-related human poses to understand object-related human actions. Second, to overcome the difficulties of skeleton-based action recognition in real-world scenarios, we explore good strategies, including informative frame selection and construction of the object-related human poses. Third, we introduce a new dataset of illegal rubbish dumping (IRD), which compiles realistic data in unconstrained environments. Fourth, our framework can run in real time while achieving significantly better performance than the state-of-the-art algorithm for skeleton-based action recognition. It is noted that the proposed strategies offer meaningful insights into how to incorporate contextual information into human-object interaction recognition in the future.

**Acknowledgements** This work was partly supported by Next-Generation ICD Program through NRF funded by Ministry of S&ICT [2017M3C4A7077582], the ICT R&D program of MSIT/IITP. (No.B0101-15-0266, Development of High Performance Visual BigData Discovery Platform for Large-Scale Realtime Data Analysis and No.B0101-15-0552, Predictive Visual Intelligence Technology).

## References

- [1] O. Barnich and M. Van Droogenbroeck. ViBe: A Universal Background Subtraction Algorithm for Video Sequences. *IEEE Transactions on Image Processing*, 20(6):1709–1724, June 2011.



- [2] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. In *ICLR*, 2014.
- [3] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.
- [4] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.
- [5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.
- [6] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, 2016.
- [7] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *NIPS*, 2015.
- [8] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016.
- [9] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015.
- [10] G. Gkioxari, R. Girshick, P. Dollar, and K. He. Detecting and Recognizing Human-Object Interactions. In *CVPR*, 2018.
- [11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [12] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation Networks for Object Detection. In *CVPR*, 2018.
- [13] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *CVPR*, 2015.
- [14] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.
- [15] C.-B. Jin, T. D. Do, M. Liu, and H. Kim. Real-Time Action Detection in Video Surveillance using a Sub-Action Descriptor with Multi-Convolutional Neural Networks. *Journal of Institute of Control, Robotics and Systems*, 24(3):298–308, Mar. 2018.
- [16] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid. A new representation of skeleton sequences for 3d action recognition. In *CVPR*, 2017.
- [17] S. W. Kim, K. Yun, K. M. Yi, S. J. Kim, and J. Y. Choi. Detection of moving objects with a moving camera using non-panoramic background model. *Machine Vision and Applications*, 24(5):1015–1028, Oct. 2012.
- [18] T. S. Kim and A. Reiter. Interpretable 3d human action analysis with temporal convolutional networks. In *CVPR Workshops*, 2017.
- [19] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- [20] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.
- [21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [22] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. Gated graph sequence neural networks. In *ICLR*, 2016.
- [23] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *ECCV*, 2016.
- [24] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104(2-3):90–126, 2006.
- [25] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.
- [26] M. Niepert, M. Ahmed, and K. Kutzkov. Learning convolutional neural networks for graphs. In *ICML*, 2016.
- [27] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *CVPR*, 2017.
- [28] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *CVPR*, 2016.
- [29] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, pages 1297–1304. Ieee, 2011.
- [30] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.
- [31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [32] Y. Tang, Y. Tian, J. Lu, P. Li, and J. Zhou. Deep progressive reinforcement learning for skeleton-based action recognition. In *CVPR*, 2018.
- [33] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.
- [34] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*, 2014.
- [35] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.
- [36] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.
- [37] P. Wang, C. Yuan, W. Hu, B. Li, and Y. Zhang. Graph based skeleton motion representation and similarity measurement for action recognition. In *ECCV*, 2016.
- [38] L. Xia, C.-C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *CVPR Workshops, 2012 IEEE computer society conference on*, pages 20–27, 2012.
- [39] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.

- [40] A. Yao, J. Gall, and L. Van Gool. Coupled action recognition and pose estimation from multiple views. *International journal of computer vision*, 100(1):16–37, 2012.
- [41] K. M. Yi, K. Yun, S. W. Kim, H. J. Chang, H. Jeong, and J. Y. Choi. Detection of Moving Objects with Non-stationary Cameras in 5.8ms: Bringing Motion Detection to Your Mobile Device. In *CVPR Workshops*, 2013.
- [42] H. Yoo, K. Kim, M. Byeon, Y. Jeon, and J. Y. Choi. On-line Scheme for Multiple Camera Multiple Target Tracking Based on Multiple Hypothesis Tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(3):454–469, Mar. 2017.
- [43] K. Yun, J. Lim, and J. Y. Choi. Scene conditional background update for moving object detection in a moving camera. *Pattern Recognition Letters*, 88:57–63, 2017.
- [44] S. Zhang, X. Liu, and J. Xiao. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *WACV*, 2017.