**PaStaNet: Toward Human Activity Knowledge Engine**

Yong-Lu Li, Liang Xu, Xinpeng Liu, Xijie Huang, Yue Xu, Shiyi Wang, Hao-Shu Fang, Ze Ma, Mingyang Chen, Cewu Lu*
Shanghai Jiao Tong University

{yonglu_li, liangxu, otaku_huang, silicxuyue, shiywang}@sjtu.edu.cn,
{xinpengliu0907, fhaoshu}@gmail.com, {mazel234556, cmy_123, lucewu}@sjtu.edu.cn

Abstract

Existing image-based activity understanding methods mainly adopt direct mapping, i.e., from image to activity concepts, which may encounter performance bottleneck since the huge gap. In light of this, we propose a new path: infer human part states first and then reason out the activities based on part-level semantics. Human Body Part States (PaSta) are fine-grained action semantic tokens, e.g. \{hand, hold, something\}, which can compose the activities and help us step toward human activity knowledge engine. To fully utilize the power of PaSta, we build a large-scale knowledge base PaStaNet, which contains 7M+ PaSta annotations. And two corresponding models are proposed: first, we design a model named Activity2Vec to extract PaSta features, which aim to be general representations for various activities. Second, we use a PaSta-based Reasoning method to infer activities. Promoted by PaStaNet, our method achieves significant improvements, e.g. 6.4 and 13.9 mAP on full and one-shot sets of HICO in supervised learning, and 3.2 and 4.2 mAP on V-COCO and images-based AVA in transfer learning. Code and data are available at [http://hake-mvig.cn/](http://hake-mvig.cn/).

1. Introduction

Understanding activity from images is crucial for building an intelligent system. Facilitated by deep learning, great advancements have been made in this field. Recent works [7, 65, 62, 41] mainly address this high-level cognition task in one-stage, i.e., from pixels to activity concept directly based on instance-level semantics (Fig. 1(a)). This strategy faces performance bottleneck on large-scale benchmarks [3, 24]. Understanding activities is difficult for reasons, e.g., long-tail data distribution, complex visual patterns, etc. Moreover, action understanding expects a knowledge engine that can generally support activity related tasks.

* Cewu Lu is the corresponding author, member of Qing Yuan Research Institute and MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China.

![Figure 1. Instance-level and hierarchical methods.](http://hake-mvig.cn/)

Figure 1. Instance-level and hierarchical methods. Besides the instance-level path, we perform body part states recognition in part-level with the PaSta annotations. With the help of PaSta, we can significantly boost the performance of activity understanding.

Thus, for data from another domain and unseen activities, much smaller effort is required for knowledge transfer and adaptation. Additionally, for most cases, we find that only a few key human parts are relevant to the existing actions, the other parts usually carry very few useful clues.

Consider the example in Fig. 1, we argue that perception in human part-level semantics is a promising path but previously ignored. Our core idea is that human instance actions are composed of fine-grained atomic body part states. This lies in strong relationships with reductionism [10]. Moreover, the part-level path can help us to pick up discriminative parts and disregard irrelevant ones. Therefore, encoding knowledge from human parts is a crucial step toward human activity knowledge engine. The generic object part states [39] reveal that the semantic state of an object part is limited. For example, after exhaustively checking on 7M manually labeled body part state samples, we find that there are only about 12 states for “head” in daily life activities, such as “listen to”, “eat”, “talk to”, “inspect”, etc. Therefore, in this paper, we exhaustively collect and annotate the possible semantic meanings of human parts in activities to build a large-scale human part knowledge base PaStaNet (PaSta is the abbreviation of Body Part State). Now PaStaNet includes 118 K+ images, 285 K+ persons,
**Related Works**

**Activity Understanding.** Benefited by deep learning and large-scale datasets, image-based [4, 25, 33, 1] or video-based [24, 46, 47, 2, 54, 51, 30] activity understanding has achieved huge improvements recently. Human activities have a hierarchical structure and include diverse verbs, so it is hard to define an explicit organization for their categories. Existing datasets [24, 4, 12, 25] often have a large difference in definition, thus transferring knowledge from one dataset to another is ineffective. Meanwhile, plenty of works have been proposed to address the activity understanding [7, 20, 11, 52, 16, 31, 57, 55]. There are holistic body-level approaches [56, 7], body part-based methods [20], and skeleton-based methods [58, 11], etc. But compared with other tasks such as object detection [50] or pose estimation [15], its performance is still limited.

**Human-Object Interaction.** Human-Object Interaction (HOI) [4, 3] occupies the most of daily human activities. In terms of tasks, some works focus on image-based HOI recognition [4]. Furthermore, instance-based HOI detection [3, 25] needs to detect accurate positions of the humans and objects and classify interaction simultaneously. In terms of the information utilization, some works utilized holistic human body and pose [56, 65, 62, 41, 5], and global context is also proved to be effective [29, 64, 63, 7]. According to the learning paradigm, earlier works were often based on hand-crafted features [7, 29]. Benefited from large scale HOI datasets, recent approaches [20, 13, 22, 19, 42, 49, 17, 34] started to use deep neural networks to extract features and achieved great improvements.

**Body Part based Methods.** Besides the instance pattern, some approaches studied to utilize part pattern [64, 13, 20, 11, 41, 66]. Gkioxari et al. [20] detects both the instance and parts and input them all into a classifier. Fang et al. [13] defines part pairs and encodes pair features to improve HOI recognition. Yao et al. [64] builds a graphical model and embed parts appearance as nodes, and use them with object feature and pose to predict the HOIs. Previous work mainly utilized the part appearance and location, but few studies tried to divide the instance actions into discrete part-level semantic tokens, and refer them as the basic components of activity concepts. In comparison, we aim at building human part semantics as reusable and transferable knowledge.

**Part States.** Part state is proposed in [39]. By tokenizing the semantic space as a discrete set of part states, [39] constructs a sort of basic descriptors based on segmentation [26, 14, 61]. To exploit this cue, we divide the human body into natural parts and utilize their states as discretized part semantics to represent activities. In this paper, we focus on the part states of humans instead of daily objects.

**3. Constructing PaStaNet**

In this section, we introduce the construction of PaStaNet. PaStaNet seeks to explore the common knowledge of human PaSta as atomic elements to infer activities.

**PaSta Definition.** We decompose human body into ten parts, namely head, two upper arms, two hands, hip, two thighs, two feet. Part states (PaSta) will be assigned to these parts. Each PaSta represents a description of the target part. For example, the PaSta of “hand” can be “hold something”
or “push something”, the PaSta of “head” can be “watch something”, “eat something”. After exhaustively reviewing collected 200K+ images, we found the descriptions of any human parts can be concluded into limited categories. That is, the PaSta category number of each part is limited. Especially, a person may have more than one action simultaneously, thus each part can have multiple PaSta, too.

**Data Collection.** For generality, we collect human-centric activity images by crowdsourcing (30K images paired with rough activity label) as well as existing well-designed datasets [4, 3, 25, 33, 67, 36] (185K images), which are structured around a rich semantic ontology, diversity, and variability of activities. All their annotated persons and objects are extracted for our construction. Finally, we collect more than 200K images of diverse activity categories.

**Activity Labeling.** Activity categories of PaStaNet are chosen according to the most common human daily activities, interactions with object and person. Referred to the hierarchical activity structure [12], common activities in existing datasets [4, 25, 67, 33, 24, 12, 1, 36] and crowdsourcing labels, we select 156 activities including human-object interactions and body motions from 118K images. According to them, we first clean and reorganize the annotated human and objects from existing datasets and crowdsourcing. Then, we annotate the active persons and the interacted objects in the rest of the images. Thus, PaStaNet includes all active human and object bounding boxes of 156 activities.

**Body Part Box.** To locate the human parts, we use pose estimation [15] to obtain the joints of all annotated persons. Then we generate ten body part boxes following [13]. Estimation errors are addressed manually to ensure high-quality annotation. Each part box is centered with a joint, and the box size is pre-defined by scaling the distance between the joints of the neck and pelvis. A joint with confidence higher than 0.7 will be seen as visible. When not all joints can be detected, we use body knowledge-based rules. That is, if the neck or pelvis is invisible, we configure the part boxes according to other visible joint groups (head, main body, arms, legs), e.g., if only the upper body is visible, we set the size of the hand box to twice the pupil distance.

**PaSta Annotation.** We carry out the annotation by crowdsourcing and receive 224,159 annotation uploads. The process is as follows: 1) First, we choose the PaSta categories considering the generalization. Based on the verbs of 156 activities, we choose 200 verbs from WordNet [45] as the PaSta candidates, e.g., “hold”, “pick” for hands, “eat”, “talk to” for head, etc. If a part does not have any active states, we depict it as “no_action”. 2) Second, to find the most common PaSta that can work as the transferable activity knowledge, we invite 150 annotators from different backgrounds to annotate 10K images of 156 activities with PaSta candidates (Fig. 2). For example, given an activity “ride bicycle”, they may describe it as ⟨hip, sit_on, something⟩, ⟨hand, hold, something⟩, ⟨foot, step_on, something⟩, etc. 3) Based on their annotations, we use the Normalized Point-wise Mutual Information (NPMI) [6] to calculate the co-occurrence between activities and PaSta candidates. Finally, we choose 76 candidates with the highest NPMI values as the final PaSta. 4) Using the annotations of 10K images as seeds, we automatically generate the initial PaSta labels for all of the rest images. Thus the other 210 annotators only need to revise the annotations. 5) Considering that a person may have multiple actions, for each action, we annotate its corresponding ten PaSta respectively. Then we combine all sets of PaSta from all actions. Thus, a part can also have multiple states, e.g., in “eating while talking”, the head has PaSta ⟨head, eat, something⟩, ⟨head, talk_to, something⟩ and ⟨head, look_at, something⟩ simultaneously. 6) To ensure quality, each image will be annotated twice and checked by automatic procedures and supervisors. We cluster all labels and discard the outliers to obtain robust agreements.

**Activity Parsing Tree.** To illustrate the relationships between PaSta and activities, we use their statistical correlations to construct a graph (Fig. 2): activities are root nodes, PaSta are son nodes and edges are co-occurrence.

Finally, PaStaNet includes 118K+ images, 285K+ persons, 250K+ interacted objects, 724K+ instance activities and 7M+ PaSta. Referred to well-designed datasets [24, 12, 4] and WordNet [45], PaSta can cover most part situations with good generalization. To verify that PaSta have encoded common part-level activity knowledge and can adapt to various activities, we adopt two experiments:

**Coverage Experiment.** To verify that PaSta can cover most of the activities, we collect other 50K images out of PaStaNet. Those images contain diverse activities and
many of them are unseen in PaStaNet. Another 100 volunteers from different backgrounds are invited to find human parts that can not be well described by our PaSta set. We found that only 2.3% cases cannot find appropriate descriptions. This verifies that PaStaNet is general to activities.

**Recognition Experiment.** First, we find that PaSta can be well learned. A shallow model trained with a part of PaStaNet can easily achieve about 55 mAP on PaSta recognition. Meanwhile, a deeper model can only achieve about 40 mAP on activity recognition with the same data and metric (Sec. 5.2). Second, we argue that PaSta can be well transferred. To verify this, we conduct transfer learning experiments (Sec. 5.4), i.e. first trains a model to learn the knowledge from PaStaNet, then use it to infer the activities of unseen datasets, even unseen activities. Results show that PaSta can be well transferred and boost the performance (4.2 mAP on image-based AVA). Thus it can be considered as the general part-level activity knowledge.

### 4. Activity Representation by PaStaNet

In this section, we discuss the activity representation by PaStaNet.

**Conventional Paradigm** Given an image $I$, conventional methods mainly use a direct mapping (Fig. 1(a)):

$$
S_{\text{inst}} = F_{\text{inst}}(I, b_h, B_o)
$$

(1)

to infer the action score $S_{\text{inst}}$ with instance-level semantic representations $f_{\text{inst}}$. $b_h$ is the human box and $B_o = \{b_o^i\}_{i=1}^m$ are the $m$ interacted object boxes of this person.

**PaStaNet Paradigm.** We propose a novel paradigm to utilize general part knowledge: 1) PaSta recognition and feature extraction for a person and an interacted object $b_o$:

$$
f_{\text{PaSta}} = R_{\text{A2V}}(I, B_o, b_o),
$$

(2)

where $B_o = \{b_o^i\}_{i=1}^{10}$ are part boxes generated from the pose estimation [15] automatically following [15] (head, upper arms, hands, hip, thighs, feet). $R_{\text{A2V}}(\cdot)$ indicates the Activity2Vec, which extracts ten PaSta representations $f_{\text{PaSta}} = \{f_{\text{PaSta}}^i\}_{i=1}^{10}$. 2) PaSta-based Reasoning (PaSta-R), i.e., from PaSta to activity semantics:

$$
S_{\text{part}} = F_{\text{PaSta-R}}(f_{\text{PaSta}}, f_o),
$$

(3)

where $F_{\text{PaSta-R}}(\cdot)$ indicates the PaSta-R, $f_o$ is the object feature. $S_{\text{part}}$ is the action score of the part-level path. If the person does not interact with any objects, we use the ROI pooling feature of the whole image as $f_o$. For multiple object case, i.e., a person interacts with several objects, we process each human-object pair $(f_{\text{PaSta}}, f_o^i)$ respectively and generate its Activity2Vec embedding.

Following, we introduce the PaSta recognition in Sec. 4.1. Then, we discuss how to map human instance to semantic vector via Activity2Vec in Sec. 4.2. We believe it can be a general activity representation extractor. In Sec. 4.3, a hierarchical activity graph is proposed to largely advance activity related tasks by leveraging PaStaNet.

#### 4.1. Part State Recognition

With the object and body part boxes $b_o, B_p$, we operate the PaSta recognition as shown in Fig. 3. In detail, a COCO [35] pre-trained Faster R-CNN [50] is used as the feature extractor. For each part, we concatenate the part feature $f_p^i$ from $b_p^i$ and object features $f_o$ from $b_o$ as inputs. For body only motion, we input the whole image feature $f_c$ as $f_o$. All features will be first input to a Part Relevance Predictor. Part relevance represents how important a body part is to the action. For example, feet usually have weak correlations with “drink with cup”. And in “eat apple”, only hands and head are essential. These relevance/attention labels can be converted from PaSta labels directly, i.e. the attention label will be one, unless its PaSta label is “no_action”, which means this part contributes nothing to the action inference. With the part attention labels as supervision, we use part relevance predictor consisting of FC layers and Sigmoids to infer the attentions $\{a_i\}_{i=1}^{10}$ of each part. Formally, for a person and an interacted object:

$$
a_i = P_{pa}(f_p^i, f_o),
$$

(4)

where $P_{pa}(\cdot)$ is the part attention predictor. We compute cross-entropy loss $L_{\text{att}}^{(i)}$ for each part and multiply $f_p^i$ with its scalar attention, i.e. $f_p^{(i)*} = f_p^i \times a_i$.

Second, we operate the PaSta recognition. For each part, we concatenate the re-weighted $f_p^{(i)*}$ with $f_o$, and input them into a max pooling layer and two subsequent 512 sized FC layers, thus obtain the PaSta score $S_{\text{PaSta}}^{(i)}$ for the $i$-th part. Because a part can have multiple states, e.g. head performs “eat” and “watch” simultaneously. Hence we use multiple Sigmoids to do this multi-label classification. With PaSta labels, we construct cross-entropy loss $L_{\text{PaSta}}^{(i)}$. The total loss of PaSta recognition is:

$$
L_{\text{PaSta}} = \sum_{i=1}^{10} (L_{\text{PaSta}}^{(i)} + L_{\text{att}}^{(i)}).
$$

(5)

#### 4.2. Activity2Vec

In Sec. 3, we define the PaSta according to the most common activities. That is, choosing the part-level verbs which are most often used to compose and describe the activities by a large number of annotators. Therefore PaSta can be seen as the fundamental components of instance activities. Meanwhile, PaSta recognition can be well learned. Thus, we can operate PaSta recognition on PaStaNet to learn the powerful PaSta representations, which have good transferability. They can be used to reason out the instance actions
in both supervised and transfer learning. Under such circumstance, PaStaNet works like the ImageNet [8]. And PaStaNet pre-trained Activity2Vec functions as a knowledge engine and transfers the knowledge to other tasks.

**Visual PaSta feature.** First, we extract visual PaSta representations from PaSta recognition. Specifically, we extract the feature from the last FC layer in PaSta classifier as the visual PaSta representation \( f_{\text{PaSta}}^{V(i)} \in \mathbb{R}^{512} \).

**Language PaSta feature.** Our goal is to bridge the gap between PaSta and activity semantics. Language priors are useful in visual concept understanding [38, 59]. Thus the combination of visual and language knowledge is a good choice for establishing this mapping. To further enhance the representation ability, we utilize the uncased BERT-Base pre-trained model [9] as the language representation extractor. Bert [9] is a language understanding model that considers the context of words and uses a deep bidirectional transformer to extract contextual representations. It is trained with large-scale corpus databases such as Wikipedia, hence the generated embedding contains helpful implicit semantic knowledge about the activity and PaSta. For example, the description of the entry “basketball” in Wikipedia: “drag one’s foot without dribbling the ball, to carry it, or to hold the ball with both hands...placing his hand on the bottom of the ball...known as carrying the ball”.

In specific, for the \( i \)-th body part with \( n \) PaStas, we divide each PaSta into tokens \( \{t_p^{(i,k)}, t_v^{(i,k)}, t_o^{(i,k)}\}_{k=1}^n \), e.g., \( \langle \text{part}, \text{verb}, \text{object} \rangle \). The \( \langle \text{object} \rangle \) comes from object detection. Each PaSta will be converted to a \( f_{\text{Bert}}^{(i,k)} \in \mathbb{R}^{2048} \) (concatenating three 768 sized vectors of part, verb, object), i.e., \( f_{\text{Bert}}^{(i,k)} = \{t_p^{(i,k)}, t_v^{(i,k)}, t_o^{(i,k)}\} \). \( \{f_{\text{Bert}}^{(i,k)}\}_{k=1}^n \) will be concatenated as the \( f_{\text{PaSta}}^{V(i)} \in \mathbb{R}^{2048 \times n} \) for the \( i \)-th part. Second, we multiply \( f_{\text{Bert}}^{(i,k)} \) with predicted PaSta probabilities \( P_{\text{PaSta}}^{(i)} \), i.e., \( f_{\text{PaSta}}^{L(i)} = f_{\text{Bert}}^{(i,k)} \times P_{\text{PaSta}}^{(i)} \) where \( P_{\text{PaSta}}^{(i)} = \text{Sigmoid}(S_{\text{PaSta}}^{(i)}) \in \mathbb{R}^n \). \( S_{\text{PaSta}}^{(i)} \) denotes the PaSta score of the \( i \)-th part, \( P_{\text{PaSta}}^{(i)} = \{P_{\text{PaSta}}^{(i)}\}_{i=1}^{10} \). This means a more possible PaSta will get larger attention. \( f_{\text{PaSta}}^{L(i)} \in \mathbb{R}^{2048 \times n} \) is the final language PaSta feature of the \( i \)-th part. We use the pre-converted and frozen \( f_{\text{Bert}}^{(i,k)} \) in the whole process. Additionally, we also try to rewrite each PaSta into a sentence and convert it into a fixed-size vector as \( f_{\text{Bert}}^{(i,k)} \) and the performance is slightly better (Sec. 5.5).

**PaSta Representation.** At last, we pool and resize the \( f_{\text{PaSta}}^{L(i)} \), and concatenate it with its corresponding visual PaSta feature \( f_{\text{PaSta}}^{V(i)} \). Then we obtain the PaSta representation \( f_{\text{PaSta}}^{(i)} \in \mathbb{R}^m \) for each body part (e.g., \( m = 4096 \)). This process is indicated as Activity2Vec (Fig. 3). The output \( f_{\text{PaSta}}^{(i)} \) is the part-level activity representation and can be used for various downstream tasks, e.g., activity detection, captioning, etc. From the experiments, we can find that Activity2Vec has a powerful representational capacity and can significantly improve the performance of activity related tasks. It works like a knowledge transformer with the fundamental PaSta to compose various activities.

### 4.3. PaSta-based Activity Reasoning

With part-level \( f_{\text{PaSta}}^{(i)} \), we construct a Hierarchical Activity Graph (HAG) to model the activities. Then we can extract the graph state to reason out the activities.

**Hierarchical Activity Graph.** Hierarchical activity graph \( G = (\mathcal{V}, \mathcal{E}) \) is depicted in Fig. 4. For human-object interactions, \( \mathcal{V} = \{\mathcal{V}_p, \mathcal{V}_o\} \). For body only motions, \( \mathcal{V} = \mathcal{V}_p \). In instance level, a person is a node with instance representation from previous instance-level methods [17, 34, 24] as a node feature. Object node \( v_o \in \mathcal{V}_o \) and has \( f_o \) as node feature. In part level, each body part can be seen as a node \( v_p \in \mathcal{V}_p \) with PaSta representation \( f_{\text{PaSta}}^{(i)} \) as node feature. Edge between body parts and object is \( e_{po} = (v_p^o, v_o) \in \mathcal{V}_p \times \mathcal{V}_o \), and edge within parts is \( e_{pp} = (v_p^p, v_p^p) \in \mathcal{V}_p \times \mathcal{V}_p \).

Our goal is to parse HAG and reason out the graph state,
i.e. activities. In part-level, we use PaSta-based Activity Reasoning (PaSta-R) to infer the activities. That is, with the PaSta representation from Activity2Vec, we use $S_{part} = F_{PaSta-R}(f_{PaSta}, f_o)$ (Eq. 3) to infer the activity scores $S_{part}$. For body motion only activities e.g. “dance”, Eq. 3 is $S_{part} = F_{PaSta-R}(f_{PaSta}, f_e)$, $f_e$ is the feature of image. We adopt different implementations of $F_{PaSta-R}$. 

**Linear Combination.** The simplest implementation is to directly combine the part node features linearly. We concatenate the output of Activity2Vec $f_{PaSta}$ with $f_o$ and input them to a FC layer with Sigmoids.

**MLP.** We also can operate nonlinear transformation on Activity2Vec output. We use two 1024 sized FC layers and an action category sized FC with Sigmoids.

**Graph Convolution Network.** With part-level graph, we use Graph Convolution Network (GCN) [32] to extract the global graph feature and use an MLP subsequently.

**Sequential Model.** When watching an image in this way: watch body part and object patches with language description one by one, human can easily guess the actions. Inspired by this, we adopt an LSTM [28] to take the part node features $f_{PaSta}$ gradually, and use the output of the last time step to classify actions. We adopt two input orders: random and fixed (from head to foot), and fixed order is better.

**Tree-Structured Passing.** Human body has a natural hierarchy. Thus we use a tree-structured graph passing. Specifically, we first combine the hand and upper arm nodes into an “arm” node, its feature is obtained by concatenating the features of three son nodes and passed a 512 sized FC layer. Similarly, we combine the foot and thigh nodes to an “leg” node. Head, arms, legs and feet nodes together form the second level. The third level contains the “upper body” (head, arms) and “lower-body” (hip, legs). Finally, the body node is generated. We input it and the object node into an MLP.

The instance-level graph inference can be operated by instance-based methods [13, 17, 34, 24] using Eq. 1: $S_{inst} = F_{inst}(I, b_h, b_o)$. To get the final result upon the whole graph, we can use either early or late fusion. In early fusion, we concatenate $f_{inst}$ with $f_{PaSta-R}$, and input them to PaSta-R. In late fusion, we fuse the predictions of two levels, i.e. $S = S_{inst} + S_{part}$. In our test, late fusion outperforms early fusion in most cases. If not specified, we use late fusion in Sec. 5. We use $L_{cls}^{inst}$ and $L_{cls}^{PaSta}$ to indicate the cross-entropy losses of two levels. The total loss is:

$$L_{total} = L_{PaSta} + L_{cls}^{PaSta} + L_{cls}^{inst}.$$  

5. Experiments

5.1. An analogy: MNIST-Action

We design a simplified experiment to give an intuition (Fig. 5). We randomly sample MNIST digits from 0 to 9 ($28 \times 28 \times 1$) and generate $128 \times 128 \times 1$ images consists of 3 to 5 digits. Each image is given a label to indicate the sum of the two largest numbers within it (0 to 18). We assume that “PaSta-Activity” resembles the “Digits-Sum”. Body parts can be seen as digits, thus human is the union box of all digits. To imitate the complex body movements, digits are randomly distributed, and Gaussian noise is added to the images. For comparison, we adopt two simple networks. For instance-level model, we input the ROI pooling feature of the digit union box into an MLP. For hierarchical model, we operate single-digit recognition, then concatenate the union box and digit features and input them to an MLP (early fusion), or use late fusion to combine scores of two levels. Early fusion achieves 43.7 accuracy and shows significant superiority over instance-level method (10.0). And late fusion achieves a preferable accuracy of 44.2. Moreover, the part-level method only without fusion also obtains an accuracy of 41.4. This supports our assumption about the effectiveness of part-level representation.

5.2. Image-based Activity Recognition

Usually, Human-Object Interactions (HOIs) often take up most of the activities, e.g., more than 70% activities in large-scale datasets [24, 12, 2] are HOIs. To evaluate PaStaNet, we perform image-based HOI recognition on HICO [4]. HICO has 38,116 and 9,658 images in train and test sets and 600 HOIs composed of 117 verbs and 80 COCO objects [35]. Each image has an image-level label which is the aggregation over all HOIs in an image and does not contain any instance boxes.

**Modes.** We first pre-train Activity2Vec with PaSta labels, then fine-tune Activity2Vec and PaSta-R together on HICO train set. In pre-training and finetuning, we exclude the HICO testing data in PaStaNet to avoid data pollution. We
Table 1. Results on HICO. “Pairwise [13]+PaStaNet” means the late fusion of [13] and our part-level result. \#i indicates the number of training images is less than or equal to \(i\). The HOI categories number of Few@1, 5, 10 are 49, 125 and 163. “PaStaNet-x” means different PaSta-R.

Table 2. Results on HICO-DET. mAP on PaStaNet recognition in PaStaNet* mode: 46.3 (head), 66.8 (arms), 32.0 (hands), 68.6 (hip), 56.2 (thighs), 65.8 (feet). This verifies that PaStaNet can be better learned than activities, thus they can be learned ahead as the basis for reasoning. In GT-PaStaNet* mode, hierarchical paradigm achieves 65.6 mAP. This is a powerful proof of the effectiveness of PaStaNet knowledge. Thus what remains to do is to improve the PaStaNet recognition and further promote the activity task performance. Moreover, in PaStaNet mode, we achieve relative 16% improvement. On few-shot sets, our best result significantly improves 13.9 mAP, which strongly proves the reusability and transferability of PaStaNet.

5.3. Instance-based Activity Detection

We further conduct instance-based activity detection on HICO-DET [3], which needs to locate human and object and classify the actions simultaneously. HICO-DET [3] is a benchmark built on HICO [4] and add human and object bounding boxes. We choose several state-of-the-arts [17, 19, 49, 34] to compare and cooperate.

Settings. We use instance-level PaStaNet labels, i.e. each annotated person with the corresponding PaStaNet labels, to train Activity2Vec, and fine-tune Activity2Vec and PaStaNet together on HICO-DET. All testing data are excluded from pre-training and fine-tuning. We follow the mAP metric of [3], i.e., true positive contains accurate human and object boxes (\(IoU > 0.5\) with regard to ground truth) and accurate action prediction. The metric for PaStaNet detection is similar, i.e., estimated part box and PaStaNet action prediction all have to be accurate. The mAP of each part and the mean mAP are calculated. For a fair comparison, we use the object detection from [17, 34] and ResNet-50[27] as backbone. We use SGD with momentum (0.9) and cosine decay restart [37] (the first decay step is 5000). The pre-training costs 80K iterations and fine-tuning costs 20K iterations. Image-level PaStaNet and HOI predictions are all generated via Multiple Instance Learning (MIL) [43] of 3 persons and 4 objects. We choose previous methods [42, 13] as the instance-level path in the hierarchical model, and uses late fusion. Particularly, [13] uses part-pair appearance and location but not part-level semantics, thus we still consider it as a baseline to get a more abundant comparison.

Results. Results are reported in Tab. 1. PaStaNet* mode methods all outperform the instance-level method. The part-level method solely achieves 44.5 mAP and shows good complementarity to the instance-level. Their fusion can boost the performance to 45.9 mAP (6 mAP improvement). And the gap between [13] and [42] is largely narrowed from 3.8 to 0.9 mAP. Activity2Vec achieves 55.9 mAP on PaStaNet recognition in PaStaNet* mode: 46.3 (head), 66.8 (arms), 32.0 (hands), 68.6 (hip), 56.2 (thighs), 65.8 (feet). This verifies that PaStaNet can be better learned than activities, thus they can be learned ahead as the basis for reasoning. In GT-PaStaNet* mode, hierarchical paradigm achieves 65.6 mAP. This is a powerful proof of the effectiveness of PaStaNet knowledge. Thus what remains to do is to improve the PaStaNet recognition and further promote the activity task performance. Moreover, in PaStaNet mode, we achieve relative 16% improvement. On few-shot sets, our best result significantly improves 13.9 mAP, which strongly proves the reusability and transferability of PaStaNet.
Table 3. Transfer learning results on V-COCO [25].

<table>
<thead>
<tr>
<th>Method</th>
<th>AP_{bbox} (Uncropped)</th>
<th>AP_{bbox} (Cropped)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gupta et al [25]</td>
<td>31.8</td>
<td></td>
</tr>
<tr>
<td>InteracNet [19]</td>
<td>40.0</td>
<td></td>
</tr>
<tr>
<td>GPINN [49]</td>
<td>44.0</td>
<td></td>
</tr>
<tr>
<td>iCAN [17]</td>
<td>45.3</td>
<td>52.4</td>
</tr>
<tr>
<td>TIN [34]</td>
<td>47.8</td>
<td>54.2</td>
</tr>
<tr>
<td>iCAN [17] + PaStaNet-Linear</td>
<td>49.2</td>
<td>55.6</td>
</tr>
<tr>
<td>TIN [34] + PaStaNet-Linear</td>
<td>51.0</td>
<td>57.5</td>
</tr>
</tbody>
</table>

Table 4. Transfer learning results on image-based A V A [24].

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>A V A-TF [23]</td>
<td>11.4</td>
</tr>
<tr>
<td>LFB-Res-50-baseline [60]</td>
<td>22.2</td>
</tr>
<tr>
<td>LFB-Res-101-baseline [60]</td>
<td>23.3</td>
</tr>
<tr>
<td>LFB-Res-50-baseline [60] + GloVeNet-Linear</td>
<td>23.4</td>
</tr>
<tr>
<td>LFB-Res-101-baseline [60] + PaStaNet-Linear</td>
<td>24.3</td>
</tr>
</tbody>
</table>

(arms), 17.5 (hands), 41.8 (hip), 22.2 (thighs), 29.9 (feet). This again verifies that PaSta can be well learned. And GT-PaStaNet* (upper bound) and PaStaNet* (more PaSta labels) modes both greatly boosts the performance. On Rare sets, our method obtains 7.7 mAP improvement.

5.4. Transfer Learning with Activity2Vec

To verify the transferability of PaStaNet, we design transfer learning experiments on large-scale benchmarks: V-COCO [25], HICO-DET [3] and A V A [24]. We first use PaStaNet to pre-train Activity2Vec and PaSta-R with 156 activities and PaSta labels. Then we change the last FC in PaSta-R to fit the activity categories of the target benchmark. Finally, we freeze Activity2Vec and fine-tune PaSta-R on the train set of the target dataset. Here, PaStaNet works like the ImageNet [8] and Activity2Vec is used as a pre-trained knowledge engine to promote other tasks.

V-COCO. V-COCO contains 10,346 images and instance boxes. It has 29 action categories, COCO 80 objects [35]. For a fair comparison, we exclude the images of V-COCO and corresponding PaSta labels in PaStaNet, and use remaining data (109K images) for pre-training. We use SGD with 0.9 momenta and cosine decay restarts [37] (the first decay is 80K). The pre-training costs 300K iterations with the learning rate as 1e-3. The pre-training costs 1.1M iterations and fine-tuning costs 710K iterations. We adopt the metric from [24], i.e. mAP of the top 60 most common action classes, using IoU threshold of 0.5 between detected human box and the ground truth and accurate action prediction. For comparison, we adopt a image-based baseline: Faster R-CNN detector [50] with ResNet-101 [27] provided by the A V A website [23]. Recent works mainly use a spatial-temporal model such as I3D [2]. Although unfair, we still employ two video-based baselines [60] as instance-level models to cooperate with the part-level method via late fusion. Results are listed in Tab. 4. Both image and video based methods cooperated with PaStaNet achieve impressive improvements, even our model is trained without temporal information. Considering the huge domain gap (films) and unseen activities, this result strongly proves its great generalization ability.

HICO-DET. We exclude the images of HICO-DET and the corresponding PaSta labels, and use left data (71K images) for pre-training. The test setting in same with Sec. 5.3. The pre-training and fine-tuning cost 300K and 1.3M iterations. PaStaNet shows good transferability and achieve 3.25 mAP improvement on Default Full set (20.28 mAP).

5.5. Ablation Study

We design ablation studies on HICO-DET with PaStaNet. Table 4 shows good transferability and achieve 3.25 mAP improvement on Default Full set (20.28 mAP).

6. Conclusion

In this paper, to make a step toward human activity knowledge engine, we construct PaStaNet to provide novel body part-level activity representation (PaSta). Meanwhile, a knowledge transformer Activity2Vec and a part-based reasoning method PaSta-R are proposed. PaStaNet brings in interpretability and new possibility for activity understanding. It can effectively bridge the semantic gap between pixels and activities. With PaStaNet, we significantly boost the performance in supervised and transfer learning tasks, especially under few-shot circumstances. In the future, we plan to enrich our PaStaNet with spatio-temporal PaSta.
References


[34] Yong-Lu Li, Siyuan Zhou, Xijie Huang, Liang Xu, Ze Ma, Hao-Shu Fang, Yanfeng Wang, and Cewu Lu. Transferable interactiveness knowledge for human-object interaction detection. In CVPR, 2019. 2, 5, 6, 7, 8


[38] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Fei Fei Li. Visual relationship detection with language priors. In ECCV, 2016. 5


[40] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. JMLR, 2008. 11


[51] Dian Shao, Yu Xiong, Yue Zhao, Qingqiu Huang, Yu Qiao, and Dahua Lin. Find and focus: Retrieve and localize video events with natural language queries. In ECCV, 2018. 2


[58] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In CVPR, 2014. 2


[60] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In CVPR, 2019. 8


[64] Bangpeng Yao and Fei Fei Li. Modeling mutual context of object and human pose in human-object interaction activities. In CVPR, 2010. 2

[65] Bangpeng Yao and Fei Fei Li. Action recognition with exemplar based 2.5d graph matching. In ECCV, 2012. 1, 2


Appendices

![Diagram of PaSta features for foot and hand.](image)

Figure 6. Visualized PaSta representations via t-SNE [40]. Sub-figure (a) depicts the PaSta features of foot and sub-figure (b) depicts PaSta features of hand. Our model can extract meaningful and reasonable features for various PaSta.

<table>
<thead>
<tr>
<th>Image</th>
<th>Human-Instance</th>
<th>Object-Instance</th>
<th>Activity</th>
<th>PaSta</th>
<th>Existing-Datasets</th>
<th>Crowdsourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>118,995</td>
<td>285,921</td>
<td>250,621</td>
<td>724,855</td>
<td>7,248,550</td>
<td>95,027</td>
<td>23,968</td>
</tr>
</tbody>
</table>

Table 5. Characteristics of PastaNet. “Existing-Datasets” and “Crowdsourcing” indicate the image sources of PastaNet.

![Pie charts showing age, major, and education degree distributions.](image)

Figure 7. Annotator backgrounds.

A. Dataset Details

In this section, we give a more detailed introduction of our knowledge base PastaNet, covering characteristics and the annotator backgrounds.

A.1. Characteristics of PastaNet

Tab. 5 shows some characteristics of PastaNet. Now PastaNet contains about 118K+ images and the corresponding instance-level and human part-level annotations. To give an intuitive presentation of our PaSta, we use t-SNE [40] to visualize the part-level features of some typical PaSta samples in Fig. 6. We use the human body part patches with different colored borders to replace the embedding points in Fig. 6, e.g. the embeddings of “hand holds something” and “foot kicks something”.
A.2. Annotator Backgrounds

There are about 360 annotators have participated in the construction of PastaNet. They have various backgrounds, thus we can ensure annotation diversity and reduce the bias. The specific information is shown in Fig. 7.

B. Selected Activities and PaSta

We list all selected 156 activities and 76 PaSta in Tab. 6 and Tab. 7. PaStaNet contains both person-object/person interactions and body only motions, which cover the vast majority of human daily life activities. And all the annotated interacted objects in interactions all belong to COCO 80 object categories [35].

B.1. Activity and PaSta

PaStaNet can provide abundant activity knowledge for both instance and part levels and help construct a large-scale activity parsing tree, as seen in Fig. 8. Moreover, we can represent the parsing tree as a co-occurrence matrix of the activities and PaSta. A part of the matrix is depicted in Fig. 9.
Figure 8. Part of the Activity Parsing Tree.

Figure 9. Part of the Activity-Pasta Co-occurrence matrix.

C. Additional Details of PaSta-R

Fig. 10 gives more details about the different PaSta-R implementations, i.e. directly input the Activity2Vec output to the Linear Combination, MLP or GCN [32], sequential LSTM-based model and the tree-structured passing model.

D. Additional Details of MNIST-Action

In this section, we provide some details of the MNIST-Action experiment. The instance-based and hierarchical models are shown in Fig. 11. The train and test set sizes are 5,000 and 800. In the instance-based model, we directly use the ROI
pooling feature of the digit union box to predict the target (summation of the largest and second largest numbers). We use optimizer RMSProp and the initial learning rate is 0.0001. The batch size is 16 and we train the model 1K epochs. In the hierarchical model, we first operate the single digit recognition and then use single digit features (part features) together with the instance feature to infer the sum (early fusion). If using late fusion, we directly fuse the scores of instance branch and part branch. We also use optimizer RMSProp and the initial learning rate is 0.001. The batch size is 32 and the training also costs 1K epochs. Two models are all implemented with 4 convolution layers with subsequent fully-connect layers.

Results are shown in Fig. 12. We can find that the hierarchical method largely outperforms the instance-based method.

**E. Effectiveness on Few-shot Problems**

In this section, we show more detailed results on HICO [4] to illustrate the effectiveness of Pasta on few-shot problems. We divide the 600 HOIs into different sets according to their training sample numbers. On HICO [4], there is an obvious positive correlation between performance and the number of training samples. From Fig. 13 we can find that our hierarchical
Figure 12. Comparison of loss and accuracy in MNIST-Action.

Figure 13. Effectiveness on few-shot problems. The y-coordinate means the mAP of different sets. The x-coordinate indicates the activity sets. For example, $i$ means the number of training images is equal to or less than $i$, if $i$ is 1 then it means one-shot problem. Our approach can obviously improve the performance on few-shot problem, for the reason of reusability and transferability of PaSta.

method outperforms the previous state-of-the-art [13] on all sets, especially on the few-shot sets.

F. Additional Activity Detection Results

We report visualized PaSta and activity predictions of our method in Figure 14. The ⟨body part, part verb, object⟩ with the highest scores are visualized in blue, green and red boxes, and their corresponding PaSta descriptions are demonstrated under each image with colors consisted with boxes. The final activity predictions with the highest scores are also represented. We can find that our model is capable to detect various kinds of activities covering interactions with various objects.
Figure 14. Visualized results of our method. Triplets under images are predicted PaSta. Human body part, verb and object are represented in blue, green and red. Green tick means right prediction and red cross is the opposite.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Pre-training Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>PaStaNet* (Tab. 1)</td>
<td>HICO train set</td>
</tr>
<tr>
<td>PaStaNet (Tab. 1)</td>
<td>PaStaNet w/o HICO test set</td>
</tr>
<tr>
<td>PaStaNet* (Tab. 2)</td>
<td>HICO-DET train set</td>
</tr>
<tr>
<td>PaStaNet (Tab. 2)</td>
<td>PaStaNet w/o HICO-DET test set</td>
</tr>
<tr>
<td>Tab. 3</td>
<td>PaStaNet w/o VCOCO</td>
</tr>
<tr>
<td>Tab. 4 (AVA)</td>
<td>PaStaNet</td>
</tr>
<tr>
<td>Sec. 5.4</td>
<td>PaStaNet w/o HICO-DET</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Finetuning Data (Activity Labels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tab. 1</td>
<td>HICO train set</td>
</tr>
<tr>
<td>Tab. 2</td>
<td>HICO-DET train set</td>
</tr>
<tr>
<td>Tab. 3</td>
<td>VCOCO train set</td>
</tr>
<tr>
<td>Tab. 4 (AVA)</td>
<td>AVA train set</td>
</tr>
<tr>
<td>Sec. 5.4</td>
<td>HICO-DET train set</td>
</tr>
</tbody>
</table>

Table 8. Data usage. Tab. 3, 4 and Sec. 5.4 are transfer learning.

G. Data usage

The data usages of pre-training and finetuning are clarified in Tab. 8. We have carefully excluded the testing data in all pre-training and finetuning to avoid data pollution.