*PaSta*Net: 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, {maze1234556, 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. $\langle hand, hold, something \rangle$*, 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/.

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.

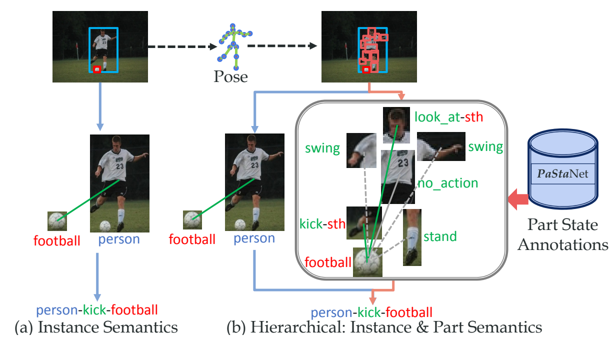


Figure 1. Instance-level and hierarchical methods. Besides the instance-level path, we perform body part states recognition in part-level with the *PaSta* annotations. With the help of *PaSta*, we can significantly boost the performance of activity understanding.

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 ***PaSta*Net** (*PaSta* is the abbreviation of Body Part State). Now *PaSta*Net includes **118 K+** images, **285 K+** persons,

250 K+ interacted objects, **724 K+** activities and **7 M+** human part states. Extensive analysis verifies that *PaSta*Net can cover most of the the part-level knowledge in general. Using learned *PaSta* knowledge in *transfer learning*, we can achieve 3.2, 4.2 and 3.2 improvements on V-COCO [25], images-based AVA [24] and HICO-DET [3] (Sec.5.4).

Given *PaSta*Net, we propose two powerful tools to promote the image-based activity understanding: 1) **Activity2Vec**: With *PaSta*Net, we convert a human instance into a vector consisting of *PaSta* representations. Activity2Vec extracts part-level semantic representation via *PaSta* recognition and combines its language representation. Since *PaSta* encodes common knowledge of activities, Activity2Vec works as a general feature extractor for both seen and unseen activities. 2) **PaSta-R**: A Part State Based Reasoning method (*PaSta-R*) is further presented. We construct a *Hierarchical Activity Graph* consisting of human instance and part semantic representations, and infer the activities by combining both instance and part level sub-graph states.

The advantages of our method are two-fold: 1) **Reusability and Transferability**: *PaSta* are basic components of actions, their relationship can be in analogy with the amino acid and protein, letter and word, *etc*. Hence, *PaSta* are **reusable**, *e.g*., $\langle hand, hold, something \rangle$ is *shared* by various actions like "hold horse" and "eat apple". Therefore, we get the capacity to describe and differentiate plenty of activities with a much smaller set of *PaSta*, *i.e*. one-time labeling and transferability. For few-shot learning, reusability can greatly alleviate its learning difficulty. Thus our approach shows significant improvements, *e.g*. we boost 13.9 mAP on one-shot sets of HICO [4]. 2) **Interpretability**: we obtain not only more powerful activity representations, but also better interpretation. When the model predicts what a person is doing, we can easily know the reasons: what the body parts are doing.

In conclusion, we believe *PaSta*Net will function as a human activity knowledge engine. Our main contributions are: 1) We construct *PaSta*Net, the first large-scale activity knowledge base with fine-grained *PaSta* annotations. 2) We propose a novel method to extract part-level activity representation named Activity2Vec and a *PaSta*-based reasoning method. 3) In supervised and transfer learning, our method achieves significant improvements on large-scale activity benchmarks, *e.g*. 6.4 (16%), 5.6 (33%) mAP improvements on HICO [4] and HICO-DET [3] respectively.

2. 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 *PaSta*Net

In this section, we introduce the construction of *PaSta*Net. *PaSta*Net 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 *PaSta*Net 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, *PaSta*Net 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

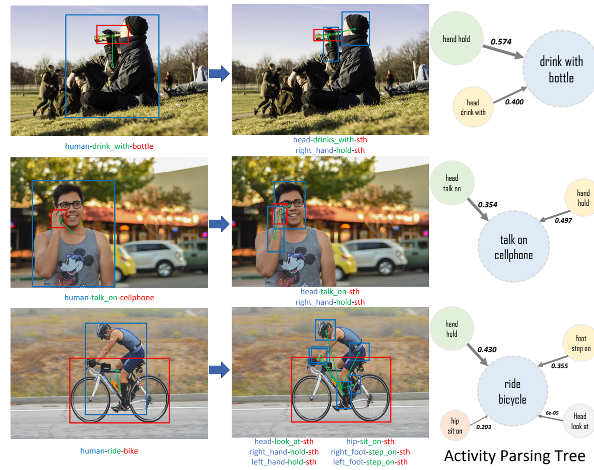


Figure 2. *PaSta* annotation. Based on instance activity labels, we add fine-grained body part boxes and corresponding part states *PaSta* labels. In *PaSta*, we use "something" [39] to indicate the interacted object for generalization. The edge in *Activity Parsing Tree* indicates the statistical co-occurrence.

it as $\langle hip, sit\_on, something \rangle$, $\langle hand, hold, something \rangle$, $\langle foot, step\_on, something \rangle$, *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* $\langle head, eat, something \rangle$, $\langle head, talk\_to, something \rangle$ and $\langle head, look\_at, something \rangle$ 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, *PaSta*Net 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 *PaSta*Net. Those images contain diverse activities and many of them are unseen in *PaSta*Net. 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 *PaSta*Net is general to activities.

Recognition Experiment. First, we find that *PaSta* can be well **learned**. A shallow model trained with a part of *PaSta*Net 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 *PaSta*Net, 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 *PaSta*Net

In this section, we discuss the activity representation by *PaSta*Net.

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

$$\mathcal{S}_{inst} = \mathcal{F}_{inst}(I, b_h, \mathcal{B}_o) \tag{1}$$

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

***PaSta*Net 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:

$$f_{PaSta} = \mathcal{R}_{A2V}(I, \mathcal{B}_p, b_o), \tag{2}$$

where $\mathcal{B}_p = \{b_p^{(i)}\}_{i=1}^{10}$ are part boxes generated from the pose estimation [15] automatically following [13] (head, upper arms, hands, hip, thighs, feet). $\mathcal{R}_{A2V}(\cdot)$ indicates the Activity2Vec, which extracts ten *PaSta* representations $f_{PaSta} = \{f_{PaSta}^{(i)}\}_{i=1}^{10}$. 2) *PaSta*-based Reasoning (*PaSta-R*), *i.e.*, from *PaSta* to activity semantics:

$$\mathcal{S}_{part} = \mathcal{F}_{PaSta-R}(f_{PaSta}, f_o), \tag{3}$$

where $\mathcal{F}_{PaSta-R}(\cdot)$ indicates the *PaSta-R*, f_o is the object feature. $\mathcal{S}_{part}$ 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 $(f_{PaSta}, f_o^{(i)})$ 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 *PaSta*Net.

4.1. Part State Recognition

With the object and body part boxes $b_o, \mathcal{B}_p$, 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:

$$a_i = \mathcal{P}_{pa}(f_p^{(i)}, f_o), \tag{4}$$

where $\mathcal{P}_{pa}(\cdot)$ is the part attention predictor. We compute cross-entropy loss $\mathcal{L}_{att}^{(i)}$ 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 $\mathcal{S}_{PaSta}^{(i)}$ 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 $\mathcal{L}_{PaSta}^{(i)}$. The total loss of *PaSta* recognition is:

$$\mathcal{L}_{PaSta} = \sum_{i}^{10}(\mathcal{L}_{PaSta}^{(i)} + \mathcal{L}_{att}^{(i)}). \tag{5}$$

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 *PaSta*Net to learn the powerful *PaSta* representations, which have good transferability. They can be used to reason out the instance actions

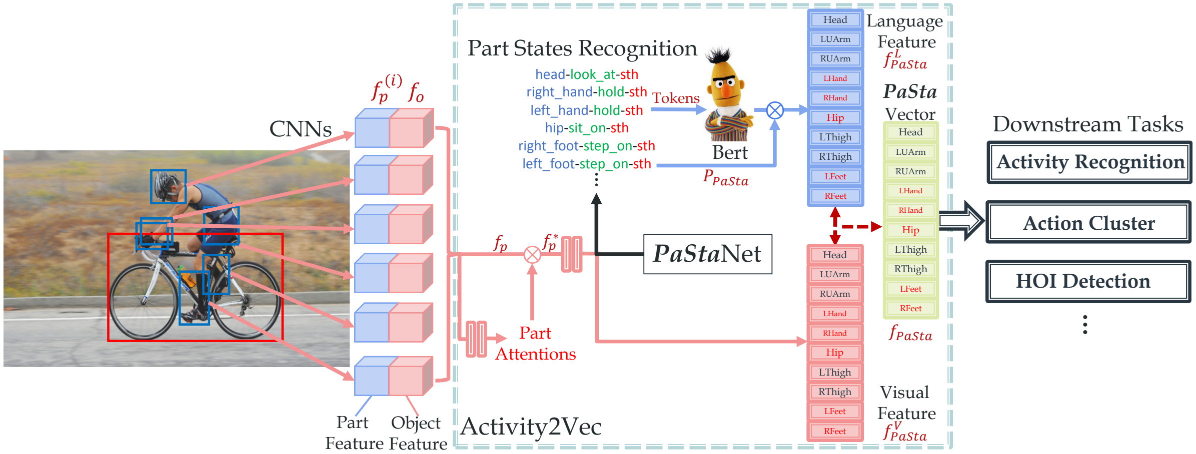


Figure 3. The overview of Part States (*PaSta*) recognition and Activity2Vec.

in both supervised and transfer learning. Under such circumstance, *PaSta*Net works like the ImageNet [8]. And *PaSta*Net 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 $f^{V(i)}_{PaSta} \in \mathbb{R}^{512}$.

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 *PaSta*, we divide each *PaSta* into tokens $\{t_p^{(i,k)}, t_v^{(i,k)}, t_o^{(i,k)}\}_{k=1}^n$, *e.g.*, $\langle part, verb, object\rangle$. The $\langle object\rangle$ comes from object detection. Each *PaSta* will be converted to a $f^{(i,k)}_{Bert} \in \mathbb{R}^{2304}$ (concatenating three 768 sized vectors of part, verb, object), *i.e.* $f^{(i,k)}_{Bert} = \mathcal{R}_{Bert}(t_p^{(i,k)}, t_v^{(i,k)}, t_o^{(i,k)})$. $\{f^{(i,k)}_{Bert}\}_{k=1}^n$ will be concatenated as the $f^{(i)}_{Bert} \in \mathbb{R}^{2304*n}$ for the i-th part. Second, we multiply $f^{(i)}_{Bert}$ with predicted *PaSta* probabilities $P^{(i)}_{PaSta}$, *i.e.* $f^{L(i)}_{PaSta} = f^{(i)}_{Bert} \times P^{(i)}_{PaSta}$, where $P^{(i)}_{PaSta} = Sigmoid(\mathcal{S}^{(i)}_{PaSta}) \in \mathbb{R}^n$, $\mathcal{S}^{(i)}_{PaSta}$ denotes the *PaSta* score of the i-th part, $P_{PaSta} = \{P^{(i)}_{PaSta}\}_{i=1}^{10}$. This means a more possible *PaSta* will get larger attention. $f^{L(i)}_{PaSta} \in \mathbb{R}^{2304*n}$ is the final language *PaSta* feature of the i-th part. We use the pre-converted and frozen $f^{(i,k)}_{Bert}$ in the whole process. Additionally, we also try to rewrite each *PaSta* into a sentence and convert it into a fixed-size vector as $f^{(i,k)}_{Bert}$ and the performance is slightly better (Sec. 5.5).

***PaSta* Representation.** At last, we pool and resize the $f^{L(i)}_{PaSta}$, and concatenate it with its corresponding visual *PaSta* feature $f^{V(i)}_{PaSta}$. Then we obtain the *PaSta* representation $f^{(i)}_{PaSta} \in \mathbb{R}^m$ for each body part (*e.g.* $m = 4096$). This process is indicated as **Activity2Vec** (Fig. 3). The output $f_{PaSta} = \{f^{(i)}_{PaSta}\}_{i=1}^{10}$ 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_{PaSta}, 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 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 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 $v_p^i \in \mathcal{V}_p$ with *PaSta* representation f^i_{PaSta} as node feature. Edge between body parts and object is $e_{po} = (v_p^i, v_o) \in \mathcal{V}_p \times \mathcal{V}_o$, and edge within parts is $e_{pp}^{ij} = (v_p^i, v_p^j) \in \mathcal{V}_p \times \mathcal{V}_p$.

Our goal is to parse HAG and reason out the graph state,

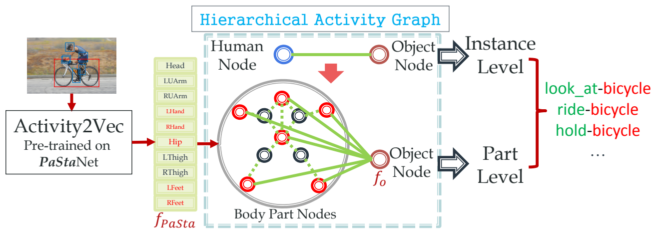


Figure 4. From *PaSta* to activities on hierarchical activity graph.

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 $\mathcal{S}_{part} = \mathcal{F}_{PaSta-R}(f_{PaSta}, f_o)$ (Eq. 3) to infer the activity scores $\mathcal{S}_{part}$. For body motion only activities *e.g*. "dance", Eq. 3 is $\mathcal{S}_{part} = \mathcal{F}_{PaSta-R}(f_{PaSta}, f_c)$, f_c is the feature of image. We adopt different implementations of $\mathcal{F}_{PaSta-R}(\cdot)$.

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 can also 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 $f^{(i)}_{PaSta}$ 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: $\mathcal{S}_{inst} = \mathcal{F}_{inst}(I, b_h, \mathcal{B}_o)$. 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}, f_o and input them to *PaSta-R*. In late fusion, we fuse the predictions of two levels, *i.e*. $\mathcal{S} = \mathcal{S}_{inst} + \mathcal{S}_{part}$. In our test, late fusion outperforms early fusion in most cases. If not specified, we use late fusion in Sec. 5. We use $\mathcal{L}^{inst}_{cls}$ and $\mathcal{L}^{PaSta}_{cls}$ to indicate

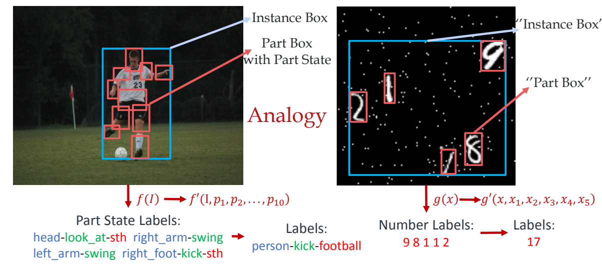


Figure 5. An analogy to activity recognition.

the cross-entropy losses of two levels. The total loss is:

$$\mathcal{L}_{total} = \mathcal{L}_{PaSta} + \mathcal{L}^{PaSta}_{cls} + \mathcal{L}^{inst}_{cls}. \tag{6}$$

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 *PaSta*Net, 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 *PaSta*Net to avoid data pollution. We

Method	mAP	Few@1	Few@5	Few@10
R*CNN [21]	28.5	-	-	-
Girdhar *et al.* [18]	34.6	-	-	-
Mallya *et al.* [42]	36.1	-	-	-
Pairwise [13]	39.9	13.0	19.8	22.3
Mallya *et al.* [42]+*PaSta*Net*-Linear	45.0	26.5	29.1	30.3
Pairwise [13]+*PaSta*Net*-Linear	**45.9**	26.2	30.6	31.8
Pairwise [13]+*PaSta*Net*-MLP	45.6	26.0	**30.8**	**31.9**
Pairwise [13]+*PaSta*Net*-GCN	45.6	25.2	30.0	31.4
Pairwise [13]+*PaSta*Net*-Seq	**45.9**	25.3	30.2	31.6
Pairwise [13]+*PaSta*Net*-Tree	45.8	24.9	30.3	31.8
*PaSta*Net*-Linear	44.5	**26.9**	30.0	30.7
Pairwise [13]+GT-*PaSta*Net*-Linear	65.6	47.5	55.4	56.6
Pairwise [13]+*PaSta*Net-Linear	**46.3**	24.7	**31.8**	**33.1**

Table 1. Results on HICO. "Pairwise [13]+*PaSta*Net" means the late fusion of [13] and our part-level result. Few@i indicates the mAP on few-shot sets. @i means 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. "*PaSta*Net-x" means different *PaSta-R*.

adopt different data mode to pre-train Activity2Vec: 1) **"*PaSta*Net*"** mode (38K images): we use the images in HICO train set and their *PaSta* labels. The **only additional supervision** here is the *PaSta* annotations compared to conventional way. 2) **"GT-*PaSta*Net*"** mode (38K images): the data used is same with **"*PaSta*Net*"**. To verify the **upper bound** of our method, we use the ground truth *PaSta* (binary labels) as the predicted *PaSta* probabilities in Activity2Vec. This means we can recognize *PaSta* perfectly and reason out the activities from the best starting point. 3) **"*PaSta*Net"** mode (118K images): we use all *PaSta*Net images with *PaSta* labels except the HICO testing data.

Settings. We use *image-level PaSta* labels to train Activity2Vec. Each image-level *PaSta* label is the aggregation over all existing *PaSta* of all active persons in an image. For *PaSta* recognition, *i.e.*, we compute the mAP for the *PaSta* categories of each part, and compute the *mean* mAP of all parts. To be fair, we use the person, body part and object boxes from [13] and VGG-16 [53] as the backbone. The batch size is 16 and the initial learning rate is 1e-5. We use SGD optimizer with momentum (0.9) and cosine decay restarts [37] (the first decay step is 5000). The pre-training costs 80K iterations and fine-tuning costs 20K iterations. Image-level *PaSta* 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. ***PaSta*Net*** 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 *PaSta* recognition in ***PaSta*Net*** mode: **46.3** (head), **66.8** (arms), **32.0** (hands), **68.6** (hip), **56.2** (thighs), **65.8** (feet). This verifies that *PaSta* can be better learned than activities, thus they can be learned ahead as the basis for reasoning. In **GT-*PaSta*Net*** mode, hierarchical paradigm achieves **65.6** mAP. This is a powerful proof of the effectiveness of *PaSta* knowledge. Thus what remains to do is to improve the *PaSta* recognition and further promote the activity task performance. Moreover, in ***PaSta*Net** 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 *PaSta*.

	Default			Known Object		
Method	Full	Rare	Non-Rare	Full	Rare	Non-Rare
InteractNet [19]	9.94	7.16	10.77	-	-	-
GPNN [49]	13.11	9.34	14.23	-	-	-
iCAN [17]	14.84	10.45	16.15	16.26	11.33	17.73
TIN [34]	17.03	13.42	18.11	19.17	15.51	20.26
iCAN [17]+*PaSta*Net*-Linear	19.61	17.29	20.30	22.10	20.46	22.59
TIN [34]+*PaSta*Net*-Linear	**22.12**	**20.19**	**22.69**	**24.06**	**22.19**	**24.62**
TIN [34]+*PaSta*Net*-MLP	21.59	18.97	22.37	23.84	21.66	24.49
TIN [34]+*PaSta*Net*-GCN	21.73	19.55	22.38	23.95	22.14	24.49
TIN [34]+*PaSta*Net*-Seq	21.64	19.10	22.40	23.82	21.65	24.47
TIN [34]+*PaSta*Net*-Tree	21.36	18.83	22.11	23.68	21.75	24.25
*PaSta*Net*-Linear	19.52	17.29	20.19	21.99	20.47	22.45
TIN [34]+GT-*PaSta*Net*-Linear	34.86	42.83	32.48	35.59	42.94	33.40
TIN [34]+*PaSta*Net-Linear	**22.65**	**21.17**	**23.09**	**24.53**	**23.00**	**24.99**

Table 2. Results on HICO-DET.

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 PaSta* labels, *i.e.* each annotated person with the corresponding *PaSta* labels, to train Acitivty2Vec, and fine-tune Activity2Vec and *PaSta-R* together on HICO-DET. All testing data are excluded from pre-training and fine-tining. We follow the mAP metric of [3], *i.e.* true positive contains accurate human and object boxes ($IoU > 0.5$ with reference to ground truth) and accurate action prediction. The metric for *PaSta* detection is similar, *i.e.*, estimated part box and *PaSta* 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 80K). The pre-training and fine-tuning take 1M and 2M iterations respectively. The learning rate is 1e-3 and the ratio of positive and negative samples is 1:4. A late fusion strategy is adopted. Three modes in Sec. 5.2 and different *PaSta-R* are also evaluated.

Results. Results are shown in Tab. 2. All ***PaSta*Net*** mode methods significantly outperform the instance-level methods, which strongly prove the improvement from the learned *PaSta* information. In ***PaSta*Net*** mode, the *PaSta* detection performance are 30.2 mAP: 25.8 (head), 44.2

Method	$AP_{role}(Scenario1)$	$AP_{role}(Scenario2)$
Gupta *et al*. [25]	31.8	-
InteractNet [19]	40.0	-
GPNN [49]	44.0	-
iCAN [17]	45.3	52.4
TIN [34]	47.8	54.2
iCAN [17]+*PaSta*Net-Linear	49.2	55.6
TIN [34]+*PaSta*Net-Linear	**51.0**	**57.5**

Table 3. Transfer learning results on V-COCO [25].

Method	mAP
AVA-TF [23]	11.4
LFB-Res-50-baseline [60]	22.2
LFB-Res-101-baseline [60]	23.3
AVA-TF [23]+*PaSta*Net-Linear	**15.6**
LFB-Res-50-baseline [60]+*PaSta*Net-Linear	**23.4**
LFB-Res-101-baseline [60]+*PaSta*Net-Linear	**24.3**

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

(arms), 17.5 (hands), 41.8 (hip), 22.2 (thighs), 29.9 (feet). This again verifies that *PaSta* can be well learned. And **GT-*PaSta*Net*** (upper bound) and ***PaSta*Net** (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 *PaSta*Net, we design transfer learning experiments on large-scale benchmarks: V-COCO [25], HICO-DET [3] and AVA [24]. We first use *PaSta*Net 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, *PaSta*Net 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 *PaSta*Net, 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 fine-tuning costs 80K iterations with the learning rate as 7e-4. We select state-of-the-arts [25, 19, 49, 17, 34] as baselines and adopt the metric AP_{role} [25] (requires accurate human and object boxes and action prediction). Late fusion strategy is adopted. With the domain gap, *PaSta*Net still improves the performance by 3.2 mAP (Tab. 3.).

Image-based AVA. AVA contains 430 video clips with spatio-temporal labels. It includes 80 atomic actions consists of body motions and HOIs. We utilize all *PaSta*Net data (118K images) for pre-training. Considering that *PaSta*Net is built upon still images, we use the **frames per second** as *still images* for **image-based** instance activity detection. We adopt ResNet-50 [27] as backbone and SGD with momentum of 0.9. The initial learning rate is 1e-2 and the first decay of cosine decay restarts [37] is 350K. For a fair comparison, we use the human box from [60]. 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 AVA 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 *PaSta*Net 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. *PaSta*Net 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 TIN [34]+*PaSta**-Linear (22.12 mAP). **1) w/o Part Attention** degrades the performance with 0.21 mAP. **2) Language Feature:** We replace the *PaSta* Bert feature in Activity2Vec with: Gaussian noise, Word2Vec [44] and GloVe [48]. The results are all worse (20.80, 21.95, 22.01 mAP). If we change the *PaSta* triplet $\langle part, verb, sth \rangle$ into a sentence and convert it to Bert vector, this vector performs sightly better (22.26 mAP). This is probably because the sentence carries more contextual information.

6. Conclusion

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

Acknowledgement: This work is supported in part by the National Key R&D Program of China, No. 2017YFA0700800, National Natural Science Foundation of China under Grants 61772332 and Shanghai Qi Zhi Institute.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 2, 3

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 2, 6, 8

[3] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. In *WACV*, 2018. 1, 2, 3, 7, 8

[4] Yu Wei Chao, Zhan Wang, Yugeng He, Jiaxuan Wang, and Jia Deng. Hico: A benchmark for recognizing human-object interactions in images. In *ICCV*, 2015. 2, 3, 6, 7, 14

[5] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In *CVPR*, 2018. 2

[6] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 1990. 3

[7] Vincent Delaitre, Josef Sivic, and Ivan Laptev. Learning person-object interactions for action recognition in still images. In *NIPS*, 2011. 1, 2

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5, 8

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 5

[10] Wendy Doniger. *Reductionism*. 1999. 1

[11] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 2015. 2

[12] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 2, 3, 6

[13] Hao Shu Fang, Jinkun Cao, Yu Wing Tai, and Cewu Lu. Pairwise body-part attention for recognizing human-object interactions. In *ECCV*, 2018. 2, 3, 4, 6, 7, 15

[14] Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *ICCV*, 2019. 2

[15] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. RMPE: Regional multi-person pose estimation. In *ICCV*, 2017. 2, 3, 4

[16] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 2

[17] Chen Gao, Yuliang Zou, and Jia-Bin Huang. ican: Instance-centric attention network for human-object interaction detection. *arXiv preprint arXiv:1808.10437*, 2018. 2, 5, 6, 7, 8

[18] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In *NIPS*, 2017. 7

[19] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. In *CVPR*, 2018. 2, 7, 8

[20] Georgia Gkioxari, Ross Girshick, and Jitendra Malik. Actions and attributes from wholes and parts. In *ICCV*, 2015. 2

[21] Georgia Gkioxari, Ross Girshick, and Jitendra Malik. Contextual action recognition with r* cnn. In *ICCV*, 2015. 7

[22] Georgia Gkioxari, Bharath Hariharan, Ross Girshick, and Jitendra Malik. R-cnns for pose estimation and action detection. *Computer Science*, 2014. 2

[23] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava. `https://research.google.com/ava/download.html`, 2018. 8

[24] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 1, 2, 3, 5, 6, 8

[25] Saurabh Gupta and Jitendra Malik. Visual semantic role labeling. *arXiv preprint arXiv:1505.04474*, 2015. 2, 3, 8

[26] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7, 8

[28] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997. 6

[29] Jian Fang Hu, Wei Shi Zheng, Jianhuang Lai, Shaogang Gong, and Tao Xiang. Recognising human-object interaction via exemplar based modelling. In *ICCV*, 2014. 2

[30] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *ICCV*, 2013. 2

[31] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *TPAMI*, 2012. 2

[32] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 6, 13

[33] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, et al. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2:3, 2017. 2, 3

[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

[35] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 4, 6, 8, 12

[36] Si Liu, Lejian Ren, Yue Liao, Guanghui Ren, Hongyi Xiang, and Guanbin Li. Pic-person in context. `http://picdataset.com/challenge/index/`, 2018. 3

[37] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 7, 8

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

[39] Cewu Lu, Hao Su, Yonglu Li, Yongyi Lu, Li Yi, Chi-Keung Tang, and Leonidas J Guibas. Beyond holistic object recognition: Enriching image understanding with part states. In *CVPR*, 2018. 1, 2, 3

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

[41] S. Maji, L. Bourdev, and J. Malik. Action recognition from a distributed representation of pose and appearance. In *CVPR*, 2011. 1, 2

[42] Arun Mallya and Svetlana Lazebnik. Learning models for actions and person-object interactions with transfer to question answering. In *ECCV*, 2016. 2, 7

[43] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In *NIPS*, 1998. 7

[44] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 8

[45] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 1995. 3

[46] Bo Pang, Kaiwen Zha, Hanwen Cao, Chen Shi, and Cewu Lu. Deep rnn framework for visual sequential applications. In *CVPR*, 2019. 2

[47] Bo Pang, Kaiwen Zha, Yifan Zhang, and Cewu Lu. Further understanding videos through adverbs: A new video task. In *AAAI*, 2020. 2

[48] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 8

[49] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *ECCV*, 2018. 2, 7, 8

[50] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2, 4, 8

[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

[52] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 2

[53] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[54] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2

[55] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E. Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *ICCV*, 2015. 2

[56] Christian Thurau and Vaclav Hlavac. Pose primitive based human action recognition in videos or still images. In *CVPR*, 2008. 2

[57] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 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

[59] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 5

[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

[61] Wenqiang Xu, Yonglu Li, and Cewu Lu. Srda: Generating instance segmentation annotation via scanning, reasoning and domain adaptation. In *ECCV*, 2018. 2

[62] Weilong Yang, Yang Wang, and Greg Mori. Recognizing human actions from still images with latent poses. In *CVPR*, 2010. 1, 2

[63] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Fei Fei Li. Human action recognition by learning bases of action attributes and parts. In *ICCV*, 2011. 2

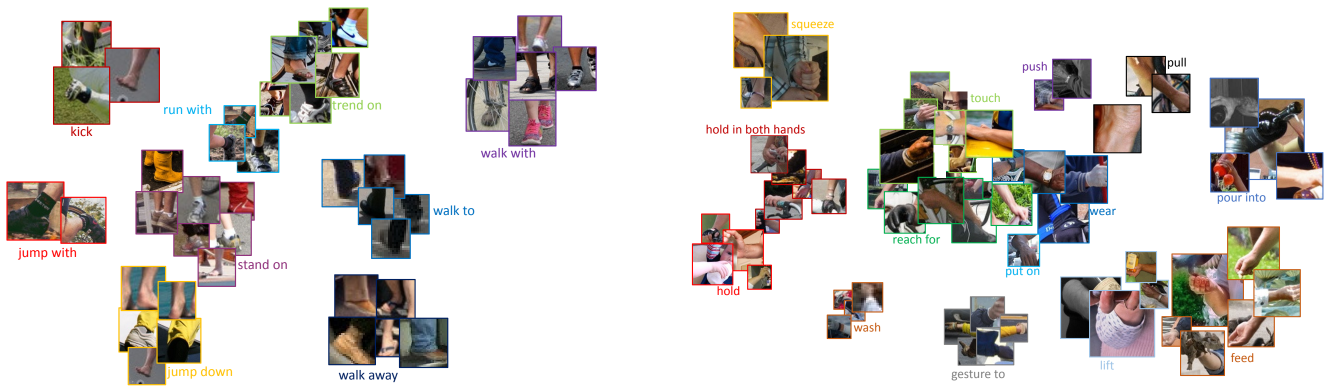
[64] Bangpeng Yao and Fei Fei Li. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010. 2

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

[66] Zhichen Zhao, Huimin Ma, and Shaodi You. Single image action recognition using semantic body part actions. In *ICCV*, 2017. 2

[67] Bohan Zhuang, Qi Wu, Chunhua Shen, Ian Reid, and Anton van den Hengel. Care about you: towards large-scale human-centric visual relationship detection. *arXiv preprint arXiv:1705.09892*, 2017. 3

Appendices

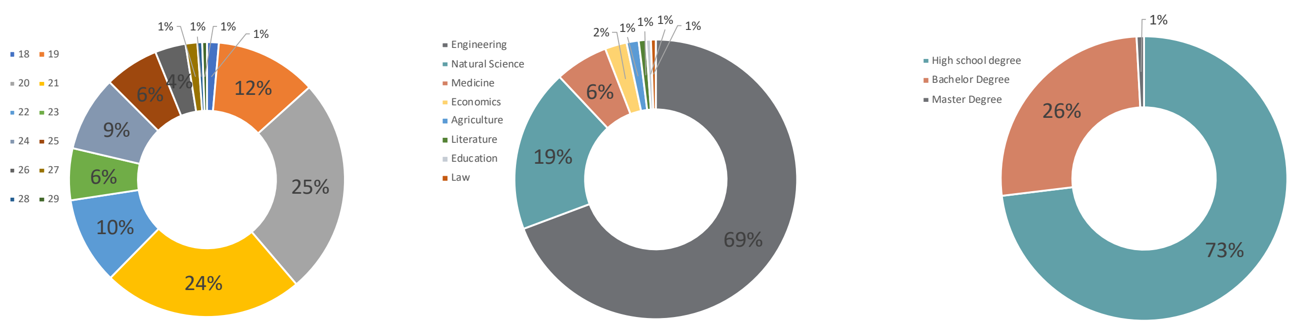


(a) *PaSta* of foot (b) *PaSta* of hand

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*.

Image	Human-Instance	Object-Instance	Activity	*PaSta*	Existing-Datasets	Crowdsourcing
118,995	285,921	250,621	724,855	7,248,550	95,027	23,968

Table 5. Characteristics of *Pasta*Net. "Existing-Datasets" and "Crowdsourcing" indicate the image sources of *Pasta*Net.



(a) Age (b) Major (c) Education Degree

Figure 7. Annotator backgrounds.

A. Dataset Details

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

A.1. Characteristics of *Pasta*Net

Tab. 5 shows some characteristics of *PaSta*Net. Now *PaSta*Net 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".

HOIs	adjust, assemble, block, blow, board, break, brush with, board gaming, buy, carry, catch, chase, check, chop, clean, clink glass, close, control, cook, cut, cut with, dig, direct, drag, dribble, drink with, drive, dry, eat, eat at, enter, exit, extract, feed, fill, flip, flush, fly, fight, fishing, give sth to sb, grab, greet, grind, groom, hand shake, herd, hit, hold, hop on, hose, hug, hunt, inspect, install, jump, kick, kiss, lasso, launch, lick, lie on, lift, light, listen to sth, listen to a person, load, lose, make, milk, move, open, operate, pack, paint, park, pay, peel, pet, play musical instrument, play with sb, play with pets, pick, pick up, point, pour, press, pull, push, put down, put on, race, read, release, repair, ride, row, run, sail, scratch, serve, set, shear, shoot, shovel, sign, sing to sb, sip, sit at, sit on, slide, smell, smoke, spin, squeeze, stab, stand on, stand under, stick, stir, stop at, straddle, swing, tag, take a photo, take sth from sb, talk on, talk to, teach, text on, throw, tie, toast, touch, train, turn, type on, walk, wash, watch, wave, wear, wield, work on laptop, write, zip
Body Motions	bow, clap, climb, crawl, dance, fall, get up, kneel, physical exercise, swim
Objects	airplane, apple, backpack, banana, baseball bat, baseball glove, bear, bed, bench, bicycle, bird, boat, book, bottle, bowl, broccoli, bus, cake, car, carrot, cat, cell phone, chair, clock, couch, cow, cup, dining table, dog, donut, elephant, fire hydrant, fork, frisbee, giraffe, hair drier, handbag, horse, hot dog, keyboard, kite, knife, laptop, microwave, motorcycle, mouse, orange, oven, parking meter, person, pizza, potted plant, refrigerator, remote, sandwich, scissors, sheep, sink, skateboard, skis, snowboard, spoon, sports ball, stop sign, suitcase, surfboard, teddy bear, tennis racket, tie, toaster, toilet, toothbrush, traffic light, train, truck, tv, umbrella, vase, wine glass, zebra

Table 6. Activities and related objects in our *PaSta*Net. HOIs indicate the interactions between person and object/person.

Head	eat, inspect, talk with sth, talk to, close with, kiss, raise up, lick, blow, drink with, smell, wear, no action
Arm	carry, close to, hug, swing, no action
Hand	hold, carry, reach for, touch, put on, twist, wear, throw, throw out, write on, point with, point to, use sth to point to, press, squeeze, scratch, pinch, gesture to, push, pull, pull with sth, wash, wash with sth, hold in both hands, lift, raise, feed, cut with sth, catch with sth, pour into, no action
Hip	sit on, sit in, sit beside, close with, no action
Thigh	walk with, walk to, run with, run to, jump with, close with, straddle, jump down, walk away, no action
Foot	stand on, step on, walk with, walk to, run with, run to, dribble, kick, jump down, jump with, walk away, no action

Table 7. Human body part states (*PaSta*) in our *PaSta*Net.

A.2. Annotator Backgrounds

There are about 360 annotators have participated in the construction of *Pasta*Net. 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. *PaSta*Net 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*

*PaSta*Net 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.

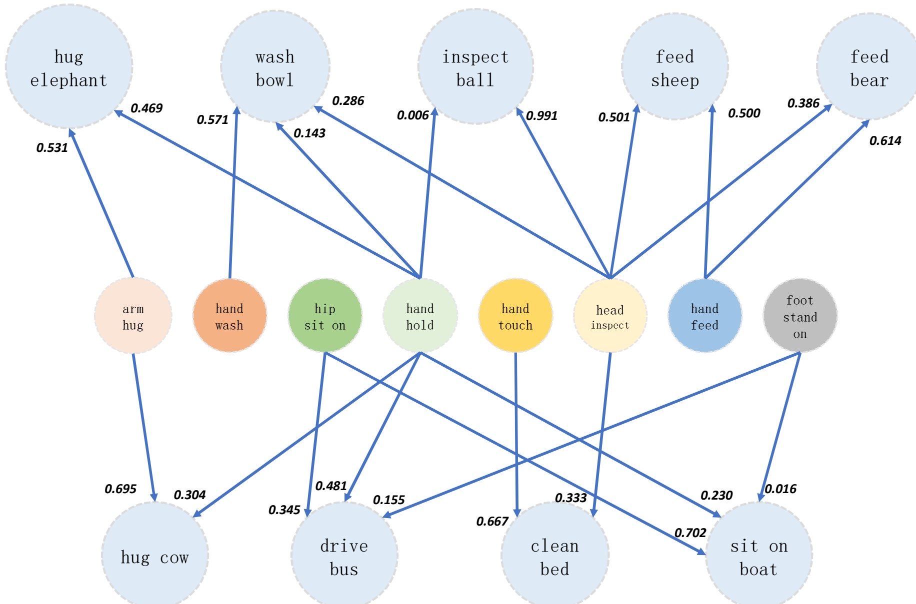


Figure 8. Part of the Activity Parsing Tree.

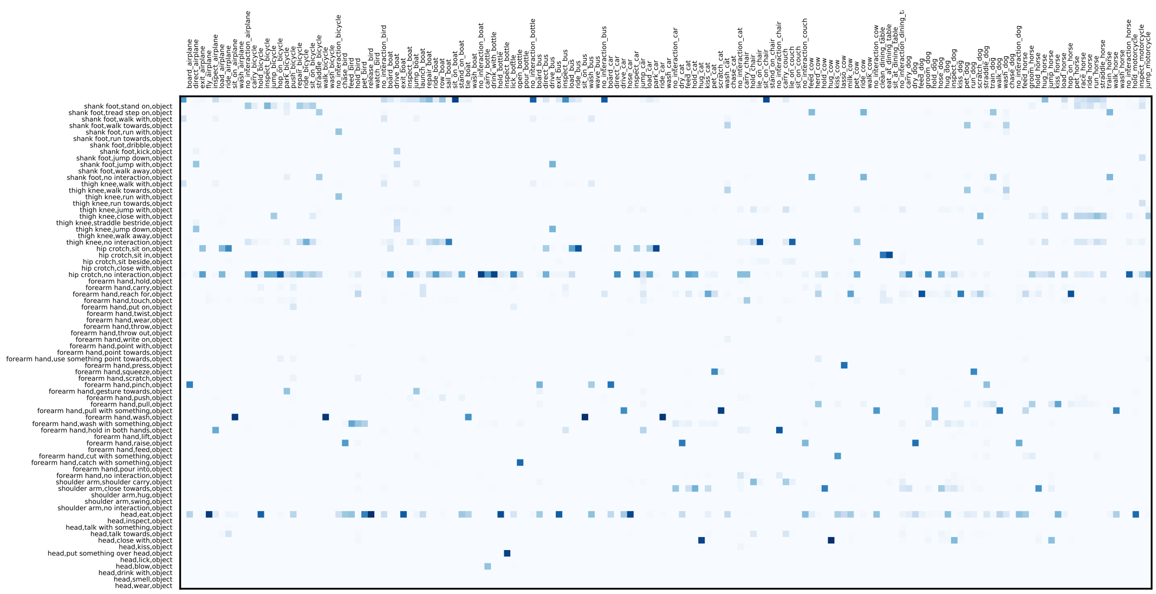


Figure 9. Part of the Activity-*Pasta* Co-occurrence matrix.

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

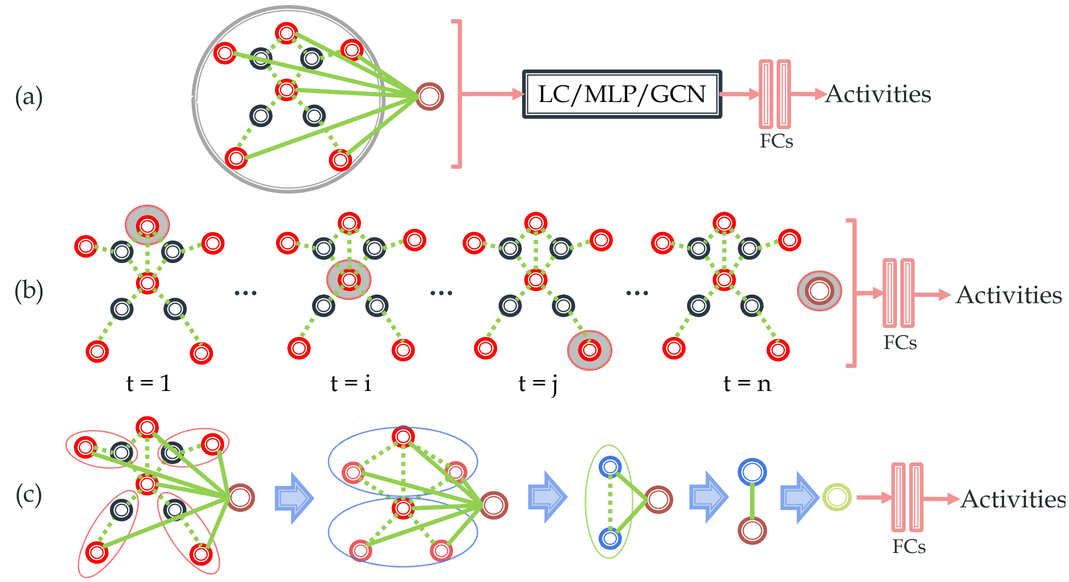


Figure 10. *PaSta-R* model: (a) Linear Combination, MLP and GCN; (b) Sequential Model; (c) Tree Structured Passing.

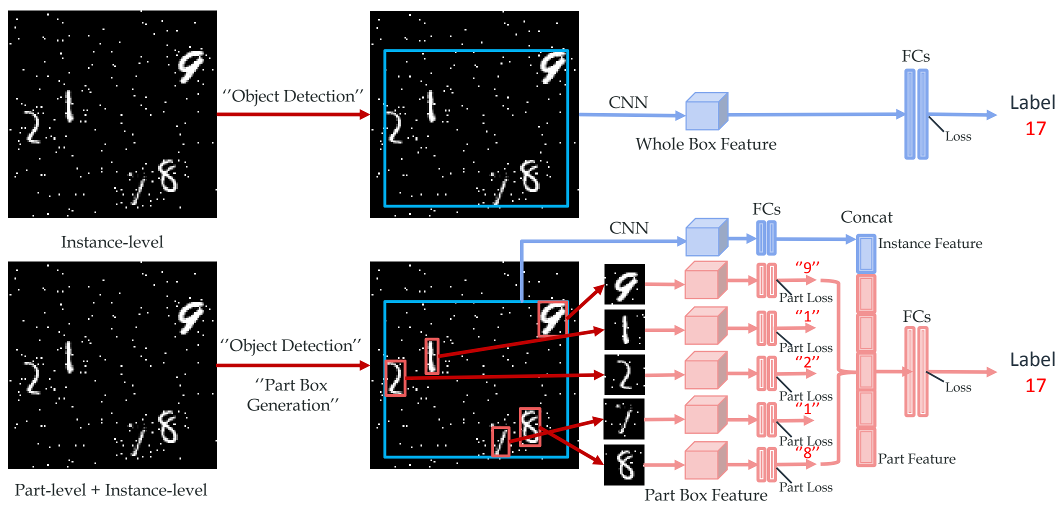


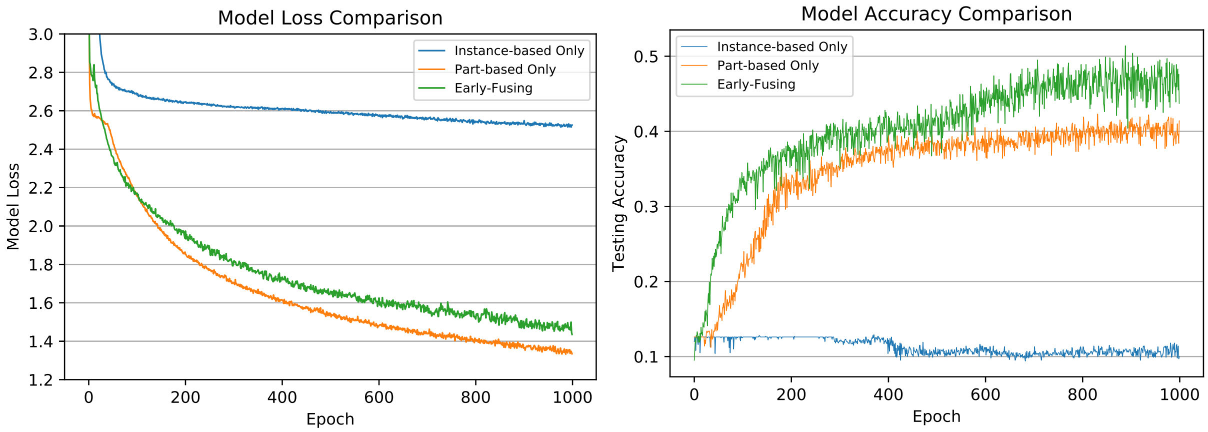
Figure 11. Instance-based and hierarchical models in MNIST-Action.

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



(a) Loss (b) Test Accuracy

Figure 12. Comparison of loss and accuracy in MNIST-Action.

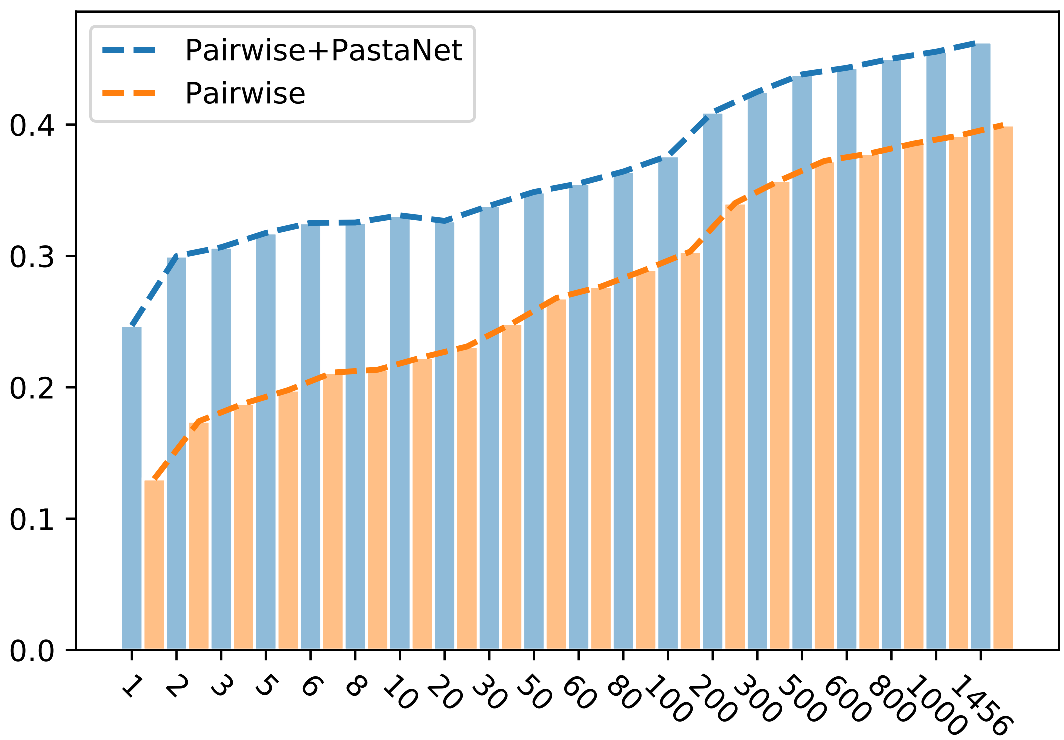


Figure 13. Effectiveness on few-shot problems. The y-coordinate means the mAP of different sets. The x-coordinate indicates the activity sets. For example, i means the number of training images is equal to or less than i, if i is 1 then it means one-shot problem. Our approach can obviously improve the 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 $\langle body\_part, part\_verb, object \rangle$ 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.

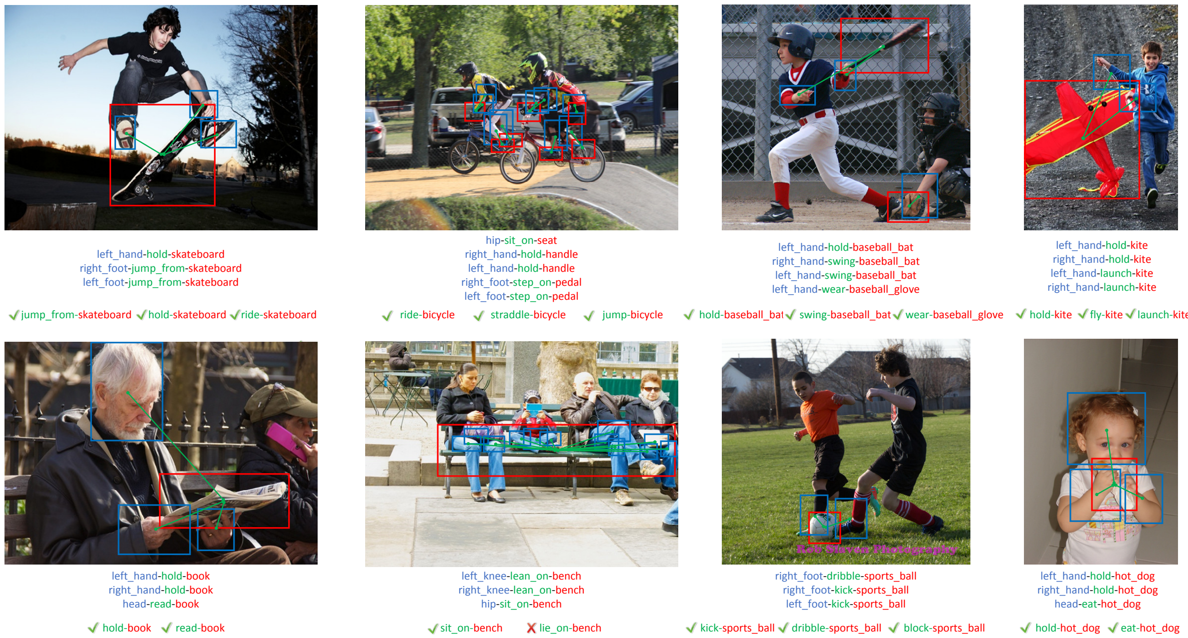


Figure 14. Visualized results of our method. Triplets under images are predicted *PaSta*. Human body part, verb and object are represented in blue, green and red. Green tick means right prediction and red cross is the opposite.

Experiment	Pre-training Data
*PaSta*Net* (Tab. 1)	HICO train set
*PaSta*Net (Tab. 1)	*PaSta*Net w/o HICO test set
*PaSta*Net* (Tab. 2)	HICO-DET train set
*PaSta*Net (Tab. 2)	*PaSta*Net w/o HICO-DET test set
Tab. 3	*PaSta*Net w/o VCOCO
Tab. 4 (AVA)	*PaSta*Net
Sec. 5.4	*PaSta*Net w/o HICO-DET
Experiment	**Finetuning Data (Activity Labels)**
Tab. 1	HICO train set
Tab. 2	HICO-DET train set
Tab. 3	VCOCO train set
Tab. 4 (AVA)	AVA train set
Sec. 5.4	HICO-DET train set

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.