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, {maezl234556, cmy_l23, 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/.

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, 64, 61, 40] 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.
250 K+ interacted objects, 724 K+ activities and 7 M+ human part states. Extensive analysis verifies that PaStaNet 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 PaStaNet, we propose two powerful tools to promote the image-based activity understanding: 1) Activity2Vec: With PaStaNet, 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., (hand, hold, something) 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 parts. 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 PaStaNet will function as a human activity knowledge engine. Our main contributions are: 1) We construct PaStaNet, 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, 45, 46, 2, 53, 50, 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, 51, 16, 31, 56, 54]. There are holistic body-level approaches [55, 7], body part-based methods [20], and skeleton-based methods [57, 11], etc. But compared with other tasks such as object detection [49] 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 [55, 64, 61, 40, 5], and global context is also proved to be effective [29, 63, 62, 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, 41, 48, 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 [63, 13, 20, 11, 40, 65]. 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. [63] 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, 60]. 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, 66, 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, 66, 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 [44] 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 ⟨head, sit_on, something⟩, ⟨hand, hold, something⟩, ⟨foot, setp_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 [44], 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 \( b_{o} \):

\[
f_{\text{PaSta}} = R_{A2V}(I, B_{p}, b_{o}),
\]

(2)

where \( B_{p} = \{b_{p}^{i}\}_{i=1}^{10} \) are part boxes generated from the pose estimation [15] automatically following [15] (head, upper arms, hands, hip, thighs, feet). \( R_{A2V}(\cdot) \) 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 [49] 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 \( L^{(i)}_{\text{att}} \) for each part and multiply \( f_{p}^{(i)} \) with its scalar attention, i.e. \( f_{p}^{(i)\ast} = f_{p}^{(i)} \times a_{i} \).

Second, we operate the PaSta recognition. For each part, we concatenate the re-weighted \( f_{p}^{(i)\ast} \) with \( f_{o} \) and input them into a max pooling layer and two subsequent 512 sized FC layers, thus obtain the PaSta score \( S^{(i)}_{\text{PaSta}} \) 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 \( L^{(i)}_{\text{PaSta}} \). The total loss of PaSta recognition is:

\[
L_{\text{PaSta}} = \sum_{i=1}^{10}(L^{(i)}_{\text{PaSta}} + L^{(i)}_{\text{att}}).
\]

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

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 \( f_{\text{PaSta}}^{V(i)} \in \mathbb{R} \times 512 \).

Language PaSta feature. Our goal is to bridge the gap between PaSta and activity semantics. Language priors are useful in visual concept understanding [38, 58]. 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 \( \{ f_p^{(i,k)}, f_o^{(i,k)}, f_v^{(i,k)} \}_{k=1}^{n} \), \( \langle \text{part}, \text{verb}, \text{object} \rangle \). The \( \langle \text{object} \rangle \) comes from object detection. Each PaSta will be converted to a \( f_{\text{Bert}} \in \mathbb{R}^{2048} \) (concatenating three 768 sized vectors of part, verb, object), i.e., \( f_{\text{Bert}} = R_{\text{Bert}}(f_p^{(i,k)}, f_v^{(i,k)}, f_o^{(i,k)}) \). \( \{ f_{\text{Bert}}^{(i,k)} \}_{k=1}^{n} \) will be concatenated as the \( f_{\text{Bert}}^{(i)} \in \mathbb{R}^{2048 \times n} \) for the \( i \)-th part. Second, we multiply \( f_{\text{Bert}}^{(i)} \) with predicted PaSta probabilities \( P_{\text{PaSta}}^{(i)} \), i.e., \( f_{\text{PaSta}}^{(i)} = f_{\text{Bert}}^{(i)} \times P_{\text{PaSta}}^{(i)} \), where \( P_{\text{PaSta}}^{(i)} = \text{Sigmoid}(\Sigma_{\text{PaSta}}^{(i)}) \in \mathbb{R}^8 \), \( \Sigma_{\text{PaSta}}^{(i)} \) 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. \( f_{\text{PaSta}}^{(i)} \in \mathbb{R}^{2304 \times n} \) is the final language PaSta feature of the \( i \)-th part. We use the pre-converted and frozen \( f_{\text{Bert}} \) in the whole process. Additionally, we also try to rewrite each PaSta into a sentence and convert it into a fixed-size vector as \( f_{\text{Bert}}^{(i,k)} \), the performance is slightly better (Sec. 5.5).

PaSta Representation. At last, we pool and resize the \( f_{\text{PaSta}}^{(i)} \), and concatenate it with its corresponding visual PaSta feature \( f_{\text{PaSta}}^{V(i)} \). Then we obtain the PaSta representation \( f_{\text{PaSta}} \in \mathbb{R}^{m} \) for each body part (e.g., \( m = 4096 \)). This process is indicated as Activity2Vec (Fig. 3). The output \( f_{\text{PaSta}} = \{ f_{\text{PaSta}}^{(i)} \}_{i=1}^{10} \) 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 \( f_{\text{PaSta}} \), 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 \( G = (V, E) \) is depicted in Fig. 4. For human-object interactions, \( V = \{ \mathcal{V}_p, \mathcal{V}_o \} \). For body only motions, \( V = \mathcal{V}_p \). 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 \( f_{\text{PaSta}} \) as node feature. Edge between body parts and object is \( e_{po} = (v_p, v_o) \in \mathcal{V}_p \times \mathcal{V}_o \), and edge within parts is \( e_{pp} = (v_p, v_p) \in \mathcal{V}_p \times \mathcal{V}_p \).

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 \( S_{\text{part}} = F_{\text{PaSta-R}}(f_{\text{PaSta}}, f_o) \) (Eq. 3) to infer the activity scores \( S_{\text{part}} \). For body motion only activities e.g. “dance”, Eq. 3 is \( S_{\text{part}} = F_{\text{PaSta-R}}(f_{\text{PaSta}}, f_c) \). \( f_c \) is the feature of image. We adopt different implementations of \( F_{\text{PaSta-R}}(\cdot) \).

**Linear Combination.** The simplest implementation is to directly combine the part node features linearly. We concatenate the output of Activity2Vec \( f_{\text{PaSta}} \) 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 MLP subsequently.

**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 \( f_{\text{PaSta}} \) 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: \( S_{\text{inst}} = F_{\text{inst}}(I, b_o, B_o) \). To get the final result upon the whole graph, we can use either early or late fusion. In early fusion, we concatenate \( f_{\text{inst}} \) with \( f_{\text{PaSta}}, f_o \) and input them to PaSta-R. In late fusion, we fuse the predictions of two levels, i.e. \( S = S_{\text{inst}} + S_{\text{part}} \). In our test, late fusion outperforms early fusion in most cases. If not specified, we use late fusion in Sec. 5. We use \( \mathcal{L}_{\text{cls}} \) and \( \mathcal{L}_c \) to indicate the cross-entropy losses of two levels. The total loss is:

\[
\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{PaSta}} + \mathcal{L}_{\text{inst}} + \mathcal{L}_c.
\]

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 (28×28×1) and generate 128×128×1 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
and different object boxes from \([\text{GT} - \text{PaSta}]\). To be fair, we use the person, body part and categories of each part, and compute the \(\text{PaSta}^{\text{Full}}\) per bound of our method, we use the ground truth \(\text{PaSta}\) the data used is same with \(\text{TIN}\). We use SGD optimizer with momentum (0.9) and cosine training costs 80K iterations and fine-tuning costs 20K iterations.

Table 1. Results on HICO. “Pairwise [13]+\text{PaStaNet}^*=\text{Linear}” 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. “\(\text{PaStaNet}^=\)“ means different \(\text{PaSta-R}\).

<table>
<thead>
<tr>
<th>Method</th>
<th>37</th>
<th>14</th>
<th>19</th>
<th>28.5</th>
<th>- - -</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-CNN [21]</td>
<td>28.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Girdhar et al. [18]</td>
<td>34.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Mallya et al. [41]</td>
<td>36.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Pairwise [13]</td>
<td>39.9</td>
<td>13.0</td>
<td>19.8</td>
<td>22.3</td>
<td>-</td>
</tr>
<tr>
<td>Mallya et al. [11]+\text{PaStaNet}^*=\text{Linear}</td>
<td>45.0</td>
<td>26.5</td>
<td>29.1</td>
<td>30.3</td>
<td>-</td>
</tr>
<tr>
<td>Pairwise [13]+\text{PaStaNet}^*\text{Linear}</td>
<td>45.9</td>
<td>26.2</td>
<td>30.6</td>
<td>31.8</td>
<td>-</td>
</tr>
<tr>
<td>Pairwise [13]+\text{PaStaNet}^*-\text{MLP}</td>
<td>45.6</td>
<td>26.0</td>
<td>30.8</td>
<td>31.9</td>
<td>-</td>
</tr>
<tr>
<td>Pairwise [13]+\text{PaStaNet}^*-\text{GCN}</td>
<td>45.6</td>
<td>25.2</td>
<td>30.0</td>
<td>31.4</td>
<td>-</td>
</tr>
<tr>
<td>Pairwise [13]+\text{PaStaNet}^*-\text{Seq}</td>
<td>45.9</td>
<td>25.3</td>
<td>30.2</td>
<td>31.6</td>
<td>-</td>
</tr>
<tr>
<td>Pairwise [13]+\text{PaStaNet}^*-\text{Tree}</td>
<td>45.8</td>
<td>24.9</td>
<td>30.3</td>
<td>31.8</td>
<td>-</td>
</tr>
<tr>
<td>(\text{PaStaNet}^\text{Linear})</td>
<td>44.5</td>
<td>26.9</td>
<td>30.0</td>
<td>30.7</td>
<td>-</td>
</tr>
<tr>
<td>Pairwise [13]+\text{GT}\text{PaStaNet}^*-\text{Linear}</td>
<td>46.6</td>
<td>47.5</td>
<td>55.4</td>
<td>56.6</td>
<td>-</td>
</tr>
<tr>
<td>Pairwise [13]+\text{GT}\text{PaStaNet}^*-\text{Linear}</td>
<td>46.3</td>
<td>24.7</td>
<td>31.8</td>
<td>33.1</td>
<td>-</td>
</tr>
</tbody>
</table>

mAP on \(\text{PaSta}\) recognition in \(\text{PaStaNet}\) mode: 46.3 (head), 66.8 (arms), 32.0 (hands), 68.6 (hip), 56.2 (thighs), 65.8 (feet). This verifies that \(\text{PaSta}\) can be better learned than activities, thus they can be learned ahead as the basis for reasoning. In GT-\(\text{PaStaNet}\) mode, hierarchical paradigm achieves 65.6 mAP. This is a powerful proof of the effectiveness of \(\text{PaSta}\) knowledge. Thus what remains to do is to improve the \(\text{PaSta}\) recognition and further promote the activity task performance. Moreover, in \(\text{PaStaNet}\) 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 \(\text{PaSta}\).

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, 48, 34] to compare and cooperate.

Table 2. Results on HICO-DET.

<table>
<thead>
<tr>
<th>Method</th>
<th>Full</th>
<th>Default</th>
<th>Non-Rare</th>
<th>Known Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{TIN})</td>
<td>22.65</td>
<td>21.17</td>
<td>23.00</td>
<td>24.53</td>
</tr>
<tr>
<td>(\text{PaStaNet})</td>
<td>22.65</td>
<td>21.17</td>
<td>23.00</td>
<td>24.53</td>
</tr>
</tbody>
</table>

**Settings.** We use image-level \(\text{PaSta}\) labels to train Activity2Vec. Each image-level \(\text{PaSta}\) label is the aggregation over all existing \(\text{PaSta}\) of all active persons in an image. For \(\text{PaSta}\) recognition, i.e., we compute the mAP for the \(\text{PaSta}\) 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 [52] 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 \(\text{PaSta}\) and HOI predictions are all generated by Multiple Instance Learning (ML) [42] of 3 persons and 4 objects. We choose previous methods [41, 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. \(\text{PaStaNet}\) 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 [41] is largely narrowed from 3.8 to 0.9 mAP. Activity2Vec achieves 55.9 mAP on \(\text{PaSta}\) recognition in \(\text{PaStaNet}\) mode: 46.3 (head), 66.8 (arms), 32.0 (hands), 68.6 (hip), 56.2 (thighs), 65.8 (feet). This verifies that \(\text{PaSta}\) can be better learned than activities, thus they can be learned ahead as the basis for reasoning. In GT-\(\text{PaStaNet}\) mode, hierarchical paradigm achieves 65.6 mAP. This is a powerful proof of the effectiveness of \(\text{PaSta}\) knowledge. Thus what remains to do is to improve the \(\text{PaSta}\) recognition and further promote the activity task performance. Moreover, in \(\text{PaStaNet}\) 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 \(\text{PaSta}\).

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, 48, 34] to compare and cooperate.

**Settings.** We use instance-level \(\text{PaSta}\) labels, i.e., each annotated person with the corresponding \(\text{PaSta}\) labels, to train Activity2Vec, and fine-tune Activity2Vec and \(\text{PaSta-R}\) 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 (\(\text{IoU} > 0.5\) with reference to ground truth) and accurate action prediction. The metric for \(\text{PaSta}\) detection is similar, i.e., estimated part box and \(\text{PaSta}\) 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 \(\text{PaSta-R}\) are also evaluated.

**Results.** Results are shown in Tab. 2. All \(\text{PaStaNet}\) mode methods significantly outperform the instance-level methods, which strongly prove the improvement from the learned \(\text{PaSta}\) information. In \(\text{PaStaNet}\) mode, the \(\text{PaSta}\) detection performance are 30.2 mAP: 25.8 (head), 44.2
Like the ImageNet detection, we adopt ResNet-50 [39] as backbone and SGD for training costs 1.1M iterations and fine-tuning costs 710K iterations. Considering that pre-training and fine-tuning cost 300K and 1.3M iterations. 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).

5.4. 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 AVA [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 fine-tuning costs 80K iterations with the learning rate as 7e-4. We select state-of-the-arts [25, 19, 48, 17, 34] as baselines and adopt the metric AProle [25] (requires accurate human and object boxes and action prediction). Late fusion strategy is adopted. With the domain gap, PaStaNet still improves the performance by 3.2 mAP (Tab. 3.).

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

<table>
<thead>
<tr>
<th>Method</th>
<th>AProle (N@correct)</th>
<th>AProle (N@correct)</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 [48]</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>LFB-Res-50-baseline</td>
<td>49.2</td>
<td>55.6</td>
</tr>
<tr>
<td>LFB-Res-101-baseline</td>
<td>51.0</td>
<td>57.5</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVA-FT [23]</td>
<td>11.4</td>
</tr>
<tr>
<td>LFB-Res-50-baseline [59]</td>
<td>22.2</td>
</tr>
<tr>
<td>LFB-Res-101-baseline [59]</td>
<td>23.3</td>
</tr>
<tr>
<td>AVA-TF [23] + PaStaNet</td>
<td>15.8</td>
</tr>
<tr>
<td>LFB-Res-50-baseline [59] + GloVeNet-Linear</td>
<td>23.4</td>
</tr>
</tbody>
</table>

**ResNet-101 [27] provided by the AVA website [23]. Recent works mainly use a spatial-temporal model such as 13D [2]. Although unfair, we still employ two video-based baselines [59] 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 with **no 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 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 [43] and GloVe [47]. The results are all worse (20.80, 21.95, 22.01 mAP). If we change the PaSta triplet (**part, verb, sth**) into a sentence and convert it to Bert vector, this vector performs slightly 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 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.

**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


[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: A video dataset of spatio-temporally localized atomic visual actions. In CVPR, 2018. 1, 2, 3, 5, 6, 8


[33] 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

[34] 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


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


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


[50] 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


[57] 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] 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. Action recognition with exemplar based 2.5d graph matching. In ECCV, 2012. 1, 2
