Visual Recognition and Synthesis of Human-Object Interactions

by

Yu-Wei Chao

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2019

Doctoral Committee:

Assistant Professor Jia Deng, Chair
Associate Professor Jason Corso
Professor Benjamin Kuipers
Professor Rada Mihalcea

Yu-Wei Chao

ywchao@umich.edu

ORCID iD: 0000-0002-5476-4343

To those I met on the journey

ACKNOWLEDGMENTS

The quest for a PhD has been a long journey, and this dissertation is a product from the interactions with those I met on the way.

First and foremost, I wish to thank my advisor Professor Jia Deng, whose presence has redefined the course of this journey. His unparalleled support and diligent mentorship has made being his student an incomparable privilege. I cannot be more fortunate to have him as my advisor.

I would also like to thank Professor Jason Corso, Professor Benjamin Kuipers, and Professor Rada Mihalcea. It is my honor to have them on my dissertation committee.

Many thanks to my colleagues at the Vision and Learning Lab: Weifeng Chen, Ankit Goyal, Oana Ignat, Hei Law, Lanlan Liu, Alejandro Newell, Jonathan Stroud, Zachary Teed, Jian Wang, Mingzhe Wang, Dawei Yang, Kaiyu Yang, and Zhefan Ye. Those stressful times have never been so stressful with their company.

I am also heartily grateful to Professor Silvio Savarese, who provided a vessel for computer vision research during my Master's study, Caroline Pantofaru, who offered significant guidance during this time, and also the people in the Vision Lab: Yingze Bao, Wongun Choi, Axel Furlan, Giorgio Gemignani, Byungsoo Kim, Laura Leal-Taixe, Jie Li, Yali Li, Roni Mittelman, Lorenzo Seidenari, Changkyu Song, Min Sun, Ryan Tokola, Yu Xiang, and Zhen Zeng. A special mention goes to Wongun Choi, who provided me an example of a top-notch researcher in the early stage of my graduate study and has been a role model for me ever since.

This journey has also been enriched by some amazing internship experiences. I am thankful to my mentors throughout those times: Jimei Yang, Brian Price, Scott Cohen at Adobe Research, and Sudheendra Vijayanarasimhan, Bryan Seybold, David Ross, Rahul Sukthankar at Google Research. A special mention goes to Jimei Yang, who has spurred many inspiring discussions and ideas even after the internship, which has shaped a large part of this dissertation.

A special thanks goes to the support of a Google PhD Fellowship, which had set the stage for me to tackle more challenging problems ahead.

Finally, my deepest gratitude goes to Chi-Ju, whose continued support throughout the years has made this journey possible, and Yueh-Kai, whose arrival at the last moment has triggered the sprint before the finish line.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The ability to perceive and understand people's actions enables humans to efficiently communicate and collaborate in society. Endowing machines with such ability is an important step for building assistive and socially-aware robots. Despite such significance, the problem poses a great challenge and the current state of the art is still nowhere close to human-level performance. This dissertation drives progress on visual action understanding in the scope of human-object interactions (HOI), a major branch of human actions that dominates our everyday life. Specifically, we address the challenges of two important tasks: visual recognition and visual synthesis.

The first part of this dissertation considers the recognition task. The main bottleneck of current research is a lack of proper benchmark, since existing action datasets contain only a small number of categories with limited diversity. To this end, we set out to construct a large-scale benchmark for HOI recognition. We first tackle the problem of establishing the vocabulary for human-object interactions, by investigating a variety of automatic approaches as well as a crowdsourcing approach that collects human labeled categories. Given the vocabulary, we then construct a large-scale image dataset of human-object interactions by annotating web images through online crowdsourcing. The new "HICO" dataset surpasses prior datasets in term of both the number of images and action categories by one order of magnitude. The introduction of HICO enables us to benchmark state-of-the-art recognition approaches and also shed light on new challenges in the realm of large-scale HOI recognition. We further discover that visual features of humans, objects, as well as their spatial relations play a central role in the representation of interaction, and the combination of three can improve the recognition outcome.

The second part of this dissertation considers the synthesis task, and focuses particularly on the synthesis of body motion. The central goal is: given an image of a scene, synthesize the course of an action conditioned on the observed scene. Such capability can predict possible actions afforded by the scene, and will facilitate efficient reactions in human-robot interactions. We investigate two types of synthesis tasks: semantic-driven synthesis and goal-driven synthesis. For semantic-driven synthesis, we study the forecasting of human dynamics from a static image. We propose a novel deep neural network architecture that extracts semantic information from the image and use it to predict future body movement. For goal-directed synthesis, we study the synthesis of motion defined by human-object interactions. We focus on one particular class of interactions—a person

sitting onto a chair. To ensure realistic motion from physical interactions, we leverage a physics simulated environment that contains a humanoid and chair model. We propose a novel reinforcement learning framework, and show that the synthesized motion can generalize to different initial human-chair configurations.

At the end of this dissertation, we also contribute a new approach to temporal action localization, an essential task in video action understanding. We address the shortcomings of prior Faster R-CNN based approaches, and show state-of-the-art performance on standard benchmarks.

CHAPTER 1

Introduction

1.1 Visual Understanding of Human Actions

As Donald Davidson described in his seminal work on action theory [39], an *action* is "anything an agent does intentionally." Humans are the main *agents* of the world—we possess the capacity to act and make choices (and we actively do so) that cause changes to our surrounding environment and ourselves. Besides, we act and make choices *intentionally* to pursue and accomplish our goals. This includes short-term goals which we accomplish on a daily basis, e.g. we choose to eat and drink in order to survive, and commute in order to work, as well as long-term or lifetime goals, e.g. we choose to exercise in order to stay healthy, and acquire new knowledge and develop new skills in order to pursue our passions.

As humans, we have the ability to perceive and interpret actions taking place around us through our vision system. When we are in a grocery store and see a man behind a cashier machine with a line of people standing before him, we can immediately recognize that he is checking out items for the customers. When we are driving down a street and observe a fast approaching child sitting on a bicycle with a helmet, we can instantly recognize that she is biking. Understanding actions of others represents a crucial ability since it informs the choice of our own actions. When we recognize the cashier, we will approach him to check out our items. When we recognize the biker, we will drive pass her cautiously.

Endowing machines with the ability to perceive and interpret human actions has been a long-standing goal of computer vision. This goal is significant from at least two aspects. First, action understanding is a key enabler for technologies that aim to provide better assistance and service to humanity: a kitchen robot that perceives a nearby person is making coffee can immediately help by grabbing cream and sugar for the person; an autonomous car that understands hitchhiking signs can pull over to let in a hitchhiker. Second, and perhaps of more importance, such capability is necessary for endowing robots with social awareness. Humans live in society, and the ability to perceive and make sense of each other's actions enables us to socially communicate and collaborate

in efficient and robust manners. Such ability will be inevitably necessary for building artificial agents that share space with human beings while at the same time be socially acceptable—a grand goal of artificial intelligence (AI). Besides, why would we consider a robot "intelligent" after all, if it is incapable of perceiving what people are doing?

Before approaching the problem, it is necessary to point out that human actions can be viewed and described at different *scopes*. The concept of an abstraction hierarchy of actions has been brought forth in many prior works [142, 9, 139, 187, 5], where different works adopt different hierarchies and use different terminologies to refer to each scope in the hierarchy. For example, Aggarwal and Ryoo [5] divide human activities into four scopes: gestures, actions, interactions, and activities. Gestures are "elementary movements of a persons body parts", such as "raising a leg"; actions are "single-person activities that may be composed of multiple gestures organized temporally", such as "walking"; interactions are "activities that involve two or more persons and/or objects", such as "hand-shaking"; group activities are "activities performed by conceptual groups composed of multiple persons and/or objects", such as "a group having a meeting". Addressing action understanding as a whole turns out to be a too ambitious goal. Therefore most prior research tackled the problem on a particular scope, which is often challenging by itself already.

1.2 Human-Object Interactions

This dissertation addresses visual action understanding in the scope of *human-object interactions* (HOI)—a class of actions that involves human agents interacting with certain physical entities in the surrounding environment. The entities can be of any type, ranging from man-made objects like furniture and vehicles to natural objects like animals. Undoubtedly, interactions with objects is a dominant class of actions and an indispensible routine that people exercise on a day to day basis for achieving their goals.

Although no explicit assumption will be made on the type of objects, a particular assumption will be made on the type of interactions—we will focus particularly on *pairwise* interactions, i.e. actions defined by a single human agent interacting with a single physical entity, such as "a man opening a fridge" and "a girl eating a sandwich". The space of human-object interactions certainly contains higher order interactions as well, including actions defined by one person interacting with multiple objects (e.g. "a chief cutting an onion with a knife") and multiple persons interacting with one object (e.g. "tug of war"). Those cases are beyond the scope of this dissertation.

After defining the scope, the next question is what specific tasks we should solve. The understanding of human-object interactions spans over a wide spectrum of tasks. This dissertation in particular studies two main visual tasks: *recognition* and *synthesis*. There are certainly other important tasks as well, such as reconstruction (i.e. recovering the 3D geometry of the person and

object from visual input), but they will not be discussed herein.

1.2.1 Recognition

The recognition task involves categorization (i.e. identify the HOI class) and localization (i.e. locate the HOI in space). The problem may resemble object recognition—a widely studied task that aims to categorize objects and localize them spatially—but have significant differences. For example, in addition to identifying object categories (e.g. "laptop", "cup"), HOI recognition also involves identifying interaction categories (e.g. "typing on", "holding"); in addition to locating the objects in space, HOI recognition also involves locating the human agents in space. Furthermore, since a scene can be presented with multiple persons and multiple objects, we also need to associate each person to the object he/she is interacting with. Therefore HOI recognition turns out to be much more challenging than object recognition. Note that this dissertation mainly addresses recognition from a single image input.

1.2.2 Synthesis

The synthesis task involves generating (or instantiating) the course of an interaction. Our main interest is visual synthesis—to generate (or instantiate) the course of an interaction conditioned on the observed scene. Depending on the setup, this task may also be referred to as "action prediction" (or "action forecasting") if the goal is to synthesize the future progression of the action based on the current visual input. For example, given the observation of a man pulling out a chair, the task is to picture how he will carry on the action (e.g. moving around and sitting down onto the chair). Clearly, this requires certain level of recognition, as we need to first build an understanding of the environment based on what we see. At the same time, the ability to synthesize also marks a deeper understanding of the scene beyond recognition as it forecasts possible futures and predicts what is possible to come in the scene. Similar to the recognition task, we focus on synthesis based on a single image observation.

1.3 Background and Related Work

We first review relevant problems and highlight the recent progress.

1.3.1 Human Detection

First and foremost, action understanding requires identifying the locations of human agents in the scene. The problem has been formulated as generating bounding boxes to enclose the regions of a person in a given image. Dalal and Triggs [37] propose Histogram of Oriented Gradients

(HOG) as a feature descriptor to encode a rigid image template, and use a sliding window classifier to check the presence of a person at each image location. This has been later extended by Felzenszwalb et al. [57, 56] to flexible part-based templates, by allowing the configuration of sub-image patches within the human region to deform. During to the recent rise of deep learning, traditional feature descriptors such as HOG have been outplayed by deep neural network (DNN) based features, leading to detectors such as R-CNN [66] that have achieved remarkable performance on many benchmarks. The output of human detection is often used to further direct the attention for action recognition [41, 155].

1.3.2 Human Pose Estimation

Besides detecting the location of humans, prior work also attempts to estimate the body pose of each individual. The problem has been formulated as identifying the locations of a set of pre-selected body joints (e.g. head, elbows, etc.). Traditional methods [213, 214] again leverage hand-engineered feature descriptors such as HOG to encode the feature of local image patches, and apply structured prediction frameworks to predict the articulated structure of human body. Similar to human detection, these methods have been outplayed by DNN based approaches [185, 184], which can directly optimize the feature representation for detecting joint locations. Apart from describing the pose, the obtained joint locations are often used as a robust mid-level representation for high-level action recognition [217, 218].

1.3.3 Object Recognition

Recognizing objects is crucial for action understanding for two reasons. First, the presented objects provide semantic information about the scene class and plausible actions that can be afforded by the scene. For example, the presence of desks and keyboards may imply an office scene, and thus "looking at a computer monitor" becomes a plausible action but not "throwing a baseball". Second, recognizing objects can be a necessary part of recognizing actions. For example, we might recognize that a man is "feeding X", but without object recognition, we are unable to tell whether he is "feeding a dog" or "feeding a cat". The performance of object recognition has been largely improved in recent years due to the introduction of large-scale datasets [43, 52, 161] and high-capacity deep neural networks [108, 66]. Consequently, it has been successfully applied in many action recognition frameworks [217, 218].

1.3.4 Action Recognition

Action recognition is traditionally formulated as a classification task: given a visual observation, determine the action class from a list of pre-defined categories. It has been studied in two

setups: (1) from a static image input and (2) from a video (i.e. frame sequence) input. When the input is a static image, prior approaches [217, 218, 29] often rely on the semantic context (e.g. human pose and the presented objects) as well as spatial relations (e.g. configurations of body parts and objects). Compared to still images, it is more common to assume a video input. A key difference is that video-based approaches can exploit motion, an essential source of information that is not available in still images. For example, Wang and Schmid [197, 198] propose dense trajectories, a widely used approach in the pre-deep learning era which captures motion of densely sampled feature points. Even after the rise of deep learning, motion feature still plays critical roles, e.g. "two-stream" models [170] that parallelly process appearance and motion features and fuse them to produce classification output.

For video-based approaches, the input is typically assumed to be temporally trimmed to fully contain a single action instance. However, an arbitrary video can be long and may contain multiple action instances as time evolves as well as action-free moments (i.e. the "background" action). Therefore the trimmed input assumption is not realistic. There has been an emerging interest in the task of temporal action localization (a.k.a. action detection) [90], where the goal is to identify the start and end time of each action instance as well as identify the action class.

1.3.5 Action Prediction

Action prediction is the capability of describing what will be performed in the upcoming future based on past observations. Compared to action recognition, less prior work has been done in this area presumably due to a higher challenge. However, the problem has begun to attract more attention in recent years. Kitani et al. [99] propose to forecast the walking trajectories of pedestrians in outdoor scenes using inverse optimal control. Lan et al. [112] aim to predict future action labels using a structured prediction framework. Vondrick et al. [191] predict DNN based representations of future frames, and further obtain action labels based on the predicted representations. Note that action prediction has also been studied in the robotics community [103], but typically relies on RGB-D sensors rather than just RGB cameras.

1.4 Challenges

We next discuss the challenges on recognition and synthesis of human-object interactions.

1.4.1 Recognition

Machine learning, specifically *supervised learning*, has been the prevailing solution for recognition problems in computer vision over the past decade. The paradigm assumes a set of data that

has been pre-labeled (a.k.a the training set) is given, and aims to learn a model that maximizes the classification outcome on this set, with the premise that the learned model will generalize to new data. The learning pipeline is typically broken down into two steps: first extracting a compact feature representation from the raw input, followed by training a classifier based on the extracted representation. A successful example of this paradigm is the recent breakthrough on object recognition, driven by the establishment of large-scale datasets with tens of thousands of object categories [43, 161], and the development of deep neural networks that can jointly optimize the feature representation and classification outcome [108, 66].

Compared to object recognition, HOI recognition still remains largely unsolved. Below we discuss two main challenges: (1) benchmark datasets and (2) representation.

1. **Benchmark datasets:** In machine learning paradigms, datasets are used for training and evaluating algorithms. Compared to standard object recognition datasets that have tens of thousands of object categories, HOI recognition has been trained and evaluated on datasets with relatively small number of HOI categories. [1] In a survey paper, Guo and Lai [72] reported that the top-used dataset for image-based action recognition between 2006 and 2013 is PASCAL VOC 2010 [52], which contains only 9 action categories. Stanford 40 Actions [219], the largest image-based action dataset before 2013, contains only 40 action categories. Some video-based action datasets have more action categories, e.g. 51 in HMDB [109], 101 in UCF101 [175], and 43 in A2D [206], but are still far behind the standard object recognition datasets.

   Besides a small number of categories, current datasets also exhibit a critical issue. In addition to recognizing objects, a crucial part of HOI recognition is to distinguish *different interactions with the same object category* (e.g. "riding a bike" versus "walking a bike" for "bike" as shown in Fig. 1.1). However, current action datasets have limited diversity of interactions with individual object categories. For example, "ride bike" is the only action category in HMDB [109] that involves the object "bike", leaving out all other possible interactions with "bike", such as "walk", "park", and "repair". Without having *diverse interactions for each object category*, HOI recognition cannot be properly evaluated as a system can recognize HOIs by recognizing objects, e.g. recognizing "ride bike" by simply recognizing "bike".

   Given the issues above, creating a new HOI recognition benchmark becomes necessary. However, this poses two fundamental questions:

   - How can we discover a vocabulary for large-scale HOI recognition, with the capacity to distinguish different plausible interactions with the same object category?

[1] Prior work adopts generic action datasets, which also include actions without direct interactions with objects.

Figure 1.1: HOI recognition requires distinguishing different types of interactions with the same object category (e.g. bike). Left: riding a bike. Right: walking a bike.

- Given a vocabulary, how can we efficiently collect and label visual examples for each HOI category?

2. **Representation:** Feature representations learned from large-scale object classification [108] have become the de facto standard for generic recognition problems. While these representations may excel in encoding the presented objects, their attributes, and the background, they are not optimized for capturing relevant information on human-object interactions. For example, the two images shown in Fig. 1.1 may have similar feature representations due to the presented objects (i.e. people and bikes) and scene context (i.e. street views), but the respective actions are very different, i.e. "riding a bike" on the left versus "walking a bike" on the right.

   The key question is: how can we build interaction-sensitive representations? To obtain task-specific representations, a common practice is to use the feature extractor trained for object classification as initialization, and optimize it further by training on the targeted recognition tasks [66, 170]. However, as aforementioned, current action datasets have limited interaction classes per object category. Therefore an object sensitive representation will already be sufficient for the recognition task, hindering the learning of interaction-sensitive representations. Without *representation optimized for interactions*, human-object interactions cannot be robustly recognized.

1.4.2 Synthesis

Action synthesis has two objectives: (1) synthesizing realistic motion and (2) synthesizing realistic appearance. Generating visual experiences with photorealistic appearance (i.e. pixels) has its own challenge and has deserved an independent research discipline—computer graphics. This dissertation bypasses such challenge and instead focuses on the synthesis of realistic *motion*. Par-

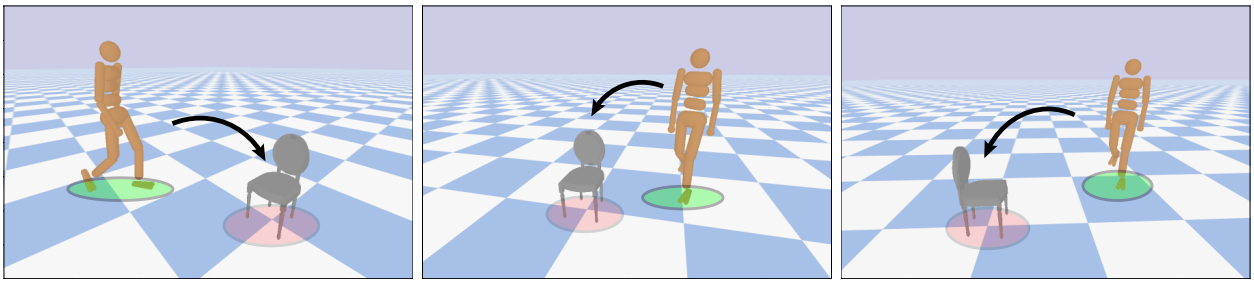


Figure 1.2: Robust synthesis of human-object interactions requires generalizing to different initial human-object configurations.

ticularly we focus on synthesizing *human body motion*. Following the common practice in human pose estimation, we model the pose of human body with an articulated skeleton, i.e. a collection of rigid body parts linked by articulated joints. The aim is to synthesize realistic kinematic movement of the skeleton model in the observed environment. Output in such form serves a useful abstract representation for actions, which can also help photorealistic synthesis by offering additional regularization using the predicted pose [190].

Motion synthesis in the context of human-object interactions is currently a research frontier that involves many challenges. Below we discuss two main challenges: (1) representation and (2) generalization.

1. **Representation:** Recall that our goal is to synthesize human-object interactions in a given scene. The scene may contain human agents with objects in a specific semantic context (i.e. the action category), and oftentimes we do not know the context a priori. In order to synthesize realistic motion, it is necessary to first infer the action context from the observation. For example, imagine you see a picture capturing the moment of a tennis player aiming at an approaching ball with a held back racket. As humans we can easily picture the upcoming dynamics: the player will swing the racket and return the ball. We do this based on not only our past observations of tennis plays, but also the context from the observation: the player's posture (e.g. standing and facing the ball), the presented objects (e.g. a tennis racket), and the background (e.g. a tennis court). Apparently a synthesis algorithm should base off these cues as well. The question is: given an image, how can we build a feature representation that encodes all the relevant context for motion synthesis?

2. **Generalization:** Even we know the specific context, we are still faced with the challenge of generalization—how can we generalize motion synthesis to different variations of the scene configuration? This turns out to be challenging especially for motion that involves interactions with the environment. Take the action "sitting onto a chair" for example (Fig. 1.2), in general a human and chair can appear in any relative position and pose in the scene, and

a system should be able to synthesize realistic sitting down motion regardless of such configuration. This implicitly assumes (at least in conventional supervised learning setups) a training set with demonstrations that sufficiently cover the space of possible configurations, which is costly and difficult to acquire. Therefore a research question is: how can we learn to robustly synthesize interactions from sparse demonstration?

1.5 Contributions

This dissertation contributes to visual understanding of human-object interactions by addressing the aforementioned challenges on recognition and synthesis.

1. **Recognition:** We tackle the challenges of dataset and representation discussed in Sec. 1.4.1. On the dataset front, Chapter 2 investigates scalable approaches to establishing a vocabulary for HOI recognition. Chapter 3 introduces a new large-scale image dataset for HOI classification. Chapter 4 further extends the dataset to address a detection scenario. On the representation front, Chapter 3 analyzes and compares state-of-the-art representations using the proposed dataset. Chapter 4 proposes a novel DNN based representation for the detection of human-object interactions.

2. **Synthesis:** We tackle the challenges of representation and generalization discussed in Sec. 1.4.2. On the representation front, we assume the particular semantic context is not know a priori and needs to be first inferred from the image (i.e. *semantic-driven* synthesis). Chapter 5 proposes a feature learning framework that learns to extract context information for future motion prediction. On the generalization front, we assume the semantic context is given, and tackle the generalization on scene configuration with few demonstrations (i.e. *goal-driven* synthesis). Chapter 6 proposes a hierarchical reinforcement learning framework to address the problem in a particular context—a person sitting onto a chair.

We should clarify that the contents above (and thus the major portion of this dissertation) only focus on image-based understanding (i.e. recognition and synthesis from a single image input). In Chapter 7, we further study action understanding in the video domain and contribute a novel approach for temporal action localization.

1.5.1 Building Vocabulary for Human-Object Interactions (Chapter 2)

While existing action datasets [109, 219, 175, 52] are unsuitable for evaluating HOI recognition, creating a new dataset immediately poses one question: how can we obtain a vocabulary for human-object interactions? Given well-established vocabularies for objects, mining semantic HOI

categories can be boiled down to answering what actions can be applied to each object category, which we refer to as the knowledge of semantic affordance. This chapter first proposes a crowdsourcing framework to collect such knowledge from humans. Using the human annotations, we then create a new benchmark and analyze a variety of automatic approaches, including text mining, visual mining, and collaborative filtering. We discover that collaborative filtering can effectively exploit the low rank structure of affordances and outperforms language and visual models. The results provide significant insights into how we can automatically establish a vocabulary for large-scale HOI recognition. This chapter is based on a joint work with Zhan Wang, Rada Mihalcea, and Jia Deng [24].

1.5.2 Classifying Human-Object Interactions (Chapter 3)

Having a vocabulary for human-object interactions, we next construct a visual dataset by collecting and annotating images from the web. We introduce a new large-scale benchmark "Humans Interacting with Common Objects" (HICO) for image-based HOI classification. HICO is distinguished from existing datasets by three key features: a diverse set of interactions with common object categories, a list of well-defined, sense-based HOI categories, and an exhaustive labeling of co-occurring interactions with an object category in each image. Given HICO, we present a thorough analysis of a number of state-of-the-art approaches on large-scale HOI classification, and show that DNN based approaches enjoy a significant edge. In addition, we tackle a data sparsity problem by proposing two knowledge transfer techniques, which lead to significant improvements in classification performance especially for uncommon categories. This chapter is based on a joint work with Zhan Wang, Yugeng He, Jiaxuan Wang, and Jia Deng [23].

1.5.3 Detecting Human-Object Interactions (Chapter 4)

Chapter 3 has formulated HOI recognition as an image-level classification problem. However, there are often multiple people and objects in the scene, and different people might be interacting with different objects. Therefore it is necessary to answer not only "what interactions are presented?" but also "who is interacting with which object?" To this end, we study the detection of human-object interactions, defined as jointly predicting a human and an object bounding box with an HOI class label. On the data front, we introduce HICO-DET, a new large-scale benchmark for HOI detection, by augmenting the HICO classification benchmark with instance annotations. On the algorithm front, we propose Human-Object Region-based Convolutional Neural Networks (HO-RCNN), a novel DNN based framework featuring the learning of spatial relations between two bounding boxes. By experimenting on HICO-DET we demonstrate significant performance improvements over baseline approaches. This chapter is based on a joint work with Yunfan Liu,

Xieyang Liu, Huayi Zeng, and Jia Deng [21].

1.5.4 Human Pose Forecasting (Chapter 5)

This chapter presents the first study on single-image human pose forecasting, a novel and significant problem in the domain of action synthesis. The task is to take a single image of a person as input, and predict the upcoming body poses as output. To address the problem, we propose 3D Pose Forecasting Network (3D-PFNet), a novel deep neural network based approach. 3D-PFNet integrates recent advances on single-image human pose estimation and sequence prediction, and converts 2D predictions into 3D space. We train the 3D-PFNet using a three-step training strategy to leverage a diverse source of training data, including image and video based human pose datasets and 3D motion capture (MoCap) data. We present qualitative and quantitative results, and show our 3D-PFNet outperforms strong baselines on 2D pose forecasting, and two state-of-the-art methods on 3D pose recovery. This chapter is based on a joint work with Jimei Yang, Brian Price, Scott Cohen, and Jia Deng [26].

1.5.5 Synthesizing Motion of Human-Object Interactions (Chapter 6)

Recent progress on physics based character animation [151, 150, 152] has shown impressive breakthroughs on human motion synthesis, through the imitation of motion capture data via deep reinforcement learning. However, these approaches have mostly been demonstrated on imitating a single distinct motion pattern, and do not generalize to interactive tasks that require flexible motion patterns due to varying human-object spatial configurations. To address this problem, we focus on one particular interactive task—sitting onto a chair. We propose a hierarchical reinforcement learning framework, which relies on a collection of subtask controllers trained to imitate simple, reusable mocap motions, and a meta controller that properly executes the subtasks to complete the main task. We experimentally demonstrate the strength of the hierarchical approach over single level approaches. We also show that our approach can be applied to motion prediction given image input. This chapter is based on a joint work with Jimei Yang, Weifeng Chen, and Jia Deng [25].

1.5.6 Temporal Action Localization (Chapter 7)

Identifying the temporal extents of actions in long video is essential for video-based action understanding. We propose TAL-Net, an improved approach to temporal action localization in video that is inspired by the Faster R-CNN object detection framework [158]. TAL-Net addresses three key shortcomings of existing approaches: (1) we improve receptive field alignment using a multi-scale architecture that can accommodate extreme variation in action durations; (2) we better exploit the temporal context of actions for both proposal generation and action classification by

appropriately extending receptive fields; and (3) we explicitly consider multi-stream feature fusion and demonstrate that fusing motion late is important. We achieve state-of-the-art performance for both action proposal and localization on THUMOS'14 detection benchmark [90] and competitive performance on ActivityNet challenge [16]. This chapter is based on a joint work with Sudheendra Vijayanarasimhan, Bryan Seybold, David A. Ross, Jia Deng, and Rahul Sukthankar [22].

1.6 Related Publications

The content of some chapters is derived from papers that have been published in various computer vision conferences. Below we list the papers that each chapter is derived from.

- Chapter 2: Yu-Wei Chao, Zhan Wang, Rada Mihalcea, Jia Deng. Mining Semantic Affordances of Visual Object Categories. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015*.
- Chapter 3: Yu-Wei Chao, Zhan Wang, Yugeng He, Jiaxuan Wang, Jia Deng. HICO: A Benchmark for Recognizing Human-Object Interactions in Images. *IEEE International Conference on Computer Vision (ICCV), 2015*.
- Chapter 4: Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, Jia Deng. Learning to Detect Human-Object Interactions. *IEEE Winter Conference on Applications of Computer Vision (WACV), 2018*.
- Chapter 5: Yu-Wei Chao, Jimei Yang, Brian Price, Scott Cohen, Jia Deng. Forecasting Human Dynamics from Static Images. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017*.
- Chapter 7: Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A. Ross, Jia Deng, Rahul Sukthankar. Rethinking the Faster R-CNN Architecture for Temporal Action Localization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018*.

In addition, Chapter 6 is derived from a manuscript submitted for publication.

- Chapter 6: Yu-Wei Chao, Jimei Yang, Weifeng Chen, Jia Deng. Learning to Sit: Synthesizing Human-Chair Interactions via Hierarchical Control. Manuscript submitted for publication, 2018.

CHAPTER 2

Building Vocabulary for Human-Object Interactions [1]

2.1 Introduction

Recent advances in object recognition have become one of the most prominent breakthroughs in computer vision over the past decade. A key enabler is the introduction of large-scale datasets such as ImageNet [43, 161]. ImageNet is featured by its scale because it provides (1) an extensive coverage (i.e. over 21K) of object categories in the visual world and (2) hundreds of image examples for each category that can be used as training examples for learning based algorithms. Inspired by the success of ImageNet, we hypothesize that a large-scale dataset of similar kind will pave the way for better recognition systems for human-object interactions (HOI).

The first question of constructing such a dataset is: *how should we obtain a vocabulary (i.e. a list of categories) for HOI recognition?* For object recognition, ImageNet acquired the vocabulary from WordNet [138], a lexical database that established a hierarchy of noun concepts. However, to the best of our knowledge, there is no established vocabulary for human-object interactions, and it is unclear how such a vocabulary can be obtained.

Fortunately, we already have an established vocabulary for object categories (from ImageNet or WordNet). The question now becomes: given an object, what actions can be performed on it? This reveals a type of knowledge that is traditionally referred to as *affordances*, a concept introduced by Gibson [64] in 1979. Affordances are fundamental attributes that reveal the functionalities of objects and the possible actions that can be performed on them. A chair affords the possibility to be sit on, but not to be eaten. A pizza affords the possibility to be eaten, but not to be hurt. While these facts might seem obvious to a human, to the best of our knowledge, there is no automated system that can acquire such knowledge and there is no knowledge base that provides comprehensive knowledge of object affordances.

The goal of this chapter is to investigate possible solutions to build such a knowledge base. To this end, we introduce the problem of mining the knowledge of *semantic affordance*: given

[1] This chapter is based on a joint work with Zhan Wang, Rada Mihalcea, and Jia Deng [24].

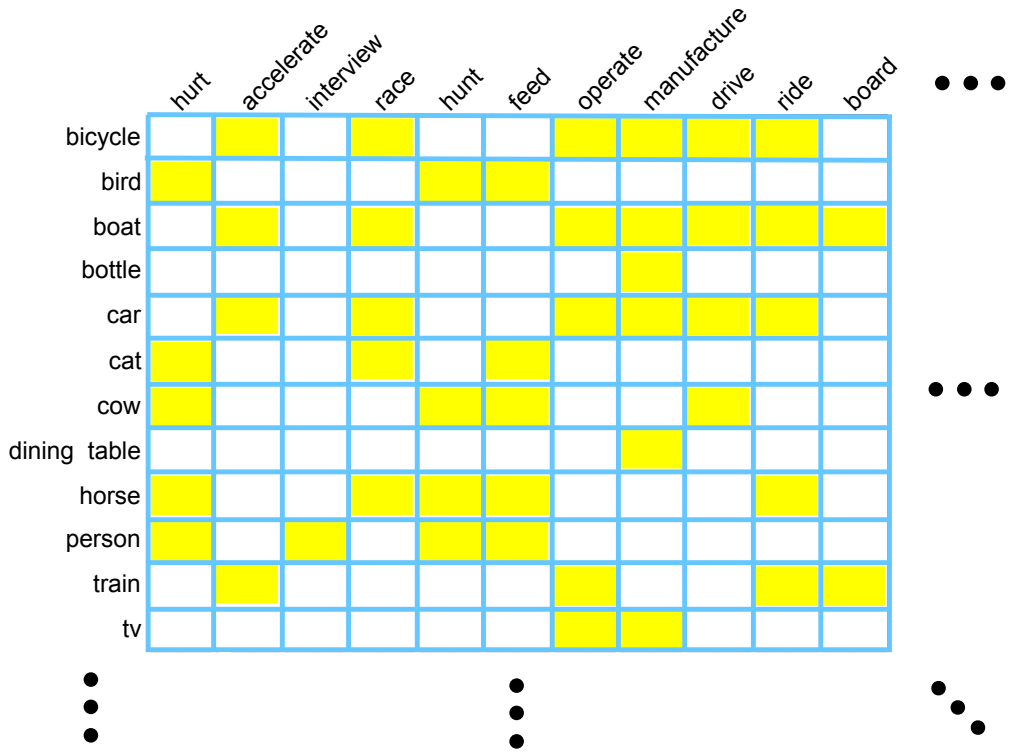


Figure 2.1: Mining the knowledge of semantic affordance is equivalent to filling an "affordance matrix" encoding the plausibility of each action-object pair.

an action and an object, determine whether the action can be applied to the object. For example, the action of "carry" forms a valid combination with "bag", but not with "skyscraper". In other words, the task is to establish connections between action concepts and object concepts, or filling an "affordance matrix" encoding the plausibility of each action-object pair (Fig. 2.1).

Note that our definition of *semantic affordance* has two important distinctions with other possible representations. First, the complete characterization of affordance is multi-dimensional—it includes not only semantic relations as explored here but also spatial information such as grasp points and human poses. Complementary to prior work that primarily addressed the spatial aspect of affordances [73, 100, 104, 221, 241], we focus on the semantic aspect, as our ultimate goal is to establish a vocabulary for recognition.

Second, the semantic affordance is defined in terms of *categories* of actions and objects, rather than individual "verbs" and "nouns". Although our task might seem similar to the linguistic problem of discovering valid verb-noun pairs, they are not equivalent. This is because the same verb or noun can have multiple meanings (senses). For example, the verb *draw* can have a meaning of *making a drawing of* or *take liquid out of a container*. To disambiguate the senses, each action or object category is represented using a WordNet [138] "synset" (a set of synonyms that have the same meaning). As a result, our task can be viewed as to enrich WordNet by drawing connections

between compatible verb synsets and noun synsets.

The key research question is: how can we collect affordance knowledge? We first propose a crowdsourcing framework to collect affordance knowledge from humans. Given the human annotated affordances, we then investigate a variety of automated approaches including text mining, visual mining, and collaborative filtering. Through text mining, we extract co-occurrence information of verb-noun combinations. Through visual mining, we discover whether images associated with a particular verb-noun combination are visually consistent. We also explore an interesting connection between the problem of affordance mining and that of collaborative filtering: can we predict if an object "likes" an action, just as a user likes a movie?

The contributions of this chapter are threefold: (1) we introduce the new problem of mining semantic affordances; (2) we create a benchmark dataset for affordance mining that contains the complete ground truth for all 20 PASCAL VOC object categories on 957 verb synsets; (3) we explore and analyze a wide variety of approaches, which provides significant insights into how we can automatically discover a vocabulary for HOI recognition. The dataset and code are available at http://www.umich.edu/~ywchao/semantic_affordance/. [2]

2.2 Related Work

2.2.1 Object Affordances

Recent work has shown that the modeling of object affordances can benefit object and action recognition [73, 100]. Gupta et al. [73] use functional constraints (human poses and motion trajectories) to aid object and action recognition. Kjellstrom et al. [100] perform simultaneous action and object recognition, showing the benefits of modeling the dependency of objects and actions. Another line of work aims to discover or predict affordances on object instances [104, 221, 241]. Koppula et al. [104] jointly predict affordance labels and activity labels on RGB-D video segments without modeling object categories. Yao et al. [221] predict affordances (expressed as spatial configurations of humans and objects) from weakly supervised images. Zhu et al. [241] predict affordance labels by reasoning over a knowledge base. The main difference between our work and prior work on affordances is that previous research has shown the importance and benefits of using affordances, but has not addressed the issue of *collecting* the knowledge of semantic affordances.

2.2.2 Action Recognition

Action recognition is a fundamental problem for general image understanding and has largely improved over the recent years [162, 198, 170]. Nonetheless, compared to state-of-the-art object

[2] The dataset has been extended to all 91 object categories of MS COCO [123].

recognition approaches, action recognition has been trained and evaluated on datasets with relatively small number of classes, e.g. 101 classes in UCF101 [175] or 487 classes in Sports-1M [96]. Furthermore, existing datasets have a very limited diversity in the action classes applied to a fixed object category, e.g. "walking dog" is the only class that involves the object "dog" in the Stanford 40 Actions dataset [219]. To advance action recognition to the next level, it is necessary to train and evaluate on a much larger number of classes. To construct such a dataset, an inevitable question would be: how can we populate the plausible action classes with regard to each object category? Mining semantic affordance provides a scalable and systemic and solution.

2.2.3 Generating Image Descriptions

Our work also adds to the line of prior work on generating natural language descriptions from images and videos [215, 141, 107, 69, 182]. Previous work typically leverages a learned language model to refine the scores of possible descriptions (e.g. subject-verb-object triplets). The language model is trained on text corpora and is mostly based on occurrence statistics. However, it is unclear how well linguistic cues can predict semantic affordances, and whether a better language model can be learned from alternative sources. Our work provides the first analysis on this question.

2.2.4 Common Sense Knowledge and Attributes

Mining semantic affordances also falls under the same category of a surge of recent work on collecting common sense knowledge [11, 18, 243, 30, 59]. The NELL [18] and the NEIL [30] project automatically extract structured knowledge from texts (NELL) and images (NEIL) respectively. An alternative line of work leverages crowdsourcing. Freebase [11] collects a large number of facts from the contribution of online volunteers. Zitnick and Parikh [243] collects visual common sense through human-created abstract images (virtual scenes composed of clip art characters and objects). Fouhey and Zitnick [59] use sequences of abstract images to learn object dynamics. Our work differs from prior work as we focus on semantic affordances, a type of common sense knowledge that has not been previously considered.

Semantic affordances can also be regarded as a special type of category-level object attributes. Therefore our work on mining semantic affordances can serve as an important basis for prior work on using attributes to improve object recognition [54, 111, 148, 169, 226, 157].

2.3 Crowdsourcing Semantic Affordances

The most reliable way of collecting affordance knowledge is presumably crowdsourcing, i.e. asking a human whether an action can be applied to an object. At the same time, it is probably also

the least scalable way: annotating all action-object pairs exhaustively would be excessively expensive. Nonetheless, it is feasible and worthwhile to collect a subset of human annotated affordances. This serves three purposes: (1) it provides insights on how humans understand object affordances; (2) it can be used as the ground truth for evaluating automatic approaches; (3) it can also be used as training data for learning-based methods.

2.3.1 Selection of Objects and Actions

For the object categories, we use the 20 object classes in PASCAL VOC [52], a widely used dataset in object recognition.

The selection of action categories is not as straightforward as that of objects. Since this work is motivated by the need for visual recognition, the desirable action categories (or verb synsets) should be both common and "visual"—meaning that they can be depicted by images or videos. This is an important concern because many verbs, especially those describing mental processes (e.g. "decide", "consider", "plan"), are quite difficult to represent visually, and there is no established way to infer the "visualness" of each verb synset.

We obtain a list of common verb synsets in two steps. First, we discover a list of common *verbs* (without disambiguating the senses). Note again the difference between verbs and verb synsets in WordNet: a verb synset is represented by one or more synonymous verbs and the same verb can appear in multiple verb synsets. We utilize the occurrence count of verbs in the Google Syntactic N-grams dataset [124]. Specifically, we first extract all verb-noun pairs with the *dobj* dependency, which ensures that we only get transitive verbs. We then sort the extracted verbs by the accumulated occurrence count, and create a top 1000 *verb* list. Second, we extract a subset of *verb synsets* by taking all WordNet verb synsets that have at least one verb in the top 1000 *verb* list. [3] This results in a list of 2375 common *verb synsets*.

To determine the "visualness" of each common verb synset, we set up a crowdsourcing task on Amazon Mechanical Turk (AMT). In this task, we first show the definition of a verb synset with synonyms and examples, as provided by WordNet. For instance,

> **align**
> definition: place in a line or arrange so as to be parallel or straight
> synonyms: align aline line_up adjust
> example: align the car with the curb

Then we ask:

[3] An additional constraint is that the WordNet count of the verb in the synset, a measure provided by WordNet, must be at least 3. This removes the cases where the verb is common but the particular verb sense is rare.

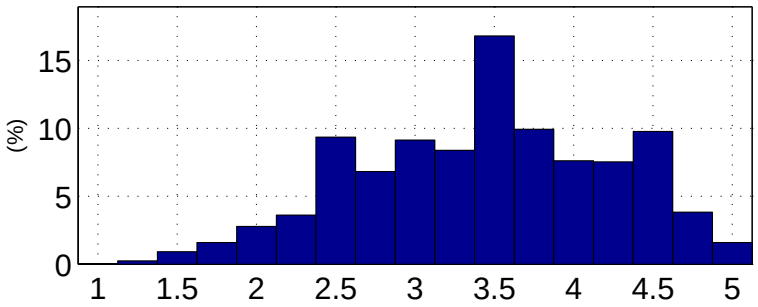


Figure 2.2: Distribution of human annotated visualness scores of common verb synsets.

> Is it possible to tell whether someone is "align-ing" (place in a line or arrange so as to be parallel or straight) something by looking at an image or watching a video clip without sound?

Note that the definition of the verb synset is repeated in the question to ensure that it is not confused with other meanings of the same verb. Given the question, a work then picks an answer from seven possible choices: "Definitely yes", "Normally yes", "Maybe", "Normally no", "Definitely no", "I don't know", and "Description doesn't make sense". To control the annotation quality, we add a quiz for each verb synset to test whether the worker understand the synset definition. We also insert a small number of gold standard questions to detect unreliable answers.

For each candidate verb synset, we collect answers from 5 different workers. Each answer is converted to a score ranging from 5.0 ("Definitely yes") to 1.0 ("Definitely no" or "Description doesn't make sense"). The "visualness" score is then calculated by the average score from the 5 workers. Fig. 2.2 shows the distribution of scores for all 2375 candidate synsets—about 30% of the synsets have a score of 4.0 or higher. Tab. 2.1 shows examples of synsets annotated with different scores. Our final selection of verb synsets is obtained by sorting the candidate synsets by visualness and retaining the synsets above a cut-off visualness score (around 3.6). This cut-off score is chosen such that we have about 1000 verb synsets. As a result, we obtain a list of 957 common and visual verb synsets.

2.3.2 Annotating Semantic Affordances

With the selected objects and actions, we then annotate semantic affordances, again, using AMT. Given an action (i.e. a verb synset) and an object (i.e. a noun synset), we ask a worker whether it is possible (for a human) to perform the action on the object. Similar to the visualness task, we first show the definition of the verb synset and then repeat the definition in the question. For instance,

> Is it possible to **hunt** (pursue for food or sport (as of wild animals)) a **car**?

Visualness	Synset Synonyms	Definition	Example Sentence
Definitely yes	{wash, launder} {drive}	Cleanse with a cleaning agent, such as soap, and water. Operate or control a vehicle.	Wash the towels, please! Drive a car or bus.
Yes	{deliver} {switch off, cut, turn off, turn out}	Bring to a destination, make a delivery. Cause to stop operating by disengaging a switch.	Our local super market delivers. Turn off the stereo, please.
Maybe	{enjoy} {respect, honor, honour, abide by, observe}	Have for one's benefit. show respect towards.	The industry enjoyed a boom. Honor your parents!
No	{intend, destine, designate, specify} {drive}	Design or destine. Compel somebody to do something, often against his own will or judgment.	She was intended to become the director. She finally drove him to change jobs.
Definitely no /Make no sense	{wish} {come}	Make or express a wish. Come to pass, arrive, as in due course.	I wish that Christmas were over. The first success came three days later.

Table 2.1: Examples of verb synsets with different visualness scores.

Plausibility	Action		Object	
	Synset Synonyms	Definition	Synset Synonyms	Definition
Definitely yes	{race, run}	Compete in a race.	{car, auto, automobile, machine, motorcar}	A motor vehicle with four wheels; usually propelled by an internal combustion engine.
	{feed, eat}	Take in food; used of animals only.	{dog, domestic dog, Canis familiaris}	A member of the genus Canis that has been domesticated by man since prehistoric times.
Yes	{repel, repulse, fight off, rebuff, drive back}	Force or drive back.	{bear}	Massive plantigrade carnivorous or omnivorous mammals with long shaggy coats and strong claws.
	{turn}	Cause to move around or rotate.	{sofa, couch, lounge}	An upholstered seat for more than one person.
Maybe	{compress, constrict, squeeze, compact, contract, press}	Squeeze or press together.	{bottle}	A glass or plastic vessel used for storing drinks or other liquids; typically cylindrical without handles.
	{repair, mend, fix, bushel, doctor furbish up, restore, touch on}	Restore by replacing a part or putting together what is torn or broken.	{wineglass}	A glass that has a stem and in which wine is served.
No	{capture, catch}	Capture as if by hunting, snaring or trapping.	{chair}	A seat for one person, with a support for the back.
	{ignite, light}	Cause to start burning; subject to fire or great heat.	{knife}	Edge tool used as a cutting instrument; has a pointed blade with a sharp edge and a handle
Definitely no /Make no sense	{cultivate, crop, work}	Prepare for crops.	{person, individual, someone somebody, mortal, soul}	A human being.
	{wear, bear}	Have on one's person.	{airplane, aeroplane, plane}	An aircraft that has a fixed wing and is powered by propellers or jets.

Table 2.2: Examples of action-object pairs with different plausibility scores.

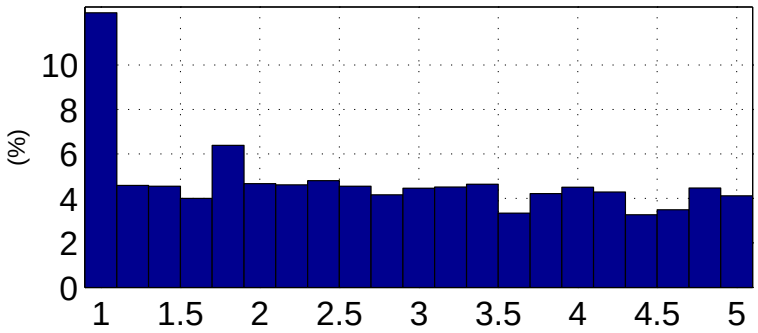


Figure 2.3: Distribution of human annotated plausibility scores for all action-object pairs.

The worker needs to choose an answer from "Definitely yes", "Normally yes", "Maybe", "Normally no", "Definitely no", "I don't know", and "Description doesn't make sense or is grammatically incorrect".

For each possible action-object combination formed by the 20 PASCAL VOC objects and the 957 verb synsets, we ask 5 workers to annotate its plausibility. This results in a total of 19K action-object questions and 96K answers. Each answer is again converted to a score from 5.0 ("Definitely yes") to 1.0 ("Definitely no" or "Description doesn't make sense or is grammatically incorrect"). The "plausibility" score of an action-object pair is then calculated by the average of 5 answers.

2.3.3 Analysis

Fig. 2.3 shows the distribution of plausibility scores for all action-object pairs—about 24% of the action-object pairs have a score of 4.0 or higher. Tab. 2.2 shows examples of action-object pairs with different plausibility scores.

How do the objects related to each other in the space of semantic affordances? It has long been hypothesized that object categories are formed based on functionality [64]. Our exhaustive annotations provide the empirical evidence to validate this hypothesis. For each object, its plausibility scores with the 957 actions form a 957-dimensional "affordance" vector. We project the affordance vectors to a 2-dimensional space using principal component analysis (PCA) and plot the coordinates of the object classes in Fig. 2.4. It is notable that the object classes form clusters that align well with a category-based semantic hierarchy—we can clearly see one cluster for vehicles, one for animals, and one for furniture. This validates the functional view of object categories.

What affordances best distinguish the different object classes? We sort the entries of the first two principal components by the maximum of their absolute values, and look at the associated verb synsets of the top entries. Fig. 2.5 shows these verb synsets and the plausibility scores for a subset of objects. This shows that affordances are indeed very discriminative attributes for object categories.

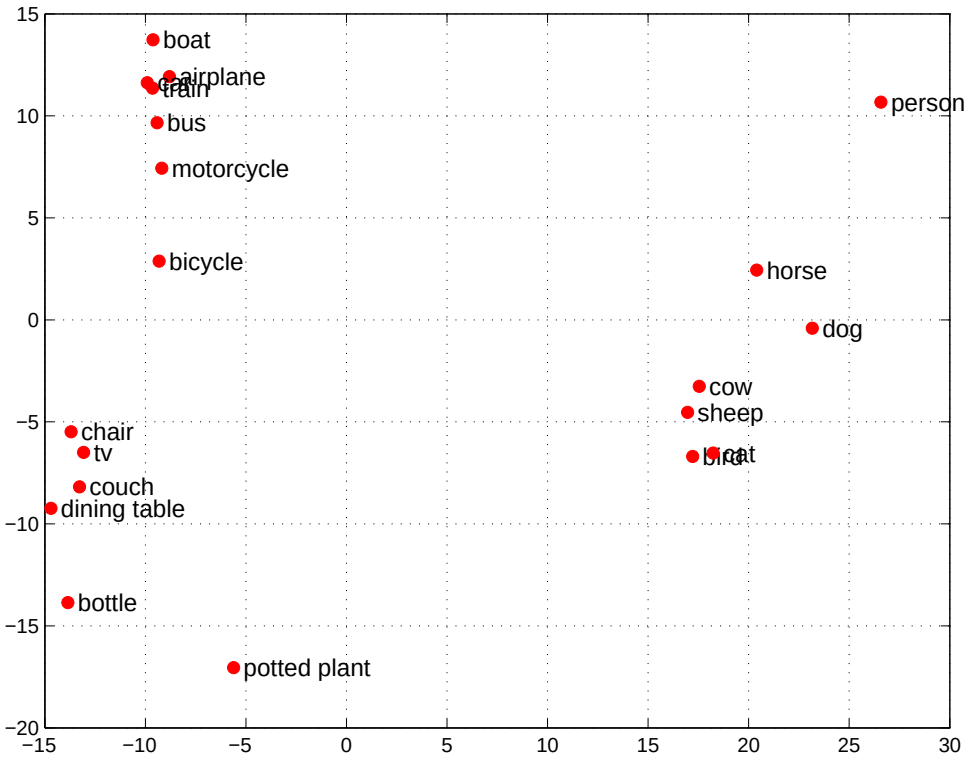


Figure 2.4: Visualizing 20 PASCAL VOC object classes in the semantic affordance space.

2.4 Mining Semantic Affordances

In this section we study possible automated approaches for mining semantic affordances. We analyze three classes of approaches: (1) mining from texts, (2) mining from images, and (3) collaborative filtering.

2.4.1 Mining from Texts

We consider the following three signals from texts to determine the plausibility of an action-object pair:

1. **N-Gram Frequency:** We assign the plausibility score by the frequency count of the verb-noun pair in Google Syntactic N-grams [124]. This is the basis of many language models used in prior work on generating descriptions from images [215, 141, 107, 69, 182].

2. **Latent Semantic Analysis (LSA):** LSA [113] is a commonly used approach to convert words to semantic vectors. This is achieved by factorizing the word-document matrix, where words that tend to co-occur in the same document would get mapped to similar vectors. The mapped vectors can then be used to compute the similarity between two words. Given a verb and a noun, we compute the cosine similarity between their mapped vectors as a proxy

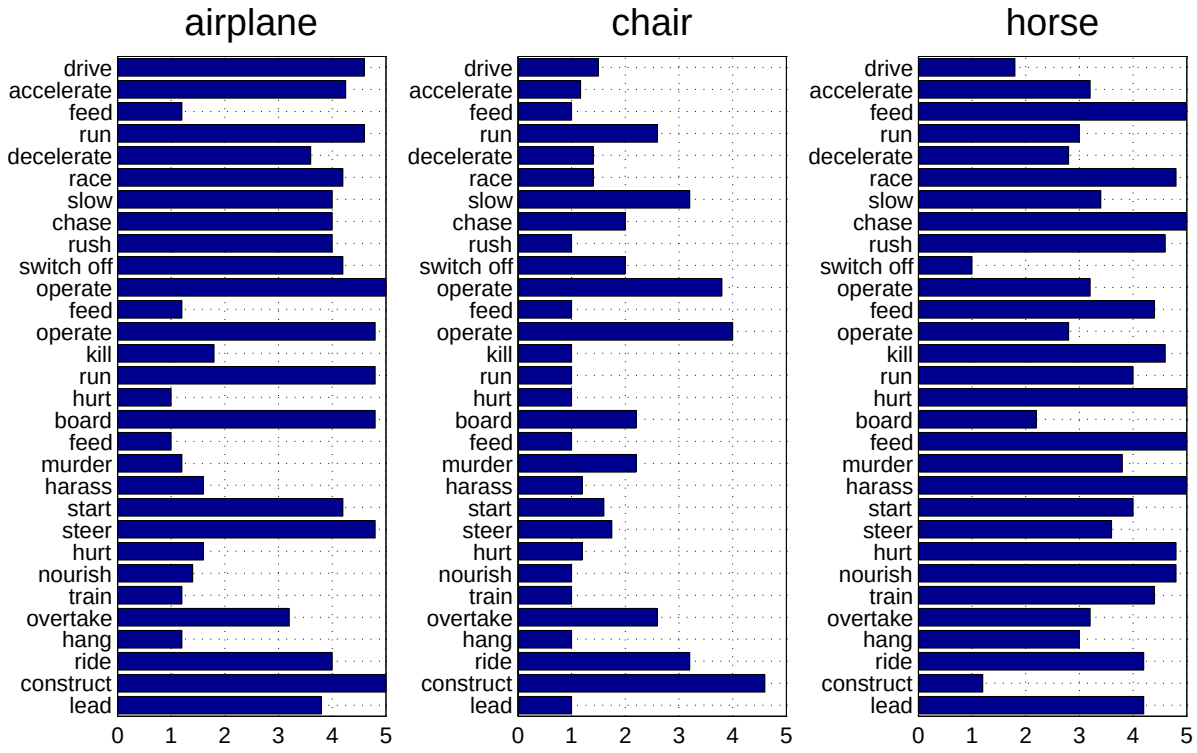


Figure 2.5: Plausibility scores on the verb synsets that have high responses in the first two principal components.

for their plausibility score. To train the model, we use the Europarl parallel corpus [101] by segmenting the corpus into sentences and training for 2 cycles.

3. **Word2Vec:** Word2Vec [137] is the state-of-the-art method for embedding words into semantic vectors. At its core is a deep neural network trained to predict the word based on its surrounding context. We use the Continuous Bag-of-Words architecture and train on the same corpus as LSA. We set the dimension to 300, window size to 5, and train for 15 iterations. Similar to LSA, we compute the cosine similarity between the verb and noun as the plausibility score for affordance.

2.4.1.1 Evaluation

To evaluate how well these signals from texts predict semantic affordances, we use the crowdsourced affordances as ground truth. We first binarize the crowdsourced plausibility scores using a threshold of 4.0 (i.e. average answer is "Normally yes" or above) [4]. Therefore the problem of predicting affordances becomes a binary classification problem: given an object and an action, predict the pair to be plausible or not. We follow the tradition of PASCAL VOC [52] by evaluating

[4]This is the threshold used throughout the chapter. We have also run all experiments with the threshold 3.0, which produce similar results and do not affect the conclusions we draw.

the average precision (AP) for each object separately and then compute the mean average precision (mAP). For each object, we rank the verb synsets using one of the textual signals and plot a precision recall curve to compute the AP.

2.4.1.2 Results

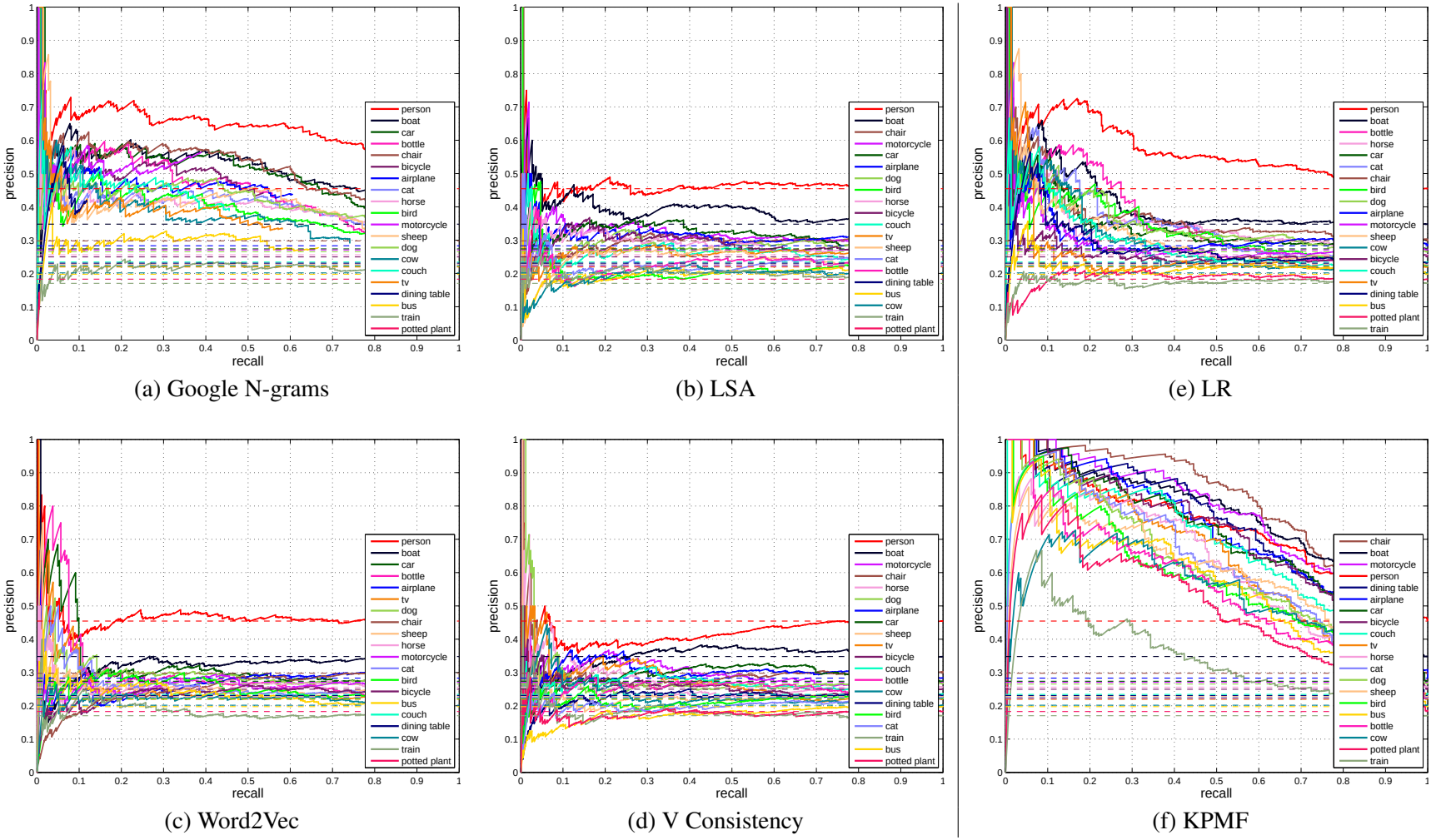
Fig. 2.6 (top left, top middle, and bottom left) plots the precision recall curves for each signal and Tab. 2.3 shows the (mean) average precision. The results show that the textual signals can indeed predict semantic affordances to some extent, although they are still far from perfect. Note that they can accurately retrieve a small number of affordances but then the precision drops rapidly with a higher recall. Surprisingly, the simplest method, Google N-grams, outperforms the more sophisticated LSA and Word2Vec, possibly because LSA and Word2Vec, by considering larger context windows, may introduce false associations between verbs and nouns, while Google N-grams only considers close-by verbs and nouns with direct dependencies.

Tab. 2.4 shows success and failure cases of the Google N-grams signal. Note that the false positives can be attributed to two cases: (1) no disambiguation of the verb (e.g. "pass a bottle" where "pass" means "go across or through"), and (2) failure in parsing the semantic dependency between the verb and noun (e.g. "feed bus" has a high count possibly because the original texts were about "twitter feed on the bus schedule"). The false negatives exhibit a more fundamental limitation with the text based signals: what if certain affordance knowledge has never been mentioned in the corpus? For example, "photograph an airplane" has a zero count in Google Syntactic N-grams, but it is a perfectly plausible action-object pair.

2.4.2 Mining from Images

In addition to textual signals, one can also mine the affordance knowledge from images. The idea is to query an image search engine with the verb-noun pair representing the action-object affordance to be predicted. Search engines can rely on historical user click data to identify the images that match the query, and the top returned images are usually good matches. Therefore under this assumption, if the affordance exists, the top returned images should be more visually coherent, while if the affordance does not exist, the returned images would be more random.

Similar ideas have been investigated by prior work (e.g. [30, 47]. For example, the LEVAN system developed by Divvala et al. [47] discovers new visual concepts by querying Google Image Search. A new concept is assumed to be visually valid if the return images are visually consistent. Following their approach, we use Google Image Search as the source of images and train a binary classifier to differentiate the top returned images against a set of random background images. The cross-validation accuracy of this classifier can then be used as a consistency measure for



(a) Google N-grams (b) LSA (e) LR

(c) Word2Vec (d) V Consistency (f) KPMF

Figure 2.6: PR curves. Each plot corresponds to different baselines: (a) occurence count from Google Syntactic N-grams, (b) LSA, (c) Word2Vec, (d) Visual Consistency, (e) logistic regression on (a),(b),(c),(d), and (f) collaborative filtering (KPMF). Dash lines represent chances.

	aero	bike	bird	boat	bottle	bus	car	cat	chair	couch	cow	table	dog	horse	mbike	person	plant	sheep	train	tv	mAP
Random	28.3	25.0	22.2	34.8	22.2	19.7	27.4	22.6	29.8	23.4	20.2	23.1	26.5	25.6	27.1	45.5	18.3	22.8	17.0	21.9	25.2
G N-grams [124]	44.1	44.4	41.4	53.9	47.9	27.5	50.0	43.3	47.8	36.7	39.5	29.5	40.6	42.2	41.2	65.3	19.5	40.6	21.6	36.7	40.7
LSA [113]	31.5	28.8	29.0	39.9	24.4	21.2	31.7	25.3	35.5	27.8	20.7	23.1	30.1	28.9	34.0	47.4	18.3	26.6	19.4	27.4	28.5
Word2Vec [137]	31.4	24.8	25.5	40.0	31.4	24.3	33.0	26.8	29.0	23.4	22.0	23.1	30.2	28.2	27.6	50.5	18.3	28.9	20.1	30.7	28.5
V Consistency	33.2	28.2	23.6	38.5	26.8	20.1	31.7	22.7	36.8	28.2	25.2	24.3	33.9	34.2	36.9	48.2	19.8	30.6	21.4	29.0	29.7
LR	35.6	33.3	38.5	45.2	39.7	23.6	39.2	38.7	38.7	31.8	34.1	30.1	36.5	39.4	35.5	60.0	20.0	34.2	18.5	30.7	35.2
NN	55.6	52.7	49.7	63.0	49.5	44.6	61.9	50.1	66.3	58.1	47.5	60.2	56.8	55.3	61.6	57.2	28.2	51.7	**41.2**	51.0	53.1
KPMF [237]	**71.1**	**69.9**	**58.3**	**76.3**	**56.4**	**56.8**	**70.2**	**62.0**	**78.1**	**67.1**	**53.6**	**71.5**	**61.8**	**62.1**	**75.2**	**73.6**	**50.5**	**59.7**	36.2	**63.3**	**63.7**

Table 2.3: Per object average precision (AP) (%) and mean average precision (mAP) (%) for a variety of automatic mining methods.

Google N-grams	Action		Object	
	Synset Synonyms	Definition	Synset Synonyms	Definition
True positives	{fly, aviate, pilot}	Operate an airplane.	{airplane, aeroplane, plane}	An aircraft that has a fixed wing and is powered by propellers or jets.
	{draw}	Represent by making a drawing of, as with a pencil, chalk, etc. on a surface.	{person, individual, someone, somebody, mortal, soul}	A human being.
	{pass, hand, reach, pass on, turn over, give}	Place into the hands or custody of.	{bottle}	A glass or plastic vessel used for storing drinks or other liquids; typically cylindrical without handles.
False positives	{fly}	Transport by aeroplane.	{airplane, aeroplane, plane}	An aircraft that has a fixed wing and is powered by propellers or jets.
	{draw, take out}	Take liquid out of a container or well.	{person, individual, someone, somebody, mortal, soul}	A human being.
	{pass,go through,go across}	Go across or through.	{bottle}	A glass or plastic vessel used for storing drinks or other liquids; typically cylindrical without handles.
	{feed, give}	Give food to.	{bus, autobus, coach, charabanc, double-decker}	A vehicle carrying many passengers; used for public transport.
False negatives	{photograph, snap, shoot}	Record on photographic film.	{airplane, aeroplane, plane}	An aircraft that has a fixed wing and is powered by propellers or jets.
	{award, present}	Give, especially as an honor or reward.	{person, individual, someone, somebody, mortal, soul}	A human being.

Table 2.4: Examples of success and failure cases for Google N-grams, the best performing text-based signal.

the returned images and also the plausibility score. Specifically, we train an SVM classifier using features extracted from the fc_7 layer of AlexNet [108] implemented in Caffe [89].

2.4.2.1 Results

We evaluate the visual consistency signal with the same metric for the individual textual signals in Sec. 2.4.1. Fig. 2.6 (bottom middle) plots the precision recall curves and Tab. 2.3 shows the (mean) average precision. Similar to textual signals, a decent precision can be achieved at a very low recall. However, the precision drops dramatically when the recall increases—in fact, not better than chances most of the time. Thus the visual signals perform much worse than textual signals.

Fig. 2.7 shows sample image search results and the corresponding cross-validation accuracy of the learned classifier (i.e. visual consistency). These results provide some insights into the sources of errors. False positives occur when Google Image Search can return images that are irrelevant to the query but are highly visually consistent due to coincidental textual match, e.g. the queries "wear bicycle" and "transport chair" return visually consistent images, but the content of the images matches poorly to the underlying concepts of the queries. False negatives occur when the search engine either fails to return sufficient relevant images (e.g. "manufacture chair"), or returns sufficient relevant images but the visual variability is to high such that the state-of-the-art feature representation is unable to learn a robust classifier (e.g. "wash bicycle").

2.4.3 Collaborative Filtering

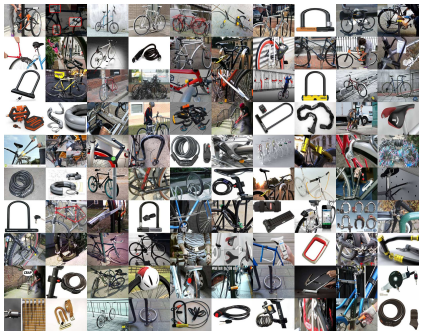
Both mining from texts and images use signals on *individual* action-object pairs. However, as suggested by the PCA result on the human annotated affordances (Fig. 2.5), the space of affordances is lower dimensional and "smooth". This observation reveals an interesting connection to the problem of collaborative filtering [177] (or matrix completion): suppose we have observed the ratings of some users on some movies, can we predict new unseen ratings? To convert it to the affordance prediction problem, we only need to replace "user" with object and "movie" with action. This leads to an alternative type of approach based on *extrapolation*, i.e. inferring new affordances based on existing ones.

A particular interesting scenario is the "cold start" setting, where for certain users (or movies) we do not have any observed ratings. This can be handled by some collaborative filtering approaches that exploit "side information"—attributes or features about the users or movies in addition to the observed ratings. Without loss of generality, side information can be expressed as similarities (kernels) between users or between movies.

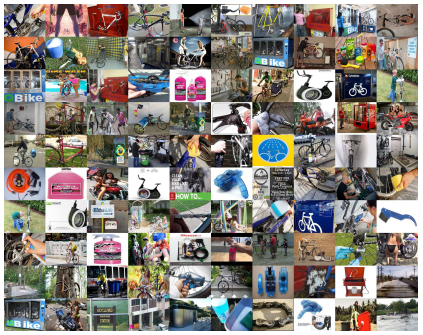
To apply this idea to affordance prediction, we adopt Kernelized Probabilistic Matrix Factorization (KPMF) [237], a state-of-the-art collaborative filtering method that allows similarity based

wear + bicycle, 90.00 %

ride + bicycle, 87.50 %

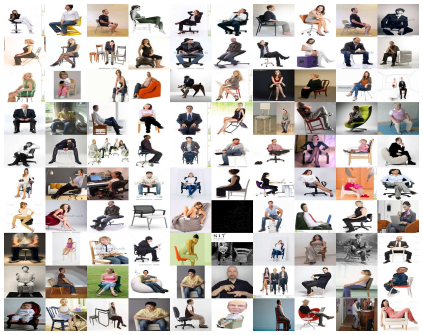


lock + bicycle, 84.00 %



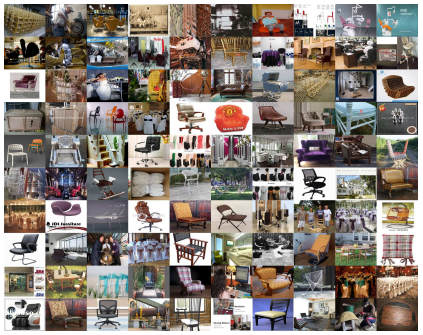
wash + bicycle, 63.50 %

transport + chair, 93.50 %



sit + chair, 92.50 %

repair + chair, 81.50 %



manufacture + chair, 73.00 %

Figure 2.7: Query keywords, the top 100 images return by Google Image Search, and visual consistency (cross-validation accuracy).

side information. In a nutshell, for N objects and M actions, the method factorizes the $N \times M$ affordance matrix into the product of a latent $N \times D$ matrix and a latent $D \times M$ matrix. There is one additional constraint: if an entire row (column) of the latent $N \times D$ ($D \times M$) matrix is missing, then that row (column) will be filled based on the observed rows (columns) and the known similarities provided as side information.

2.4.3.1 Evaluation

Following the evaluation setting for text and visual mining, we evaluate the AP for each object class separately and compute mAP. For an object class, we assume that none of its affordances is observed but the ground-truth affordances for all other 19 object classes are given. We define side information using the similarity measures (e.g. PATH, LCH, WUP [149] for noun synsets provided by WordNet. We then run KPMF to predict plausibility scores for the unobserved entries using the observed ground truths and side information. We repeat this process for all 20 object classes and plot the precision recall curves to compute AP.

We further compare KPMF with two baselines. First, we evaluate a nearest neighbor (NN) baseline: we predict the plausibility scores for each object class by simply transferring the affordance labels from the most similar object class among the other 19. Second, we learn a logistic regression (LR) classifier that linearly combines the textual and visual signals to predict the plausibility scores. For each object class, the weights of the classifier are learned using the ground-truth labels on the other 19 objects.

2.4.3.2 Results

The results are shown in Fig. 2.6 (top right and bottom right) and Tab. 2.3. Note that collaborative filtering (KPMF) outperforms textual and visual signals by a very large margin on all object classes (e.g. 63.7 mAP for KPMF versus 40.7 mAP for Google N-grams). Besides, KPMF also outperforms NN and LR, suggesting that both side information and matrix factorization are essential. Interestingly, the logistic regression classifier performs worse than Google N-grams, suggesting that the learned weights do not generalize across classes. These results also validate that collaborative filtering, by exploiting the low rank structure of the affordance matrix and side information, is indeed a promising way to predict new unseen affordances.

2.5 Summary and Future Work

We introduce the problem of mining semantic affordances, a crucial task that will facilitate the construction of large-scale action datasets. We introduce a new benchmark with crowdsourced

ground-truth affordances on 20 PASCAL VOC object classes and 957 action classes. We show that human annotated affordances have low dimensional structure that reveals the functionalities and semantic relations between objects. We evaluate a wide variety of automatic mining approaches using the human annotations as ground truths. We discover that: (1) language models based on co-occurrence statistics have substantial limitations in predicting affordances due to the challenge of sense disambiguation and inevitable data sparsity; (2) visual models are less reliable than language models in predicting affordances; (3) collaborative filtering can effectively exploit the low rank structure of affordances and outperforms language and visual models.

Our study suggests one plausible bootstrapping strategy to scale up the collection of affordance knowledge. We first select an initial set of objects and use crowdsourcing to collect high quality affordance annotations. We can then use collaborative filtering to "fill" the affordances for new unseen objects that are sufficiently "similar" to the objects in the initial set, and expand our object set. Once we encounter a new object that is sufficiently "dissimilar" to the existing objects, we will again use crowdsourcing to label its affordances. We believe this human-collaborating strategy is a promising way to reduce cost and scale up affordance mining in future data collection.

CHAPTER 3

Classifying Human-Object Interactions [1]

3.1 Introduction

In the last chapter we explored possible solutions for mining semantic categories of human-object interactions (HOI). In this chapter, we use the explored mining techniques and proceed to construct a large-scale dataset for still image-based HOI classification. This new dataset provides us with the opportunity to analyze state-of-the-art action classification approaches [2] on large-scale HOI recognition, and sheds light on new challenges for future research.

We first present the new dataset: "Humans Interacting with Common Objects" (HICO). HICO is featured by its scale—it has a total of 47,774 images, covering 600 categories of human-object interactions (i.e. verb-object pairs such as "riding a bike" and "eating an apple") over 117 common actions (e.g. "ride" and "feed", including a special "no interaction" class) performed on 80 common objects (e.g. "bike" and "bear").

Apart from the scale, we also highlight three key features of HICO. First, it includes *diverse interactions* for each object category—an average of 6.5 distinct interactions per object category (not including the "no interaction" categories). Second, the HOI categories are *based on senses instead of words*. That is, we do not have "repairing a bike" and "fixing a bicycle" as separate categories, as opposed to natural language based datasets such as TUHOI [115]. Third, our annotations are *multi-labeled*, motivated by the fact different interactions with the same object often co-occur, e.g. "riding a bike and holding it" and "riding a bike but not holding it (hands-free)" are both plausible. Fig. 3.1 shows example images and annotations in HICO.

The introduction of HICO enables us to evaluate and analyze state-of-the-art approaches on human-object interactions at a much larger scale. Particularly, we study the following two questions in this chapter:

[1] This chapter is based on a joint work with Zhan Wang, Yugeng He, Jiaxuan Wang, and Jia Deng [23].

[2] HOI recognition can be viewed as a subset of the larger problem of action recognition, which also includes categories involving no direct interactions with objects, such as "walking" and "jumping". In this chapter, we will use "action" and "human-object interactions" interchangeably.

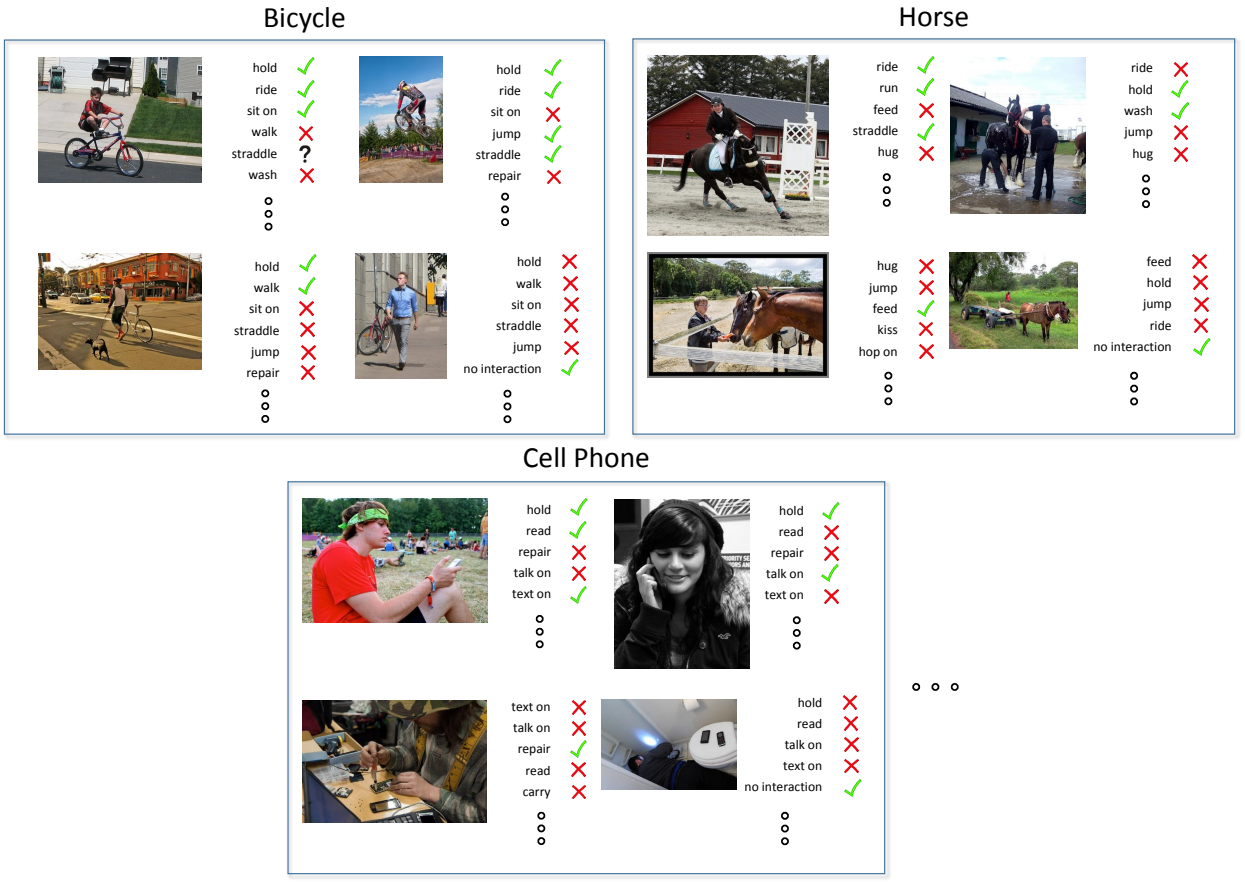


Figure 3.1: The "Humans Interacting with Common Objects" (HICO) dataset.

1. *How well do the current state-of-the-art action classification approaches perform on HICO?* Current approaches have only been tested on small datasets. It is unclear how they compare to each other on a dataset with a large number of action categories. Therefore we benchmark a number of representative action recognition approaches including deep neural network (DNN) based methods.

2. *Can semantic knowledge help recognizing uncommon human-object interactions?* As we will show later, one challenge of HOI recognition at a large scale is that the data is highly imbalanced for different interactions. For instance, "riding a bike" is observed much more frequently than "washing a bike". A key research question is how we can boost the recognition of uncommon classes. We explore possible solutions by leveraging the semantic relations between the HOI classes (e.g. "washing dishes" and "washing a bike" share the action "wash") and co-occurrence knowledge.

The contributions of this chapter are twofold: (1) we introduce a new, publicly available dataset for recognizing human-object interactions, targeting the evaluation of HOI recognition at a large scale; (2) we analyze a number of current representative approaches, which provides insights into

the challenges of large-scale HOI recognition and future research directions. The dataset and code are publicly available at http://www.umich.edu/~ywchao/hico/.

3.2 Constructing HICO

The construction of HICO consists of two steps: (1) selecting HOI categories and (2) collecting and annotating images.

3.2.1 Selecting HOI Categories

The first step of constructing a dataset for human-object interactions is to select a list of HOI categories. This involves the selection of a set of common objects and for each object, their respective common interactions. For common objects, we use the 80 object categories proposed in the MS COCO dataset [123] [3], which were carefully selected based on children's vocabularies.

Next, we need a set of common interactions for each object category. Since an established set does not exist, we leverage the techniques explored in the last chapter. We take a text-based approach by first mining the actions described in the image captions of MS COCO. Our assumption is that we are likely to obtain more "visual" actions from the descriptions in image captions than those in a generic text corpus [124]. We use the Stanford Dependency Parser [174] to extract verbs appearing in the form of "verb-noun" or "verb-preposition-noun", where the noun is a particular object name. To further expand the coverage of interactions, we also use the Google Syntactic N-grams dataset [124], which comes with dependency parsing results, and retrieve top verbs that precede the targeted object name. Since parsing is still a challenging NLP problem, we need to manually remove many incorrect results (e.g. "wear bike", "fly bike"). Combining the manually filtered results from MS COCO and Google N-grams, we obtain a set of candidate "common" verbs for each object category.

We then manually group the verbs with identical meanings into categories (e.g. "repair" is merged with "fix" in the context of "bicycle") and link them to nodes in WordNet (i.e. verb "synsets"). This is followed by a manual check that removes categories that are considered too vague or abstract (e.g. "use", "take", and "put"). The final selection is 520 verb-object pairs with 116 distinct actions (verb senses) and 80 objects. For each of the 80 object categories, we add an extra "no interaction" category, e.g. "a person is in proximity but not interacting with the bicycle". This results in a total of 600 HOI categories including the "no interaction" categories.

[3] MS COCO contains a total of 91 object categories but only 80 of them have annotations.

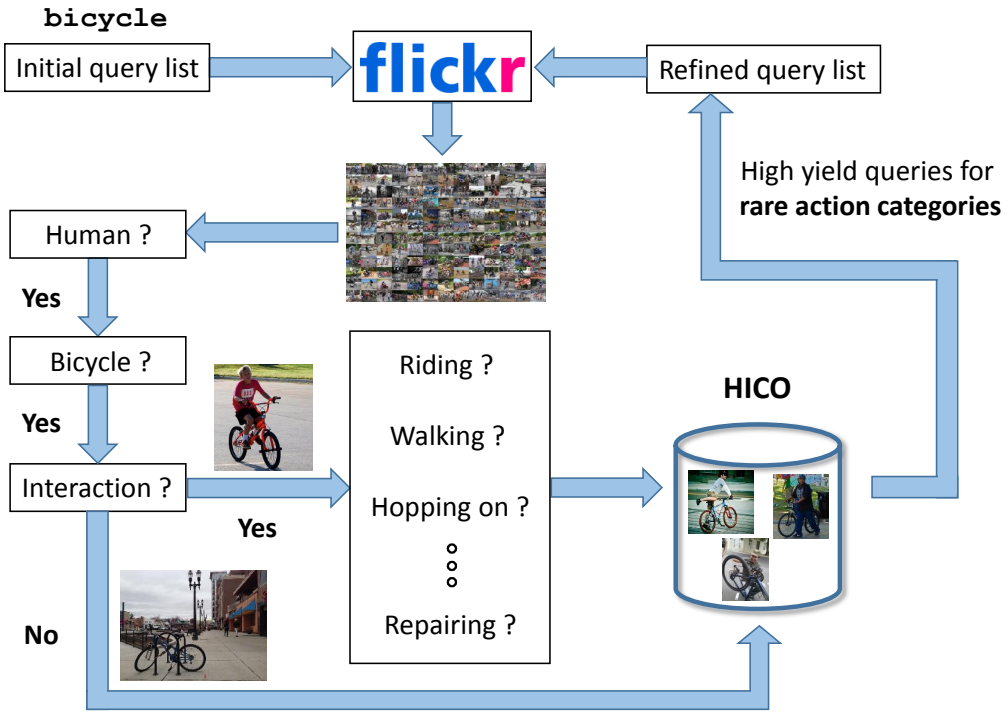


Figure 3.2: The pipeline of image collection and annotation.

3.2.2 Image Collection and Annotation

Given the selected HOI categories, we next collect and annotate images from the web. We use Creative Common images from Flickr as the source of candidate images. Fig. 3.2 illustrates our annotate pipeline. We process each of the 80 object categories independently. Given an object category (e.g. "bicycle"), we query Flickr with a group of related keywords ("bike", "fix bike", "fixing bike", "person bike", etc.).

We process each of the image retrieved from Flickr with a series of annotation task on Amazon Mechanical Turk (AMT). For example, we first verify that the candidate image contains both "person" and "bicycle". If not, the image is discarded without further processing. Next we check whether there is an interaction at all. If no, then the image is placed under the category of "person not interacting with bicycle". If yes, then we check whether each of the pre-selected interactions is presented. This completes the annotation of one candidate image. To improve annotation quality we ask up to 3 workers to answer each question and use a simple rule to combine the answers: if there is any disagreement, the final answer is marked as "ambiguous/uncertain", otherwise we use the agreed answer. The "ambiguous/uncertain" label only occurs a small fraction of time (9.77%) and is often a result of an ambiguity of interactions in images. For instance, the annotation on "straddle bicycle" for the top-left image of Fig. 3.1 is an "ambiguous/uncertain" case.

After an initial round of image collection and annotation, we found that the distribution of

	#total	#no bike/person	#no inter	#ride	#repair
iter 1	2645	1369 (52%)	65 (2%)	763 (29%)	29 (1%)
iter 2	2561	1267 (49%)	149 (6%)	694 (27%)	51 (2%)

Table 3.1: Statistics of candidate images for ”bicycle“ from Flickr in Iteration 1 and Iteration 2 (with targeted queries for rare interactions). “no bike/person” means that the image either has no person or has no bike. “no inter” means that the image has a person and a bike but there is no interaction.

images depicting different interactions for a given object is highly skewed, i.e. a few interactions dominate the candidate images (e.g. “ride”, ’hold“, ”sit on“, and ”straddle“ for ”bicycle“), while the rest of the interactions (e.g. ”walk“, ”hop on“ and ”wash“) are very hard to come by. This will create an issue for benchmarking because too few images for the long tail categories will create too big a statistical variance for evaluation. To alleviate this issue, we analyze the ”yield“ of each query keyword on the long tail classes and use those high yield keywords to perform one more round of image collection and annotation. Tab. 3.1 compare the statistics of candidate images between the second round of image collection and the initial round, which shows a notable improvement in the percentage of uncommon categories.

In the final output, each of the 80 common object categories is associated with a set of images. The images in each set are guaranteed to contain both human and the associated object category, and are annotated with ”yes“, ”no“ or occasionally ”uncertain“ exhaustively for each of the possible human interactions with this object category from a pre-selected list. Fig. 3.1 shows sample annotation of this structure. [4] Note that since each object is processed independently, a post-processing step is required to merge the duplicating images between objects. We have performed deduplication and found that around 1.09% of images are annotated with interactions with more than one object.

3.3 Details of HICO

In additional to Fig. 3.1, we show more sample images and annotations of HICO in Fig. 3.3. A complete list of the 600 HOI categories, along with the 117 actions (verb senses) and 80 objects, are shown in a matrix illustration in Fig. 3.4. Each row and column corresponds to an object and verb respectively. A blue entry marks the presence of an HOI category (i.e. verb-object pair). Fig. 3.5 shows the sorted number of positive examples for all 600 HOI categories. The long tail distribution highlights the presence of dominant and uncommon categories.

[4] In addition to the automatic pipeline, we also manually collected some images for categories with very few images.

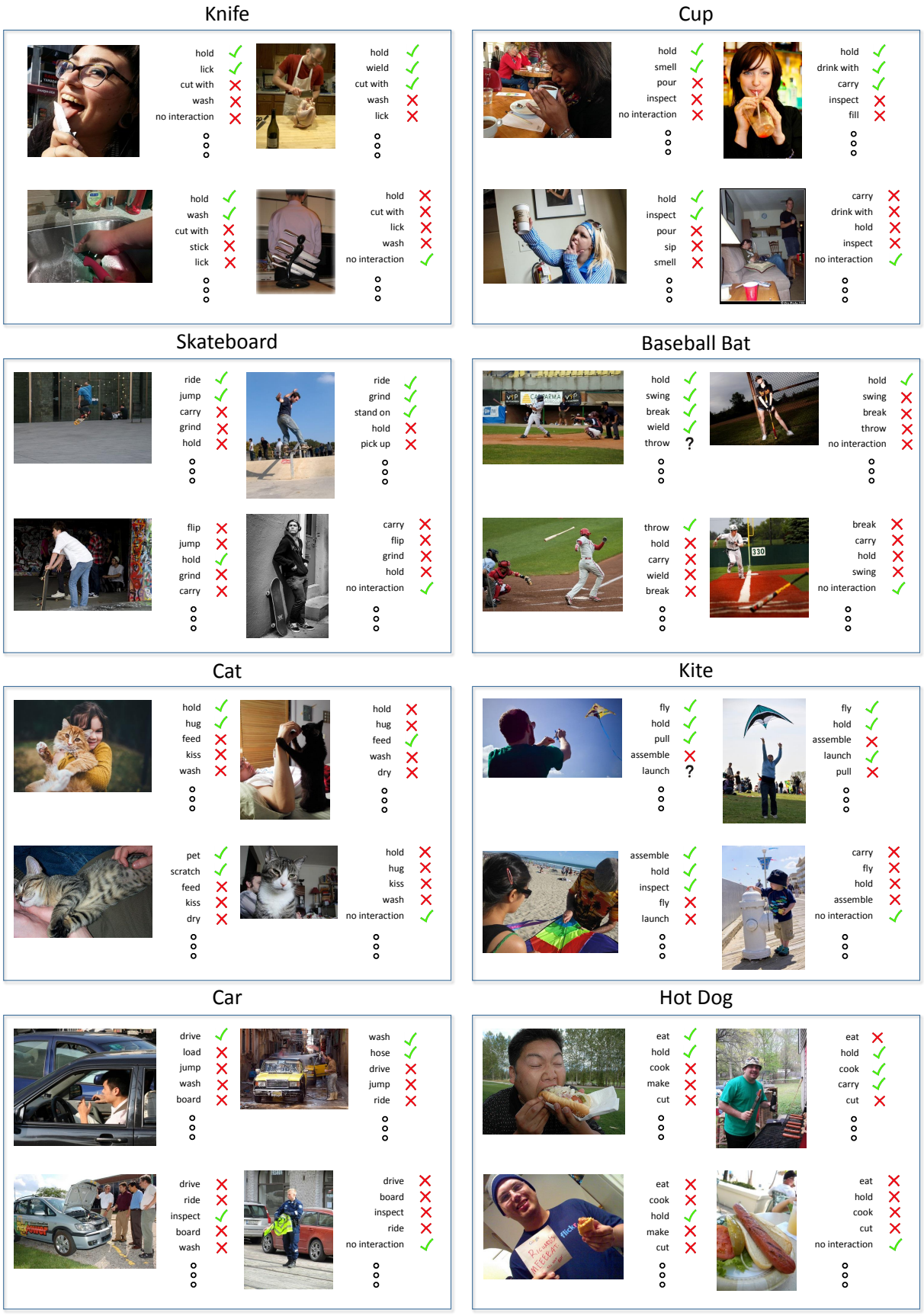


Figure 3.3: Sample images and annotations in the HICO dataset.

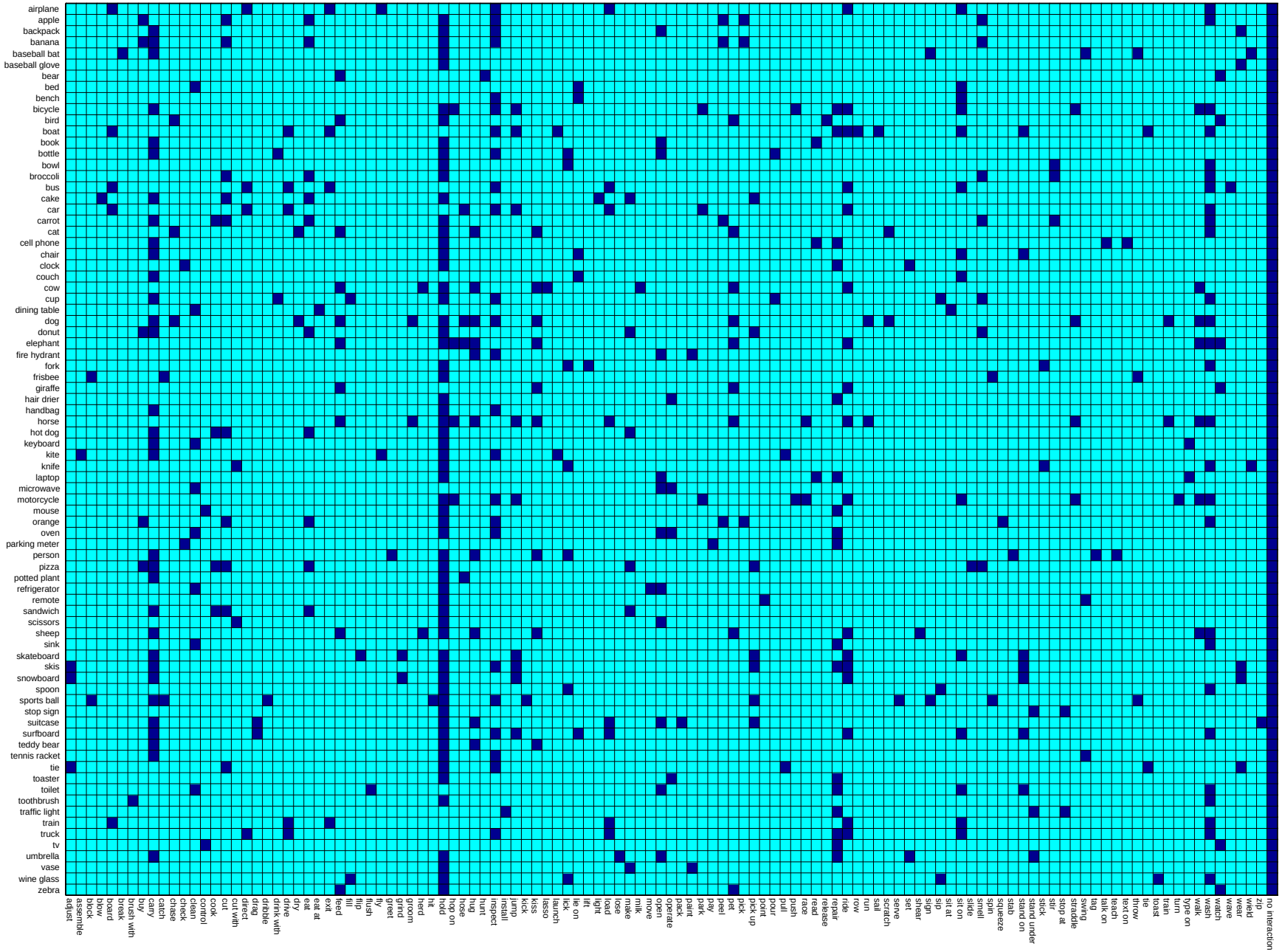


Figure 3.4: A full list of HICO's HOI categories (row: objects, column: verbs). A blue entry marks a presented HOI category (i.e. verb-object pair).

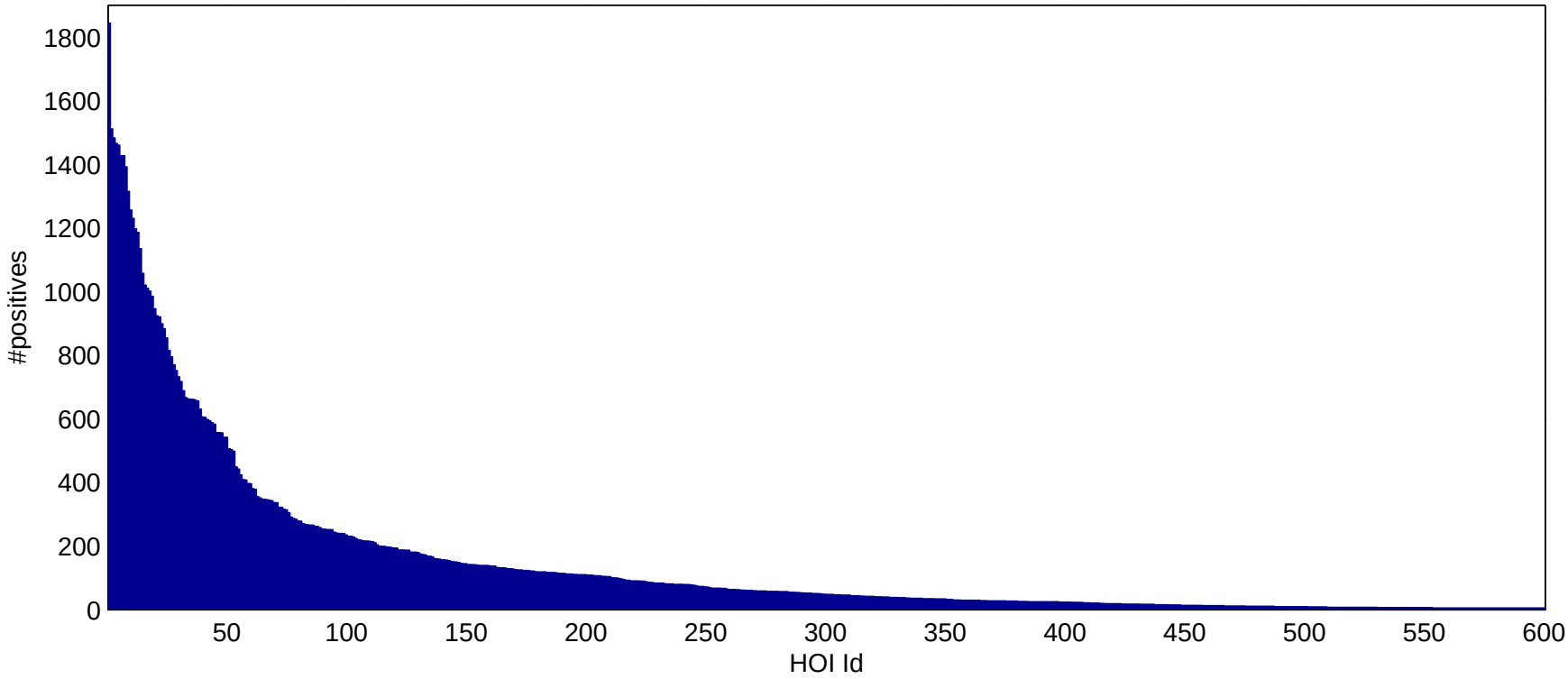


Figure 3.5: Sorted number of positive examples per HOI category. The long tail distribution highlights the presence of dominant and uncommon HOI categories.

Dataset	#images	#actions	Sense	Clean
Sports event dataset [118]	1579	8	Y	Y
Ikizler et al. [85]	467	6	Y	Y
Ikizler-Cinbis et al. [86]	1727	5	Y	Y
The sports dataset [73]	300	6	Y	Y
Pascal VOC 2010 [52]	454	9	Y	Y
Pascal VOC 2011 [52]	2424	10	Y	Y
Pascal VOC 2012 [52]	4588	10	Y	Y
PPMI [216]	4800	12	Y	Y
Willow dataset [41]	968	7	Y	Y
Stanford 40 Actions [219]	9532	40	Y	Y
TBH dataset [155]	341	3	Y	Y
HICO (ours)	**47774**	**600**	**Y**	**Y**
89 action dataset [114]	2038	89	N	Y
TUHOI [115]	10805	2974	N	Y
MPII Human Pose [8]	40522	410	Y	Y
Google Image Search [156]	102830	2938	N	N

Table 3.2: Comparison of existing image datasets on action recognition. "Sense" means whether the category list is based on senses instead of words. "Clean" means whether the dataset is human verified.

3.4 Related Datasets

Tab. 3.2 presents statistics and properties of related image datasets for action/HOI recognition. [5] Most existing work has been trained and evaluated on small-scale datasets such as PASCAL VOC Action Classification Challenge [52] and Stanford 40 Actions [219]. Our HICO dataset is one order of magnitude larger than these datasets in term of both the number of images and action categories. In the rest of this section we highlight and compare several recent efforts towards scaling up action/HOI recognition.

3.4.1 TUHOI

The "Trento Universal Human Object Interaction" (TUHOI) dataset [115] consists of 10,805 images over 2,974 actions. The images are from the ILSVRC 2013 [161] detection benchmark and are annotated with action descriptions provided by humans with no constraints on the vocabulary. At the first glance our HICO dataset and TUHOI might look similar but there are two key differences.

First, HICO uses a bounded, sense-based category list (e.g. "fix a bike" and "repair a bike" both belong to the same category), whereas TUHOI is annotated with an open, word-based category list (e.g. "fix a bike" and "repair a bike" would count as separate categories). As a result, 1,576 of the

[5] We omit video datasets here since we focus on still image-based action recognition.

verb tag	#	verb tag	#	verb tag	#	verb tag	#
bike	1	move	3	rid	2	sit on	10
cycle	7	na	1	ridden	1	stand	9
flip	1	no	1	ride	272	stand beside	1
freewheel	2	park	1	ride a bike	1	stand by	1
guide	1	pedal	5	ride on	6	stand near	1
hang	2	peddle	5	ridenon	1	steer	1
hold	21	play	3	riding	1	stop	1
hold up	1	prop	1	roll	1	touch	3
jump	2	push	5	sat	1	tricks	1
lean	1	race	4	sit	10	walk	1
look	1	repair	1	sit by	1	walk next to	1
						walk with	1

Table 3.3: Interactions with "bicycle" in the TUHOI dataset.

verb	#im	definition
carry, transport	30	move while supporting, either in a vehicle or in one's hands or on one's body
hold, take hold	1392	held by hand; to have or maintain in the grasp; to attach the hand to
inspect	124	to look at (something) carefully in order to learn more about it, to find problems, etc.
jump, leap	150	cause to jump or leap
hop on, mount, mount up, get on, jump on, climb on, bestride	26	climb up onto; get up on the back of
park	18	place temporarily
push, force	117	move with force, "He pushed the table into a corner"
repair, mend, fix, bushel, doctor, furbish up, restore, touch on	89	restore by replacing a part or putting together what is torn or broken
ride	1460	sit on and control a vehicle
sit on	1197	be seated
straddle	1511	sit or stand astride of
walk	187	to accompany on foot; to cause to move by walking
wash, rinse	6	clean with some chemical process
no interaction	174	

Table 3.4: Interactions with "bicycle" in our HICO dataset.

2,974 categories in TUHOI have only 1 associated image. Tab. 3.3 shows the interaction categories with the "bike" object in TUHOI, where "rid", "ridden", "ride", "ride a bike", "ride on", "rideon", and "riding" are counted as separate categories and most of them have only 1 image per category. In contrast, our "bike" interactions are grouped by senses and most have over 50 images (Tab. 3.4).

Second, for each object category, we exhaustively annotate all of its pre-selected interactions

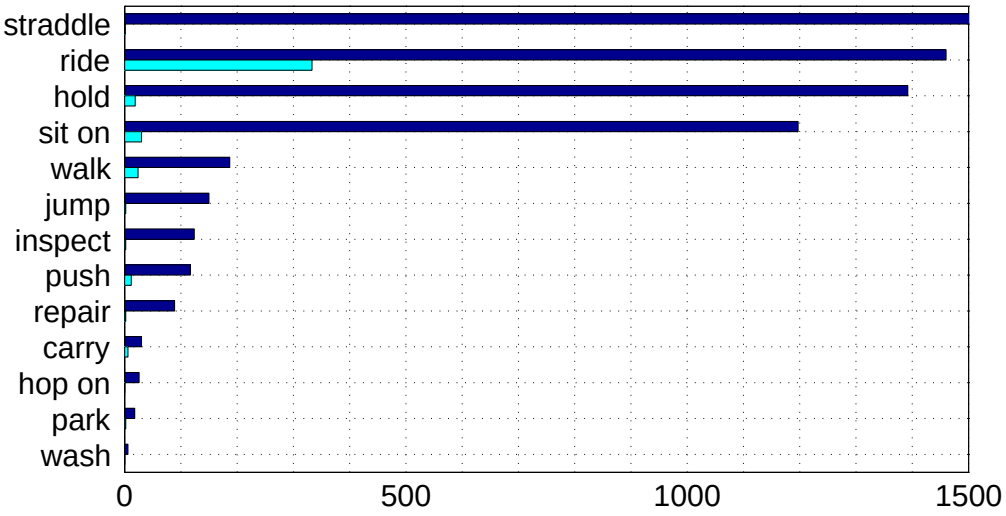


Figure 3.6: Number of images for interactions with “bicycle” (cyan: MS COCO, blue: our HICO dataset).

in each associated image, i.e. we individually verify “riding a bike”, “holding a bike”, and other “bike“ involved interactions for each image associated with ”bike“. Thus we are able to identify images of ”riding a bike but no holding it“ and retrieve accurate co-occurrence statistics of the interactions. In contrast, TUHOI does not have this exhaustive verification. That is, the absence of ”holding a bike“ does not mean that the person is in fact not holding the bike—it could simply be that the annotator did not bother to mention it. Thus the annotations of TUHOI are affected by what annotators *choose* to describe.

The above differences suggest that our HICO dataset and TUHOI are *complementary* to each other. The language annotations provided by humans in TUHOI is ideal for benchmarking the task of describing actions with natural language descriptions. Our HICO, on the other hand, insulates HOI recognition from the variations of language expressions and the elusive process of humans choosing what is worth describing.

3.4.2 MS-COCO

MS COCO [123] was originally proposed for the tasks of object detection and image captioning. Thus it has a large number of annotations of object segmentation masks and image captions. While it is not designed for action recognition, the verb phrases in the captions can in principle be extracted to provide action labels. In fact, we have done so semi-automatically in order to select our list of HOI categories. However, one main issue is that the provided image set of MS COCO does not have enough images for many long tail categories. Fig.3.6 compares the distribution of ”bicycle“ interactions between our HICO dataset and the images extracted with MS COCO captions. This suggests that without targeted collection of new images for the long tail categories, MS COCO in its current form is less suitable for benchmarking HOI recognition. Besides, as with

	#action	#HOI	#object	#action/object
MPII Human Pose [8]	410	102	66	1.55
HICO (ours)	**520**	**520**	**80**	**6.50**

Table 3.5: Comparison of action/HOI categories between MPII Human Pose [8] and our HICO dataset (excluding "no interaction" classes).

TUHOI, MS COCO has the complication of human bias in terms of what is worth describing.

3.4.3 MPII Human Pose

The MPII Human Pose dataset [8] (or MPII in short) is a large-scale benchmark for 2D human pose estimation and action recognition. It contains 40,522 images and 410 action categories. While similar in scale, the selection of action categories are driven by covering common daily activities instead of depicting different interactions with a given object category. Tab. 3.5 compares the HOI categories of MPII (those taking a verb-noun form in its definition) with our HICO dataset. Note that MPII has on average only 1.55 different interactions per object category whereas HICO has 6.5 (not including the "no interaction" categories). Thus our dataset is more suitable for HOI recognition due to the diverse interactions with the same object categories.

3.4.4 Google Image Search

Ramanathan et al. [156] report results on a large-scale dataset with 27K action categories. However, only a subset of the data—2,938 actions and 102,830 images—is released publicly. In this subset, each action category contains the top 35 images returned by Google Image Search, which are directly treated as ground truth positives without any human verification. The lack of manual cleanup makes it less authoritative as our dataset especially on benchmarking fine-grained recognition tasks such as human-object interaction.

3.5 Benchmarking Representative Approaches

In this section we benchmark a set of representative approaches for action recognition on our HICO dataset.

3.5.1 Related Work

Recognizing actions or human-object interactions in still images has gained increasing interests in computer vision. Prior work has tackled the problem with different approaches and strategies. Some work exploits information on human body poses (i.e. locations of human body parts) and

derive effective feature representations accordingly [204, 183, 85, 131, 212, 236], while some others focus on modeling the spatial relations between humans and objects [73, 217, 216, 45, 42, 155]. Some other work also explores the use of discriminative patch templates [220, 166], color [98], or exemplars [82]. Recent work by Ramanathan et al. [156] shows action recognition can also be improved by exploiting semantic relations.

3.5.2 Evaluation Setup

We adopt mean average precision (mAP) as our evaluation metric. Given an input image, the task is to output a classification score for each of the 600 HOI categories. For each HOI category, we first compute the average precision (AP) by ranking the test images by their classification scores. We then compute the average of AP over all 600 HOI categories to obtain the mAP. This evaluation metric is motivated by the fact that many HOI categories are not mutually exclusive. This is similar to the metric used by the PASCAL VOC classification competitions [52], where multiple object classes can co-occur in the same image.

To compute the AP for each HOI category, we are still yet to determine what test images to be treated as ground truth negatives. For better explanation, here we use "riding a bike" as a running example. Recall that for each HOI category, the test images can be put into four groups:

1. Verified positives: those verified to contain "riding a bike".
2. Verified negatives: those verified to contain a person and a bike but no "person riding bike".
3. Ambiguous/Uncertain images: those verified to contain a person and a bike, but with disagreements among crowd workers on whether there is "person riding bike".
4. "Unknown" images: those verified to contain a person and *another object class*, e.g. "cat".

One setting it to use the verified positives as positives, skip the ambiguous images and the "Unknown" images, and use the verified negatives as negatives. This setting assumes that we will be able to perfectly filter out images with no "bicycles" before trying to recognize the interactions. We refer to this setting as the "Known Object" (KO) setting.

The "Known Object" setting might be "too easy" and unrealistic in the sense that a recognition algorithm will not be distracted by images *not* containing the correct object category. A more realistic setting is to include images without "bicycles" but with some other objects as part of the negative set, since in most cases we are unable to pre-filter the images by their object classes robustly. We approximate this setting by treating the "Unknown" images as extra ground-truth negatives. Although there is a chance that some "Unknown" images may actually contain "person riding bike" (thus corrupting the evaluation), we have checked that the risk is small enough

to be acceptable: we randomly sampled 10 HOI categories, manually went through all of their "Unknown" images, and found only 0.3% of them are positives. In this section, unless otherwise noted, all evaluations default to this more realistic setting.

We split the dataset using a random 80-20 training-test split, with an additional constraint that each HOI category must have at least 5 test images. Note that in our dataset all HOI categories have at least 6 images, so all categories have at least 1 training image. The subset of categories (51 out of 600) with only 1 training images provides an one-shot learning scenario, which is a natural result of the long tail distribution of HOI categories and is a challenge that any practical HOI recognition approach needs to address.

3.5.3 Representative Approaches

We benchmark the following four approaches on HICO:

1. **RandomForest (RF) [220]:** This is the winning approach for both the PASCAL VOC 2011 and PASCAL VOC 2012 Action Classification competitions [52]. It uses random forests to select discriminative image regions, represented using SIFT features. We include it here to represent the state of the art of action/HOI recognition in static images.

2. **FisherVectors (FV) [164]:** Prior to the recent breakthrough made by deep neural networks [108], Fisher Vectors based approaches were the state of the art on image classification and won the ILSVRC2011 competition [161]. We include it here to represent traditional image classification approaches. Given the Fisher Vectors, we train a binary SVM for each HOI category.

3. **DNN:** We extract features from the fc_7 layer of AlexNet [108] pre-trained on ImageNet, and learn one binary SVM for each HOI category. This represents the current state-of-the-art approach on image classification. We also evaluate the following three variants with different fine-tuning strategies:

   (a) *Fine-tune V*: Fine-tune AlexNet to classify only verb categories (i.e. a "wash" category obtained by grouping "wash bike", "repair bike", "wash cat", etc. all into one category). This is to learn features that are common to a particular action such as "wash" regardless of the objects.

   (b) *Fine-tune O*: Fine-tune AlexNet to classify only object categories (i.e. a "bike" category obtained by grouping "wash bike", "repair bike", "ride bike", etc. all into one category). This is to learn features that are common to a particular object such as "bike" regardless of the actions.

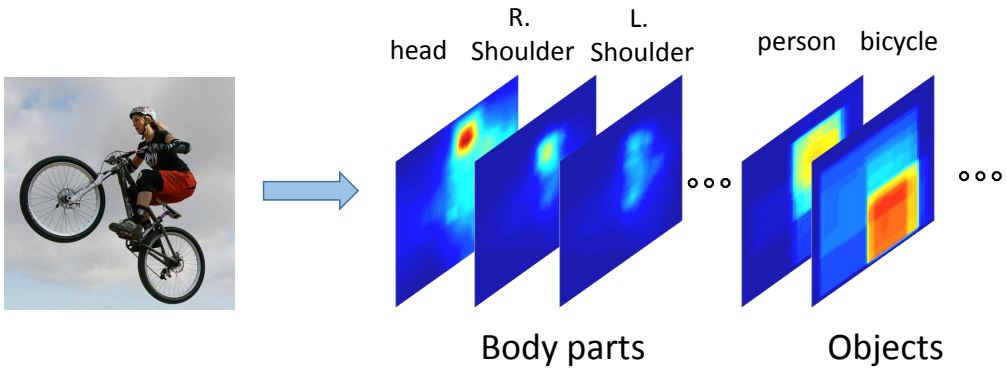


Figure 3.7: Stacked heatmaps of human pose estimation and object detection as input to HOCNN.

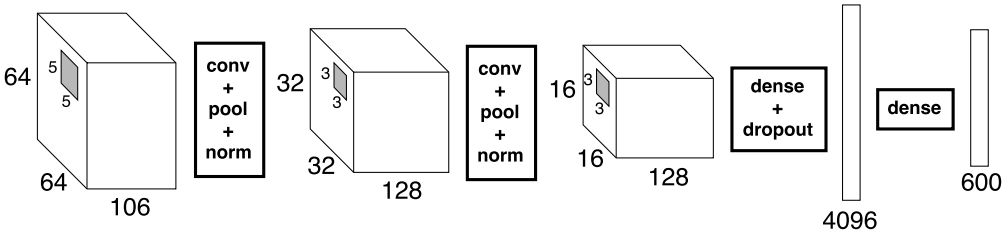


Figure 3.8: Network architecture of HOCNN.

(c) *Fine-tune VO*: Fine-tune AlexNet to directly classify verb-object pairs, i.e. HOI categories.

4. **HOCNN (Human-Object CNN):** In this approach, we use the outputs of object detection and human pose estimation as features, on top of which we learn a convolutional neural network (CNN) to classify the HOI categories. Specifically, given an input image, we first run object detectors and a human pose estimator, which together generate a set of heatmaps, one per object category and one per body part. A total of 106 heatmaps (80 object categories plus 26 body parts) are stacked together as the input to a CNN (Fig. 3.7 and Fig. 3.8).

   We include this approach to shed light on the role of the spatial relations between humans and objects for HOI recognition. Human-object spatial relation has been widely exploited in prior work to design discriminative feature representations [73, 217, 216, 45, 42, 155]. Here we study to what extent such a mid-level representation can help in large-scale HOI recognition, especially with a more powerful learning tool such as CNN.

   To generate object heatmaps, we first take off-the-shelf R-CNN object detectors [66] for the 20 PASCAL VOC object categories (a subset of 80 MS COCO object categories). For the remaining 60 object categories, we train 60 R-CNN detectors using the training set of MS COCO. [6] We use the network of AlexNet pre-trained on ILSVRC 2012 without fine-tuning.

[6] We found 1,776 images (out of 47,774) in HICO are also in MS COCO. These duplicated images are put into the

	mAP	mAP (KO)
Random	0.57	33.37
RF [220]	7.30	38.15
FV [164]	4.21	37.74
DNN (ImageNet)	18.58	48.22
DNN (fine-tune V)	17.65	**49.07**
DNN (fine-tune O)	**19.38**	47.42
DNN (fine-tune VO)	18.08	47.89
HOCNN	4.90	39.05

Table 3.6: Performance of representative approaches.

All the features are obtained from the output of the fc_7 layer. To validate our trained models, we evaluate the 60 object detectors with the MS COCO validation set. Tab. 3.8 reports the detection average precision (AP) for the 60 object classes. We see that the AP is generally higher for larger objects such as elephants and trucks, while lower for smaller objects such as forks and remotes. More than half of the object classes have AP below 20%, showing the challenge of detecting objects and a potential limitation of using object detection as a mid-level representation for HOI recognition.

To generate body part heatmaps, we use the pre-trained human pose estimator developed by Chen et al. [31].

3.5.3.1 Results

Tab. 3.6 reports the mAP of the benchmarked approaches in both the default and "Known Object" settings. In the default setting, DNN based approaches overwhelmingly outperforms traditional approaches (RandomForest and FisherVectors). Although the ordering is not surprising given the recent success of DNNs, the large gap (18.58 mAP versus 7.30 and 4.21 mAP) is still astonishing. Unsurprisingly, RandomForest (RF) outperforms FisherVectors (FV), as the former was specifically designed for action recognition. Among the fine-tuned DNN variants, fine-tuning for objects (fine-tune O) achieves the highest mAP in the default setting, suggesting that a major source of error is on recognizing the objects.

In the "Known Object" setting, however, fine-tuning for verbs (fine-tune V), instead of objects, achieves the highest mAP. This agrees with the fact that object recognition is no longer a source of error in this setting. More importantly, in this setting, all methods are not that much better than random chance, suggesting that the core problem of HOI recognition—*recognizing the interactions*—is still largely unsolved, even with DNNs.

training split of HICO to ensure that test images of HICO are not used in any training.

HOCNN, while not performing as well as RandomForest (RF) in the default setting, outperforms both RandomForest (RF) and FisherVectors (FV) in the "Known Object" setting. This validates the importance of the use of human-object spatial relations in distinguishing different interactions, and the high mAP of RF in the default setting should be attributed to better object recognition. Note that in both setting, there is still a large gap between HOCNN and the DNN based methods, which is able to extract information from input pixels. This is possibly because that end-to-end DNNs have the flexibility to select and combine all type of cues, from low level to high level and from local to global, whereas HOCNN is restricted to spatial relations between human and objects.

Fig. 3.9 shows the top ranked images returned by DNN, FV, and HOCNN for a few HOI categories in the default setting. These examples show that all the approaches perform better for HOI categories involving salient objects (e.g. "sailing a boat", "jumping a horse"). AP drops significantly when the objects are smaller and harder to detection (e.g. "talking on a cellphone"). Besides, even when the objects can be reliably detected, distinguishing the interactions can still be very challenging. For example, DNN (ImageNet), the dominating approach, still fail to distinguish "kissing a giraffe" from "feeding a giraffe".

3.5.4 Using Semantic Knowledge

As shown earlier in Fig. 3.5, a major challenge of HOI recognition is on the long-tail categories that have very few training images. How can we improve the recognition outcome on these uncommon/rare categories? Here we explore two possible solutions by leveraging *semantic knowledge* (i.e. knowledge about the semantic relations between HOI categories): (1) knowledge on compositions and (2) knowledge on co-occurrences.

3.5.4.1 Knowledge on Compositions

Humans can recognize uncommon HOIs with ease even if they have not seen many examples in the past, e.g. even if we have not seen "washing a bike", we can still reliably recognize the interaction. This is likely because we have understood the concept of "washing", regardless of the object being washed, from observing common interactions with people washing something such as "washing dishes". In our HOI dataset, there is a significant sharing of actions (verb senses) between HOI categories—on average, each action (verb sense) is paired with 4.5 different object categories. This provides an opportunity to test the hypothesis that such knowledge of compositions can be used to improve the recognition of rare HOI categories.

We experiment with the simplest possible strategy. Using the training data of HICO, we first train three types of *basic* classifiers: those for classifying verb-object (VO) pairs, those for classi-

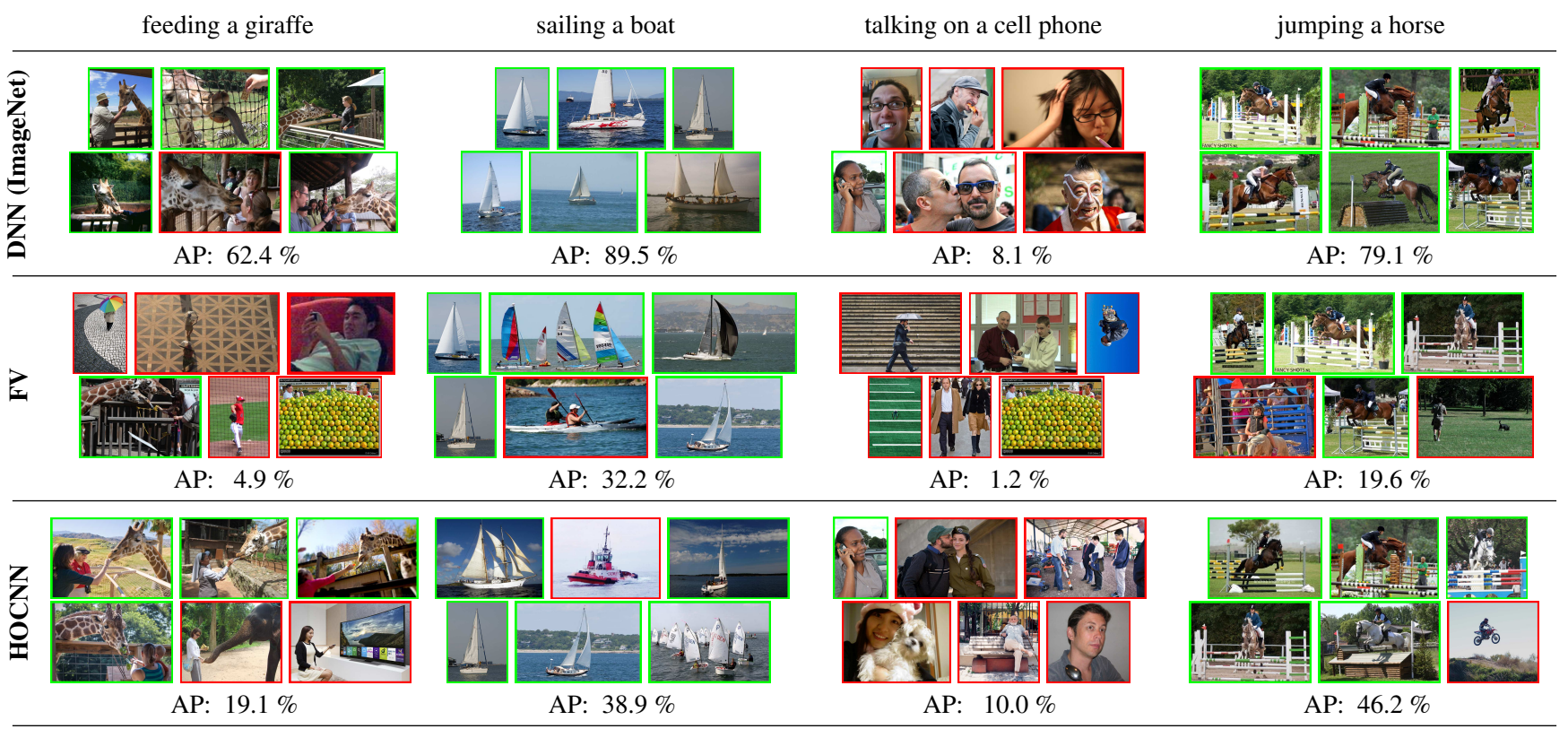


Figure 3.9: Top ranked images for different HOI categories in the default setting. Each row (column) represents one representative approach (HOI category). Green and red boxes represent ground-truth positives and negatives respectively.

fying verbs (V) only, and those for classifying objects (O) only. This is similar to the fine-tuning of DNNs in Sec. 3.5.3. Then we train *combined* classifiers by exploring various combinations of these three types of basic classifiers. For example, for "feeding a zebra", we can learn a V+O classifier by combining the outputs of a V classifier trained to recognize "feeding" regardless of the objects and an O classifier trained to recognize "zebra" regardless of the verbs.

We detail the training of basic and combined classifiers as follows:

- Basic Classifiers: We train three types of basic classifiers: verb-object (VO) pairs, verbs (V), and objects (O), resulting in 600, 117, and 80 classifiers respectively. Each classifier is trained using a linear SVM [53]. We perform 5-fold cross-validation to determine the hyperparamter C.

- Combined Classifiers: For each HOI category, we train combined classifiers by linearly combining the output scores of a set of basic classifiers:

$$\overline{\phi}_j = \sum_{i \in S} w_i \phi_{ij}, \tag{3.1}$$

  where ϕ_{ij} denotes the output score of a basic classifier i on an image sample j, and w_i denotes the learned weight. The set S is determined by the choice of basic classifiers in use. For example, we take $S = \{\text{V}, \text{O}, \text{VO}\}$ for training a V+O+VO classifier. The weights w_i are learned by maximizing the training AP using a grid search over $[0, 1]^{|S|}$. Due to the reuse of training data, directly using the output scores from the trained basic classifiers will lead to over-fitting. Instead, we set ϕ_{ij} based on the output scores from cross-validation, i.e. ϕ_{ij} is obtained from the model trained on all splits not containing the sample j.

3.5.4.2 Knowledge on Co-occurrences

We also explore another type of semantic knowledge, namely the co-occurrences of actions. The intuition is that the prediction of a rare category might be helped from a co-occurring interaction that has more training data and easier to recognize. Our HICO dataset provides an ideal setting to test this hypothesis because co-occurring interactions are exhaustively annotated. For instance, Fig. 3.10 shows the co-occurrences of interactions for some object categories in HICO.

Again we evaluate the simplest possible method: we learn a *combined* classifier (in a similar manner for the knowledge on compositions) to combine the outputs of all VO classifiers for interactions that might co-occur on the same object category. For example, suppose we have trained the VO classifiers for "eating a hot dog", "holding a hot dog", "cutting a hot dog", and "riding a bike". Then we learn a new classifier for "eating a hot dog" by combining the outputs of the original two VO classifiers for "eating a hot dog" and "holding a hot dog"—"cutting a hot dog" is not used

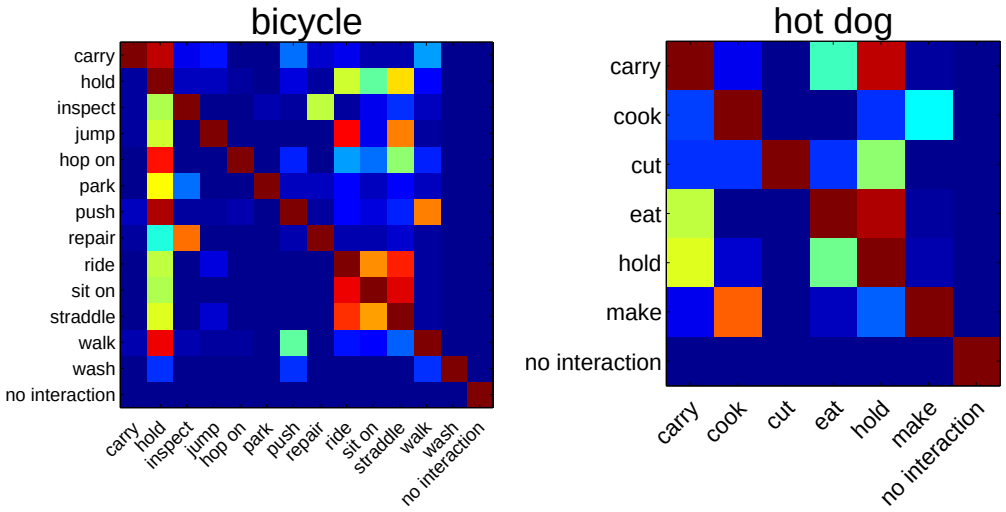


Figure 3.10: Co-occurrences of interactions.

because the interaction rarely co-occurs with "eating a hot dog", and "riding a bike" is not used because the interaction is not on the same object.

To measure the level of co-occurrence of two HOI classes, we adopt the normalized co-occurrence measure used in [135]. For an HOI class i, the normalized co-occurrence of an HOI class j is defined by:

$$s_{ij} = \frac{c_{ij}}{c_i}, \tag{3.2}$$

where c_{ij} is the number of images that are labeled positive for both class i and j, and c_i is the number of images that are labeled positive for class i. To apply the co-occurrence knowledge, we first compute the normalized co-occurrences of HOI classes for each object category separately using the training annotations. We set an HOI class j as a co-occurring class of HOI class i if $s_{ij} > 0.5$ and $i \neq j$. To train combined classifiers, we include only the basic classifiers of co-occurring HOI classes and ignore the non-co-occurring ones.

It is important to note that in all of our experiments no test images are ever used to learn any new classifiers that combine the outputs of existing classifiers. All learning is done using only the training set of HICO and cross-validation is used to prevent overfitting.

3.5.4.3 Results

Tab. 3.7 (top) reports the results in the default setting. The "F" columns report the mAP on the full list of 600 HOI categories, while the "R" columns on 167 rare categories—those with less than 5 positive training examples. We highlight a few observations: First, regardless of the approaches, adding the V classifiers leads to moderate but consistent improvement for overall mAP as well as mAP for rare classes (e.g. for DNN(ImageNet), from 18.58 to 18.64 on the full dataset and

	VO		V+O		V+VO		O+VO		V+O+VO		VO+coocc		V+O+VO+coocc	
	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Random	0.57	0.18	0.57	0.18	0.57	0.18	0.57	0.18	0.57	0.18	0.57	0.18	0.57	0.18
DNN (ImageNet)	18.58	**0.74**	18.42	5.04	18.64	**1.73**	20.95	**5.07**	20.71	4.98	20.13	4.18	21.06	5.72
DNN (fine-tune V)	17.65	0.29	17.47	4.75	16.94	1.42	19.41	4.67	18.86	3.96	18.76	3.64	19.26	5.45
DNN (fine-tune O)	**19.38**	0.39	**18.68**	**5.99**	**19.43**	1.31	**21.52**	4.70	**21.33**	**5.07**	**20.91**	**4.31**	**21.66**	**5.91**
DNN (fine-tune VO)	18.08	0.39	17.44	4.79	18.41	1.68	19.36	4.40	19.38	4.51	18.98	3.94	19.62	5.35
HOCNN	4.90	0.16	5.40	0.51	5.09	0.21	5.38	0.32	5.47	0.32	5.18	0.32	5.51	0.41

	VO		V+O		V+VO		O+VO		V+O+VO		VO+coocc		V+O+VO+coocc	
	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Random	33.37	19.26	33.37	19.26	33.37	19.26	33.37	**19.26**	33.37	19.26	33.37	19.26	33.37	19.26
DNN (ImageNet)	48.22	**23.41**	49.40	30.76	51.52	32.66	46.59	16.94	49.15	20.61	47.97	22.32	49.11	20.04
DNN (fine-tune V)	**49.07**	22.80	**49.98**	31.81	**51.56**	**33.86**	**48.04**	17.58	**49.58**	**25.52**	**49.08**	**22.85**	**49.31**	**23.80**
DNN (fine-tune O)	47.42	22.12	47.97	27.37	50.68	32.00	45.83	16.58	47.87	19.23	47.14	22.08	47.65	18.23
DNN (fine-tune VO)	47.89	22.17	49.87	**32.46**	50.51	32.78	46.81	17.02	48.30	20.99	47.76	21.79	48.17	20.30
HOCNN	39.05	20.81	39.74	21.79	40.74	23.09	38.15	18.12	39.80	19.51	38.81	20.29	39.72	19.57

Table 3.7: Performance of different combinations of V, O, and VO classifiers on different approaches. Top: default setting, Bottom: "Known Object" setting. Performance measured as mAP (%) on all 600 HOI classes (F) and 167 rare classes (R)—those with less than 5 positive training examples.

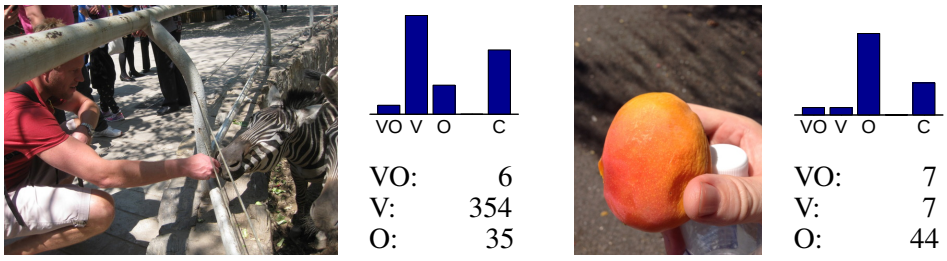
truck	tf light	hydrant	sp sign	pk meter	bench	elephant	bear	zebra	giraffe	backpack	umbrella	handbag	tie	suitcase
22.9	13.2	56.0	61.4	23.2	10.8	48.7	50.3	56.9	56.7	9.7	19.1	1.9	16.1	13.4

frisbee	skis	snowbd	sp ball	kite	bb bat	bb glove	skatebd	surfbd	racket	wn glass	cup	fork	knife	spoon
23.7	11.6	11.2	17.2	14.7	13.4	18.4	19.6	14.6	28.2	13.3	15.0	9.4	9.2	9.6

bowl	banana	apple	sandwich	orange	broccoli	carrot	hot dog	pizza	donut	cake	bed	toilet	laptop	mouse
23.4	14.6	13.1	24.6	20.3	16.8	9.4	24.4	41.8	17.5	11.5	27.4	39.4	35.1	23.6

remote	keyboard	phone	microwave	oven	toaster	sink	fridg	book	clock	vase	scissors	td bear	hr drier	tbrush
8.1	21.5	13.4	27.3	17.5	11.1	15.1	24.4	1.0	48.0	17.6	15.8	36.3	0.3	0.6

Table 3.8: Object detection average precision (%) of the 60 non-PASCAL VOC object detectors on the MS-COCO validation set.



(a) feeding a zebra (b) squeezing an orange

Figure 3.11: Improving HOI recognition with knowledge of compositions. Left: input image. Top-right: prediction scores of the VO classifier, V classifier, O classifier, and the combined (C) classifier. Bottom-right: number of training examples for the VO, V, and O classifiers.

from 0.74 to 1.73 on rare categories). Second, the biggest improvements come from adding the O classifiers, especially on the rare classes. Fig. 3.11 presents the predictions of different types of classifiers on two example images, illustrating how V classifier and O classifier help when a VO classifier is trained with very few images. Third, adding co-occurrence knowledge consistently improves performance. Finally, the best result is achieved by combining the compositional knowledge and the co-occurrence knowledge (V+O+VO+coocc).

Tab. 3.7 (bottom) reports the results for the same set of approaches in the "Known Object" (KO) setting. This setting assumes perfect object recognition and focuses the evaluation on recognizing the interactions. In contrast to the default setting, we see that adding O classifiers to any combination significantly hurts performance, which is expected given that the objects are already recognized and adding O classifiers will only cause overfitting. On the contrary, adding V classifiers leads to much more pronounced improvements (e.g. on rare categories, a 9.25% absolute increase in mAP from VO to V+VO for DNN (ImageNet)). This highlights the promise of leveraging semantic knowledge for large-scale HOI recognition.

3.6 Summary

We introduce a new benchmark—"Humans Interacting with Common Objects" (HICO)—for recognizing human-object interactions (HOI). Our dataset possesses three key features: a diverse set of interactions with common object categories, a list of well-defined sense-based HOI categories, and an exhaustive labeling of co-occurring interactions with an object category in each image. We analyze a number of current representative approaches, and show that semantic knowledge can significantly improve HOI recognition, especially for uncommon categories.

CHAPTER 4

Detecting Human-Object Interactions [1]

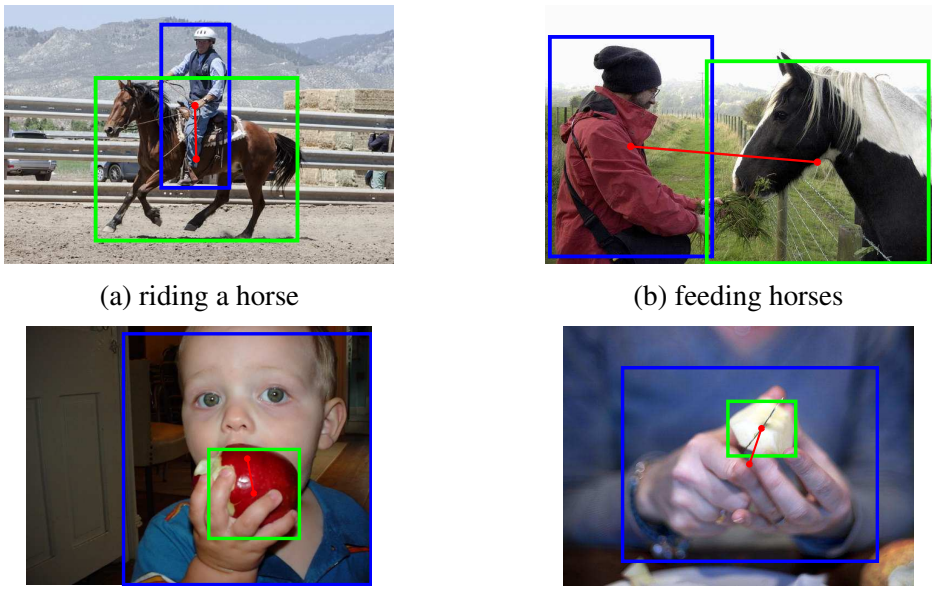
4.1 Introduction

In the last chapter we addressed the understanding of human-object interactions (HOI) by formulating it as an image-level classification problem, i.e. given an input image, what are the observed HOI categories? In this chapter, we take one step further and formulate HOI recognition as a detection problem. Specifically, we introduce two additional questions: *Where are the people and objects in the image? And which person is interacting with which object?* We construct a new benchmark by augmenting the HICO dataset introduced in the last chapter and propose a novel deep neural network (DNN) based approach to address the problem.

Visual recognition of human-object interactions (e.g. riding a horse, eating a sandwich) has recently attracted increasing attention in the field of computer vision [73, 217, 216, 45, 131, 42, 44, 155, 82]. Despite the significant progress made by recent efforts, HOI recognition is still far from being solved. One limitation is that previous approaches have only been tested on small datasets with few action/HOI categories, e.g. 10 categories in PASCAL VOC [52] and 40 categories in Stanford 40 Actions [219]. Furthermore, these benchmark datasets offer a very limited number of interaction classes for each object category. For example, in Stanford 40 Actions, "repairing a car" is the only HOI category involving the object "car". It is unclear whether a successful algorithm really recognizes the interactions (e.g. "repairing"), or whether it is simply recognizing the presented objects (e.g. "car"). This limitation has recently been addressed by Chao et al. [23] (work presented in the last chapter), who introduced "Humans interacting with Common Objects" (HICO), a large image dataset containing 600 HOI categories over 80 common object categories and featuring a diverse set of interactions for each object category. Given the HICO dataset, Chao et al. [23] provide the first benchmark for image-based HOI classification at a large scale.

While the introduction of HICO may facilitate progress in the study of HOI classification, HOI recognition still cannot be fully addressed, since with only HOI classification a vision system is

[1] This chapter is based on a joint work with Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng [21].



(a) riding a horse (b) feeding horses

(c) eating an apple (d) cutting an apple

Figure 4.1: Detecting human-object interactions. Blue boxes mark the humans. Green boxes mark the objects. Each red line links a person and an object involved in the labeled HOI class.

not able to accurately localize the presented interactions in images. To be able to ground HOIs to image regions, this chapter propose studying a new problem: detecting human-object interactions in static images. The goal of HOI detection is not only to determine the presence of HOIs, but also to estimate their locations. Formally, we define the problem of HOI detection as predicting a pair of bounding boxes—first for a person and second for an object—and identifying the interaction class, as illustrated in Fig. 4.1. This is different from conventional object detection, where the output is only a single bounding box with a class label. Addressing HOI detection will bridge the gap between HOI classification and object detection by identifying the interaction relations between detected objects.

The contributions of this chapter are two-fold: (1) We introduce HICO-DET, the first large benchmark for HOI detection, by augmenting the current HICO classification benchmark with instance annotations. HICO-DET offers more than 150K annotated instances of human-object pairs, spanning the 600 HOI categories in HICO, i.e. an average of 250 instances per HOI category. (2) We propose Human-Object Region-based Convolutional Neural Networks (HO-RCNN), a DNN-based framework that extends state-of-the-art region-based object detectors [66, 65, 158] from detecting a single bounding box to a pair of bounding boxes. At the core of our HO-RCNN is the Interaction Pattern, a novel DNN input that characterizes the spatial relations between two bounding boxes. Experiments on HICO-DET demonstrate that our HO-RCNN, by exploiting human-object spatial relations through Interaction Patterns, significantly improves the performance of HOI detection over baseline approaches. The dataset and code are publicly available at

http://www.umich.edu/~ywchao/hico/.

4.2 Related Work

4.2.1 HOI Recognition

A surge of work on HOI recognition has been published since 2009. Results produced in these works were evaluated on either action classification [217, 216, 45, 131, 42, 44, 155, 82], object detection [217], or human pose estimation [217, 44]; none of them were directly evaluated on HOI detection. Chao et al. [23] recently contributed a large image dataset "HICO" for HOI classification [23, 132]. However, HICO does not provide ground-truth annotations for evaluating HOI detection, which motivates us to construct a new benchmark by augmenting HICO. We also highlight a few other recent datasets. Gupta and Malik [75] augmented MS COCO [123] by connecting interacting people and objects and labeling their semantic roles. Yatskar et al. [231] contributed an image dataset for situation recognition, defined as identifying the activity together with the participating objects and their roles. Both datasets, unlike HICO, do not offer a diverse set of interaction classes for each object category. Lu et al. [129] and Krishna et al. [106] separately introduced two image datasets for detecting object relationships. While they feature a diverse set of relationships, the relationships are not exhaustively labeled in each image. As a result, follow-up works [34, 121, 120, 232, 233] which benchmark on these datasets can only evaluate their detection result with recall, but not precision. In contrast, we exhaustively labeled all the instances for each positive HOI label in each image, enabling us to evaluate our result with mean Average Precision (mAP).

4.2.2 Object Detection

Standard object detectors [65, 158, 128, 35] only produce a class-specific bounding box around each object instance; they do not label the interaction among objects. Sadeghi and Farhadi [163] proposed "visual phrases" by treating each pair of interacting objects as a unit and leveraged object detectors to localize them. HOI detection further extends the detection of "visual phrases" to localize individual objects in each pair. Our proposed HO-RCNN, built on recent advances in object detection, extends region-based object detectors [66, 65, 158] from taking single bounding boxes to taking bounding box pairs.

4.2.3 Grounding Text Descriptions to Images

HOI detection grounds the semantics of subjects, objects, and interactions to image regions, which is relevant to recent work on grounding text descriptions to images. Given an image and

its caption, Kong et al. [102] and Plummer et al. [154] focus on localizing the mentioned entities (e.g. nouns and pronouns) in the image. HOI detection, besides grounding entities, i.e. people and objects, also grounds interactions to image regions. Karpathy and Fei-Fei [95] and Johnson et al. [91] address region-based captioning, which can be used to generate HOI descriptions in image regions. However, they are unable to localize individual persons and objects involved in the HOIs.

4.3 HO-RCNN

Our HO-RCNN detects HOIs in two steps. First, we generate proposals of human-object region pairs using state-of-the-art human and object detectors. Second, each human-object proposal is passed into a ConvNet to generate HOI classification scores. Our network adopts a multi-stream architecture to extract features on the detected humans, objects, and human-object spatial relations.

4.3.1 Human-Object Proposals

We first generate proposals of human-object region pairs. One naive way is to exploit a pool of class-agnostic bounding boxes like other region-based object detection approaches [66, 65, 158]. However, since each proposal is a pairing between a human and object bounding box, the number of proposals will be quadratic in the number of the candidate bounding boxes. To ensure high recall, one usually needs to keep hundreds to thousands of candidate bounding boxes, which results in more than tens of thousands of human-object proposals. Classifying HOIs for all proposals will be intractable. Instead, we assume having a list of HOI categories of interest (e.g. "riding a horse", "eating an apple") beforehand, so we can first detect bounding boxes for humans and the object categories of interest (e.g. "horse", "apple") using state-of-the-art object detectors. We keep the bounding boxes with top detection scores. For each HOI category (e.g. "riding a horse"), the proposals are then generated by pairing the detected humans and the detected objects of interest (e.g. "horse") as illustrated in Fig. 4.2.

4.3.2 Multi-stream Architecture

Given a human-object proposal, our HO-RCNN classifies its HOIs using a multi-stream network (Fig. 4.3), where different streams extract features from different sources. To illustrate our idea, consider the classification of one HOI class "riding a bike". Intuitively, local information around humans and objects, such as human body poses and object local contexts, are critical in distinguishing HOIs: A person riding a bike is more likely to be in a sitting pose rather than standing; a bike being ridden by a person is more likely to be occluded by the person's body in the upper region than those not being ridden. In addition, human-object spatial relations are also important

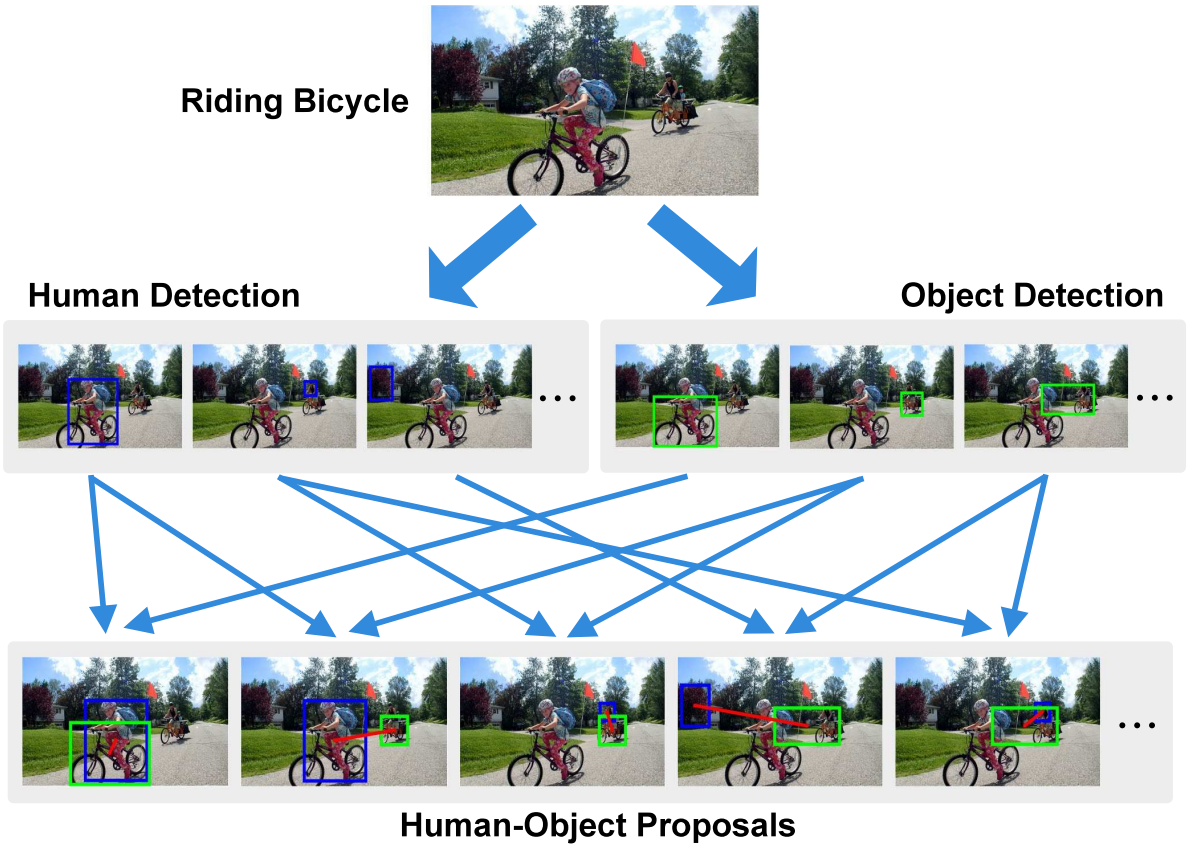


Figure 4.2: Generating human-object proposals from human and object detections.

cues: The position of a person is typically at the middle top of a bicycle when he is riding it. Our multi-stream architecture is composed of three streams which encode the above intuitions: (1) The *human stream* extracts local features from the detected humans. (2) The *object stream* extracts local features from the detected objects. (3) The *pairwise stream* extracts features which encode pairwise spatial relations between the detected human and object. The last layer of each stream is a binary classifier that outputs a confidence score for the HOI "riding a bike". The final confidence score is obtained by summing the scores over all streams. To extend to mulitple HOI classes, we train one binary classifier for each HOI class at the last layer of each stream. The final score is summed over all streams separately for each HOI class.

4.3.3 Human and Object Stream

Given a human-object proposal, the human stream extracts local features from the human bounding box, and generates confidence scores for each HOI class. The full image is first cropped using the bounding box and resized to a fixed size. This normalized image patch is then passed into a ConvNet that extracts features through a seires of convolutional, max pooling, and fully-connected layers. The last layer is a fully-connected layer of size K, where K is the number of

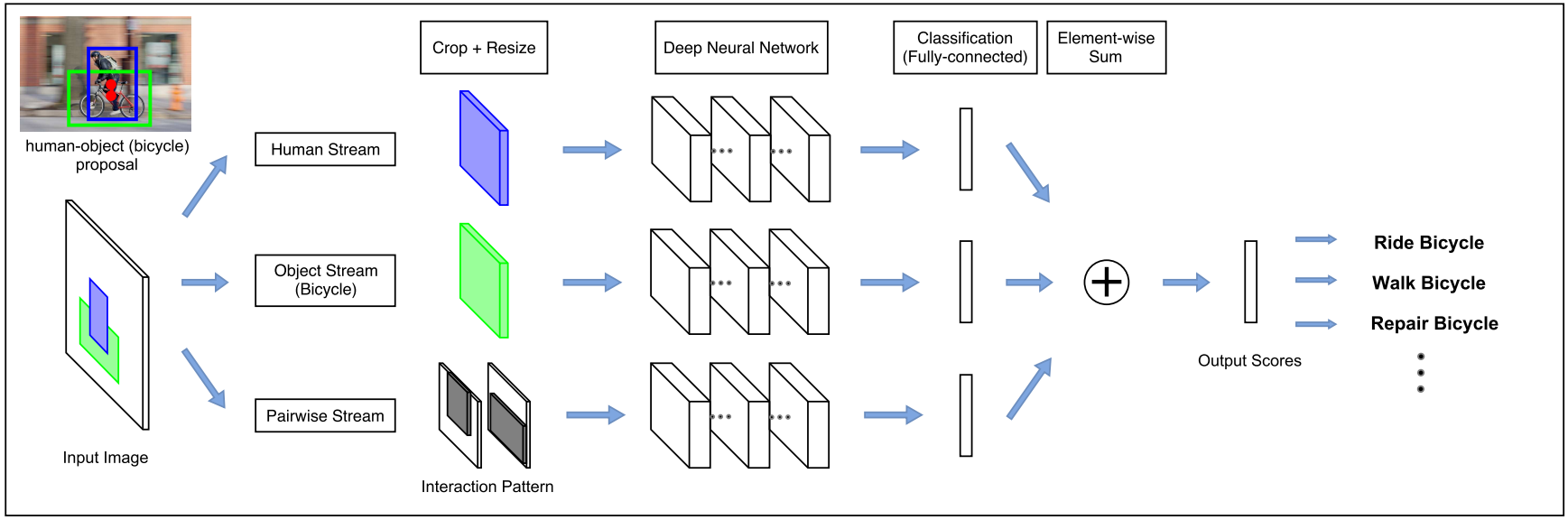


Figure 4.3: Multi-stream architecture of our HO-RCNN.

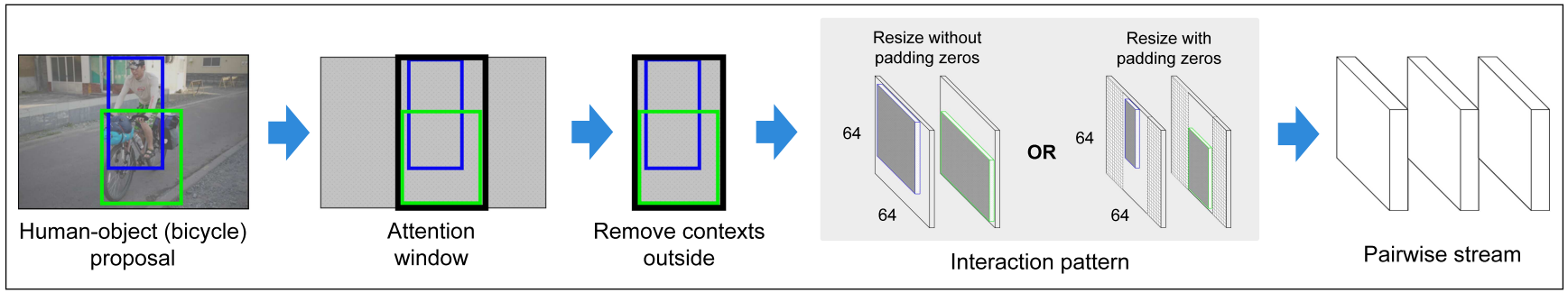


Figure 4.4: Construction of Interaction Patterns for the pairwise stream.

HOI classes of interest, and each output corresponds to the confidence score of one HOI class. The object stream follows the same design except that the input is cropped and resized from the object bounding box of the human-object proposal.

4.3.4 Pairwise Stream

Given a human-object proposal, the pairwise stream extracts features that encode the spatial relations between the human and object, and generates a confidence score for each HOI class. Since the focus is on spatial configurations of humans and objects, the input of this stream should ignore pixel values and only exploit information of bounding box locations. Instead of directly taking the bounding box coordinates as inputs, we propose *Interaction Patterns*, a special type of DNN input that characterizes the relative location of two bounding boxes. Given a pair of bounding boxes, its Interaction Pattern is a binary image with two channels: The first channel has value 1 at pixels enclosed by the first bounding box, and value 0 elsewhere; the second channel has value 1 at pixels enclosed by the second bounding box, and value 0 elsewhere. [2] In our pairwise stream, the first channel corresponds to the human bounding box and the second channel corresponds to the object bounding box. Take the input image in Fig. 4.3 as an example, where the person is "riding a bike". The first (human) channel will have value 1 at the upper central region, while the second (object) channel will have value 1 at the lower central region. This representation enables DNN to learn 2D filters that respond to similar 2D patterns of human-object spatial configurations.

While the Interaction Patterns are able to characterize pairwise spatial configurations, there are still two important details to work out. First, the Interaction Patterns should be invariant to any joint translations of the bounding box pair. In other words, the Interaction Patterns should be identical for identitcal pair configurations whether the pair appears on the right or the left side of the image. As a result, we remove all the pixels outside the "attention window", i.e. the tightest window enclosing the two bounding boxes, from the Interaction Pattern. This makes the pairwise stream focus solely on the local window containing the target bounding boxes and ignore global translations. Second, the aspect ratio of Interaction Patterns may vary depending on the attention window. This is problematic as DNNs take input of fixed size (and aspect ratio). We propose two strategies to address this issue: (1) We resize both sides of the Interaction Pattern to a fixed length regardless of its aspect ratio. Note that this may change the aspect ratio of the attention window. (2) We resize the longer side of the Interaction Pattern to a fixed length while keeping the aspect ratio, followed by padding zeros on both sides of the shorter side to achieve the fixed length. This normalizes the size of the Interaction Pattern while keeping the aspect ratio of the attention

[2] In this work, we apply the *second-order* Interaction Pattern for learning pairwise spatial relations. The Interaction Pattern can be extended to n-th order ($n \in N$) by stacking additional images in the channel axis for learning higher-order relations.

window. The construction of Interaction Patterns is illustrated in Fig. 4.4.

4.3.5 Training with Multi-Label Classification Loss

Given a human-object proposal, our HO-RCNN generates confidence scores for a list of HOI categories of interest. As noted in [23], a person can concurrently perform different classes of actions to a target object, e.g. a person can be "riding" and "holding" a bicycle at the same time. Thus HOI recognition should be treated as a mulit-label classification as opposed to the standard K-way classification. As a result, we train the HO-RCNN by applying a sigmoid cross entropy loss on the classification output of each HOI category, and compute the total loss by summing over the individual losses.

4.4 Constructing HICO-DET

We contribute a new large-scale benchmark for HOI detection by augmenting HICO [23] with instance annotations. HICO currently contains only image-level annotations, i.e. 600 binary labels indicating the presence of the 600 HOI classes of interest (e.g. "feeding a cat", "washing a knife"). We further annotate the HOI instances presented in each image, where each instance is represented by a pairing between a human and object bounding box with a class label (Fig. 4.5).

We collect human annotations by setting up annotation tasks on Amazon Mechanical Turk (AMT). However, there are two key issues: First, given an image and a presented HOI class (e.g. " riding a bike"), the annotation task is not as trivial as drawing bounding boxes around all the humans and objects associated with the interaction (e.g. "bike") — we also need to identify the interacting relations, i.e. linking each person to the objects he is interacting with. Second, although this linking step can be bypassed if the annotator is allowed to draw only one human bounding box followed by one object bounding box each time, such strategy is time intensive. Considering the cases where there are multiple people interacting with one object (e.g. "boarding an airplane" in Fig. 4.5), or one person interacting with multiple objects (e.g. "herding cows" in Fig. 4.5), the annotator then has to repeatedly draw bounding boxes around the shared persons and objects. [3] To efficiently collect such annotations, we adopt a *three-step* annotation procedure (Fig. 4.6). For each image, the annotator is presented with a sentence description, such as "A person riding a bicycle", and asked to proceed with the following three steps:

- **Step 1: Draw a bounding box around each person.** The first step is to draw bounding

[3] Although we formulate HOI detection as localizing interactions between a single person and a single object, actual interactions can be more complex such as the one-versus-many and many-versus-one cases. However, these types of interactions can be decomposed into multiple instances of person-object interaction pairs. Our goal is to detect all the decomposed person-object pairs in such cases.

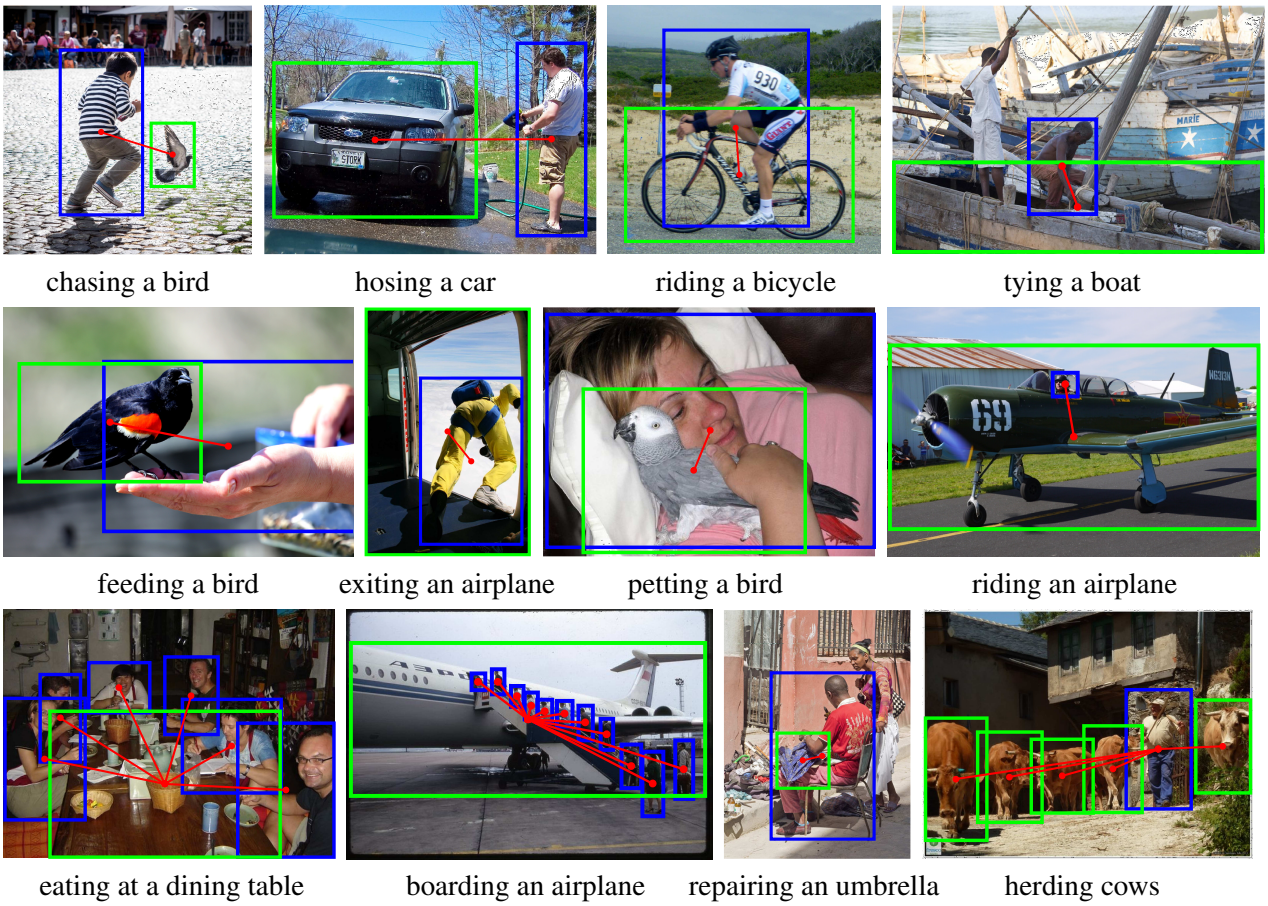


Figure 4.5: Sample annotations of our HICO-DET.

boxes around each person involved in the described interaction (e.g. each person riding bicycles). Note that the annotators are explicitly asked to ignore any person not involved in the described interaction, (e.g. any person not riding a bicycle), since those people do not participate in any instances of "riding a bicycle".

- **Step 2: Draw a bounding box around each object.** The second step is to draw bounding boxes around each object involved in the described interaction, (e.g. each bicycle being ridden by someone). Similar to the first step, the annotator should ignore any object that is not involved in the described interaction (e.g. any bicycles not being ridden by someone).

- **Step 3: Linking each person to objects.** The final step is to link a person bounding box to an object bounding box if the described interaction is taking place between them (e.g. link a person to a bicycle if the person is riding the bicycle). Note that one person can be linked to multiple objects if he is interacting with more than one objects (e.g. "herding cows" in Fig. 4.5), and one object can be linked with multiple people if it is the case that more than one person are interacting with it (e.g. "boarding an airplane" in Fig. 4.5).

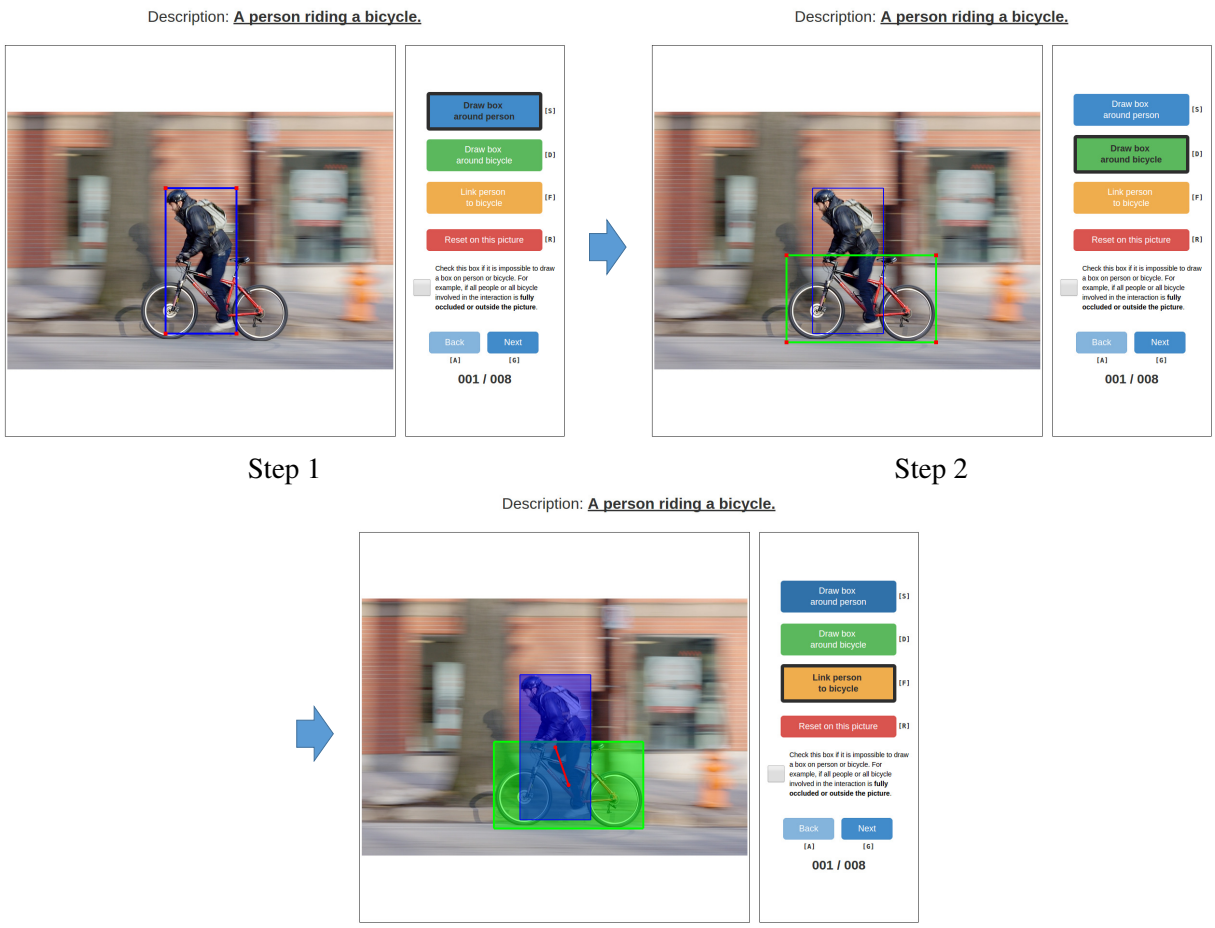


Figure 4.6: Our data annotation task for each image involves three steps.

Note that in some rare cases, the involved person or object may be invisible, even though the presence of the HOI can be inferred from the image. (e.g. We can tell a person is "sitting on a chair" although the chair is fully-occluded by the person's body.) If the involved person or object is completely invisible in the image, the annotator is asked to mark those images as "invisible". Among all 90641 annotation tasks (each corresponds to one positive HOI label for one image in HICO), we found that there are 1209 (1.33%) tasks labeled as "invisible". Since our instance annotations are built upon HICO's HOI class annotations, our HICO-DET also has a long-tail distribution in the number of instances per HOI class as in HICO. By keeping the same training-test split, we found that there are 2 out of 600 classes ("jumping a car" and "repairing a mouse") which have no training instances due to the invisibility of people or objects. As a result, we added 2 new images to our HICO-DET so we have at least one training instance for each of the 600 HOI classes. Tab. 4.1 shows the statistics of the newly collected annotations. We see that each image in HICO-DET has on average more than one (1.67) instance for each positive HOI label. Note

	HICO-DET			
	#image	#positive	#instance	#bounding box
Train	38118	70373	117871 (1.67/pos)	199733 (2.84/pos)
Test	9658	20268	33405 (1.65/pos)	56939 (2.81/pos)
Total	47776	90641	151276 (1.67/pos)	256672 (2.83/pos)

Table 4.1: Statistics of annotations in our HICO-DET.

that the total number of bounding boxes (256672) is less than twice the total number of instances (151274). This is because different instances can share people or objects, as shown in Fig. 4.5.

4.5 Experiments

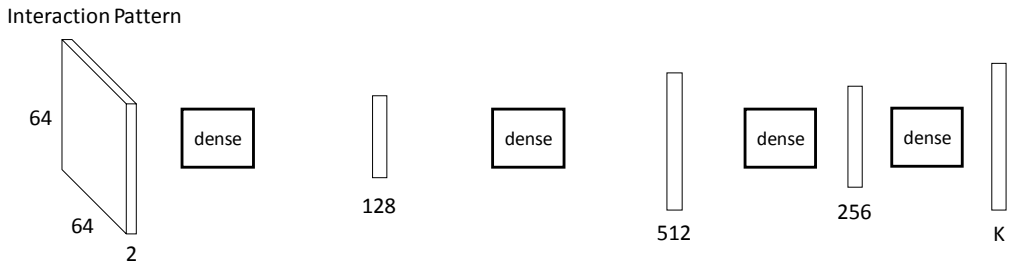
4.5.1 Evaluation Setup

Following the standard evaluation metric for object detection, we evaluate HOI detection using mean average precision (mAP). In object detection, a detected bounding box is assigned a true positive if it overlaps with a ground truth bounding box of the same class with intersection over union (IoU) greater than 0.5. Since we predict one human and one object bounding box in HOI detection, we declare a true positive if the minimum of human overlap IoU_h and object overlap IoU_o exceeds 0.5, i.e. $\min(\text{IoU}_h, \text{IoU}_o) > 0.5$. We report the mean AP over three different HOI category sets: (a) all 600 HOI categories in HICO (Full), (b) 138 HOI categories with less than 10 training instances (Rare), and (c) 462 HOI categories with 10 or more training instances (Non-Rare). All reported results are evaluated on the test set.

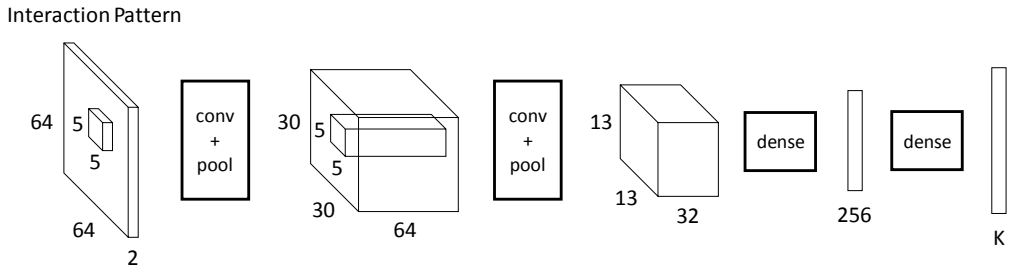
Following the HICO classification benchmark [23], we also consider two different evaluation settings: (1) *Known Object* setting: For each HOI category (e.g. "riding a bike"), we evaluate the detection only on the images containing the target object category (e.g. "bike"). The challenge is to localize HOI (e.g. human-bike pairs) as well as distinguishing the interaction (e.g. "riding"). (2) *Default* setting: For each HOI category, we evaluate the detection on the full test set, including images both containing and not containing the target object category. This is a more challenging setting as we also need to distinguish background images (e.g. images without "bike").

4.5.2 Training HO-RCNN

We first generate human-object proposals using state-of-the-art object detectors. Since HICO and MS COCO [123] share the same 80 object categories, we train 80 object detectors using Fast-RCNN [65] on the MS COCO training set. As detailed in Sec. 4.3, we generate proposals for each HOI category (e.g. "riding a bike") by pairing the top detected humans and objects (e.g. "bike")



(a) Fully-connected network (fc).



(b) Convolutional network (conv).

Figure 4.7: Two different architectures for the pairwise stream. For fair comparison, both networks have approximately the same number of parameters, and are trained with identical schemes.

in each image. In our experiments, we adopt the top 10 detections for human and each object category, resulting in 100 proposals per object category per image.

We implement our HO-RCNN using Caffe [89]. For both the human and object streams, we adopt the CaffeNet architecture with weights pre-trained on the ImageNet classification task [161]. To train on HICO-DET, we run SGD with a global learning rate 0.001 for 100k iterations, and then lower the learning rate to 0.0001 and run for another 50k iterations. We use an *image-centric* sampling strategy similar to [65] for mini-batch sampling: Each mini-batch of size 64 is constructed from 8 randomly sampled images, with 8 randomly sampled proposals for each image. These 8 proposals are from three different sources. Suppose a sampled image contains interactions with "bike", we sample: (a) 1 *positive example*: human-bike proposals that have $\min(\text{IoU}_h, \text{IoU}_o) \geq 0.5$ with at least one ground-truth instance from a category involving "bike". (b) 3 *type-I negatives*: non-positve human-bike proposals that have $\min(\text{IoU}_h, \text{IoU}_o) \in [0.1, 0.5)$ with at least one ground-truth instance from a category involving "bike". (c) 4 *type-II negatives*: proposals that do not involve "bike".

4.5.3 Ablation Study

We first perform an ablation study on the pairwise stream. We consider the two different variants of the Interaction Patterns described in Sec. 4.3, i.e. without padding (IP0) and with padding (IP1), each paired with two different DNN architectures: a fully-connected network (fc)

	Default			Known Object		
	Full	Rare	Non-Rare	Full	Rare	Non-Rare
HO	5.73	3.21	6.48	8.46	7.53	8.74
HO+vec0 (fc)	6.47	3.57	7.34	9.32	8.19	9.65
HO+vec1 (fc)	6.24	3.59	7.03	9.13	8.09	9.45
HO+IP0 (fc)	7.07	4.06	7.97	10.10	8.38	10.61
HO+IP1 (fc)	6.93	3.91	7.84	10.07	8.43	10.56
HO+IP0 (conv)	7.15	4.47	7.95	10.23	8.85	10.64
HO+IP1 (conv)	**7.30**	**4.68**	**8.08**	**10.37**	**9.06**	**10.76**

	Default			Known Object		
	Full	Rare	Non-Rare	Full	Rare	Non-Rare
HO+vec1 (fc) vs. HO	< 0.001	0.132	< 0.001	< 0.001	0.077	< 0.001
HO+IP1 (conv) vs. HO	< 0.001	0.001	< 0.001	< 0.001	0.005	< 0.001
HO+IP1 (conv) vs. HO+vec1 (fc)	< 0.001	0.001	< 0.001	< 0.001	0.049	< 0.001

Table 4.2: Performace comparison of difference pairwise stream variants. Top: mAP (%). Bottom: p-value for the paired t-test.

and a convolutional network (conv). The two architectures are illustrated in Fig. 4.7. We also report baselines that use the same fc architecture but take the 2D vector from human's center to object's center (vec0: without padding, vec1: with padding). Tab. 4.2 (top) reports the mAP of using the human and object stream alone (HO) as well as combined with different pairwise streams (vec0 (fc), vec1 (fc), IP0 (fc), IP1 (fc), IP0 (conv), IP1 (conv)). Note that for all the methods, the Default setting has lower mAPs than the Known Object setting due to the increasing challenge in the test set, and the rare categories have lower mAP than the none-rare categories due to sparse training examples. Although the mAPs are low overall (i.e. below 11%), we still observe in both settings that adding a pairwise stream improves the mAP. Among all pairwise streams, using Interaction Patterns with the conv architecture achieves the highest mAP (e.g. for IP1 (conv) on the full dataset, 7.30% in the Default setting and 10.37% in the Known Object setting). To demonstrate the signficance of the improvements, we perform *paired t-test*: We compare two methods by their AP difference on each HOI category. The null hypothesis is that the mean of the AP differences over the categories is zero. We show the p-values in Tab. 4.2 (bottom). While the 2D vector baselines outperform the HO baseline in mAP, the p-value is above 0.05 on rare categories (e.g. 0.13 for "HO+vec1 (fc) vs. HO" in the Default setting). On the other hand, "HO+IP1 (conv) vs. HO" and "HO+IP1 (conv) vs. HO+vec1 (fc)" both have all p-values below 0.05, suggesting that using Interaction Patterns with the conv architecture has a significant improvement not only over the HO baseline, but also over the 2D vector baseline.

We show the average Interaction Patterns obtained from the ground-truth annotations of different HOIs in Fig. 4.8. We see distinguishable patterns for different interactions on the same object

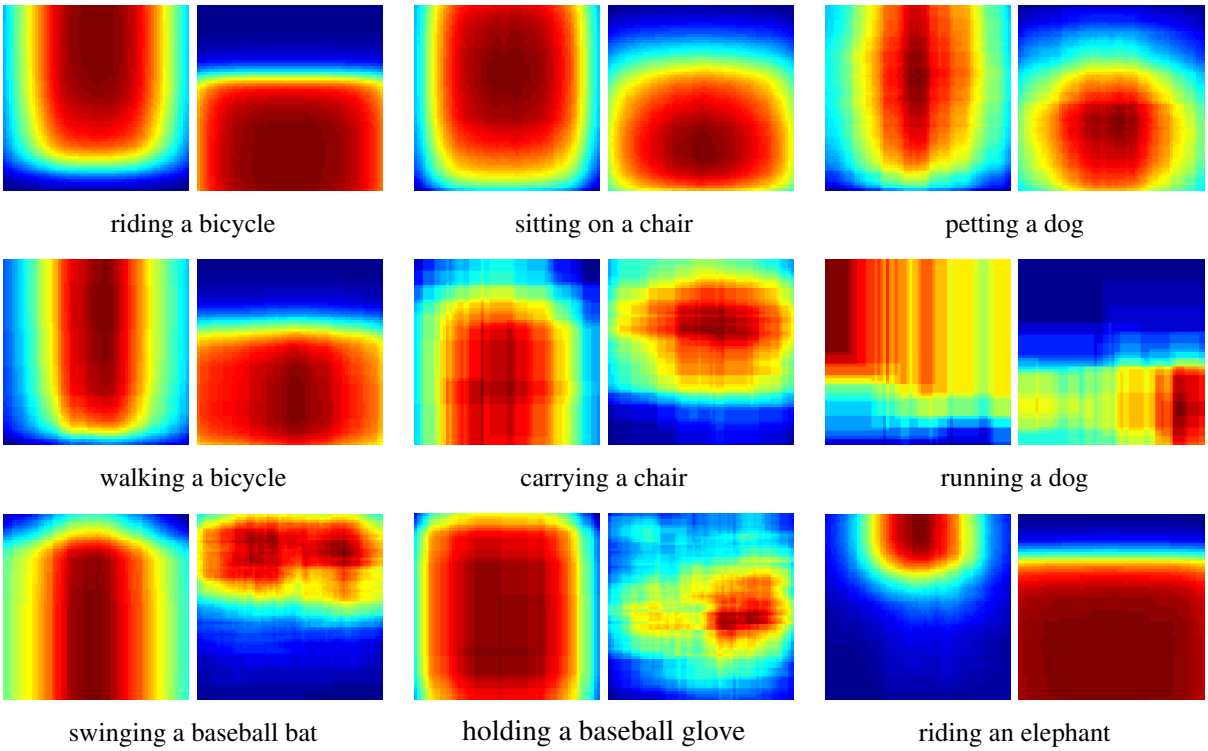


Figure 4.8: Average Interaction Patterns for different HOI categories btained from ground-truth annotations. Left: average for the human channel. ight: average for the object channel.

	Default			Known Object		
	Full	Rare	Non-Rare	Full	Rare	Non-Rare
Human	0.70	0.08	0.88	2.44	2.14	2.53
Object	**2.11**	**1.19**	**2.39**	3.09	**2.98**	3.13
Pairwise	0.30	0.06	0.37	**3.21**	2.80	**3.33**

Table 4.3: mAP (%) of each stream on HO+IP1 (conv).

category. For example, a chair involved in "sitting on a chair" is more likely to be in the lower region of the Interaction Pattern, while a chair involved in "carrying a chair" is more likely to be in the upper region.

We also separately evaluate the output of human, object, and pairwise stream. Tab. 4.3 shows the mAP of each stream on HO+IP1 (conv). The object stream outperforms the other two in the Default setting. However, in the Known Object setting, the pairwise stream achieves the highest mAP, demonstrating the importance of human-object spatial relations for distinguishing interactions.

4.5.4 Leveraging Object Detection Scores

So far we assume the human-object proposals always consist of true object detections, so the HO-RCNN is only required to distinguish the interactions. In practice, the proposals may contain

	Default			Known Object		
	Full	Rare	Non-Rare	Full	Rare	Non-Rare
HO	5.73	3.21	6.48	8.46	7.53	8.74
HO+S	6.07	3.79	6.76	8.09	6.79	8.47
HO+IP1 (conv)	7.30	4.68	8.08	10.37	**9.06**	10.76
HO+IP1 (conv)+S	**7.81**	**5.37**	**8.54**	**10.41**	8.94	**10.85**

	Default		
	Full	Rare	Non-Rare
HO+S vs. HO	0.002	0.024	0.016
HO+IP1 (conv)+S vs. HO+IP1 (conv)	< 0.001	0.028	< 0.001

Table 4.4: Performance comparison of combining object detection scores. Top: mAP (%). Bottom: p-value for the paired t-test.

	Number of human (object) detections			
	10	20	50	100
Full	46.75	51.56	57.17	60.37
Rare	54.15	58.62	64.98	68.40
Non-Rare	44.54	49.45	54.84	57.97

Table 4.5: Mean recall (%) of human-object proposals on the training set.

false detections, and the HO-RCNN should learn to generate low scores for all HOI categories in such case. We thus add an extra path with a single neuron that takes the raw object detection score associated with each proposal and produces an offset to the final HOI detection scores. This provides a means by which the final detection scores can be lowered if the raw detection score is low. We show the effect of adding this extra component (HO+S and HO+IP1 (conv)+S) in Tab. 4.4 (top) and the signifiance of the improvements in Tab. 4.4 (bottom). The improvement is significant in the Default setting, since the extra background images increase the number of false object detections.

4.5.5 Error Analysis

We hypothesize that the low AP classes suffer from excessive false negatives. To verify this hypothesis, we compute the recall of the human-object proposals for each HOI category. Tab. 4.5 shows the mean recall on the training set by varying the numbers of used human (object) detections. When adopting 10 human (object) detections, we see a low mean recall (46.75%), which expalins the low mAPs in our results. Although the mAPs can be potentially improved by adopting more human (object) detections, the number of human-object proposals will increase quadratically, making the evaluation of all proposals infeasible. This thus calls for better approaches to construct

	Default			Known Object		
	Full	Rare	Non-Rare	Full	Rare	Non-Rare
Random	1.35×10^{-3}	5.72×10^{-4}	1.62×10^{-3}	0.19	0.17	0.19
Fast-RCNN [65] (union)	1.75	0.58	2.10	2.51	1.75	2.73
Fast-RCNN [65] (score)	2.85	1.55	3.23	4.08	2.37	4.59
HO	5.73	3.21	6.48	8.46	7.53	8.74
HO+IP1 (conv)	7.30	4.68	8.08	10.37	**9.06**	10.76
HO+IP1 (conv)+S	**7.81**	**5.37**	**8.54**	**10.41**	8.94	**10.85**

Table 4.6: Comparison of mAP(%) with prior approaches.

high-recall human-object proposals in future studies.

4.5.6 Comparison with Prior Approaches

To compare with prior approaches, we consider two extensions to Fast-RCNN [65]. (1) Fast-RCNN (union): For each human-object proposal, we take their attention window as the region proposal for Fast-RCNN. This can be seen as a "single-stream" version of HO-RCNN where the feature is extracted from the tightest window enclosing the human and object bounding box. (2) Fast-RCNN (score): Given the human-object proposals obtained from the object detectors, we train a classifier to classify each HOI category by linearly combining the human and object detection scores. Note that this method does not use any features from the human and object regions nor their spatial relations. We also report a baseline that randomly assigns scores to our human-object proposals (Random). Tab. 4.6 shows the mAP of the compared methods and different vairants of our HO-RCNN. In both settings, Fast-RCNN (union) performs worse than all other methods except the random baseline. This suggests that the feature extracted from the attention window is not suitable for distinguishing HOI, possibly due to the irrelevant contexts between the human and object when the two bounding boxes are far apart. Fast-RCNN (score) performs better than Fast-RCNN (union), but still worse than all our HO-RCNN variants. This is because object detection scores alone do not contain sufficient information for distinguishing interactions. Finally, our HO+IP1 (conv)+S and HO+IP1 (conv) outperform all other methods in both the Default and the Known Object setting. Fig. 4.9 shows qualitative examples of the detected HOIs from our HO-RCNN. We show both the true positives (left) and false positives (right).

4.6 Summary

We study the detection of human-object interactions in static images. We introduce HICO-DET, a new large benchmark, by augmenting the HICO classification benchmark with instance annotations. We propose HO-RCNN, a novel DNN-based framework. At the core of HO-RCNN

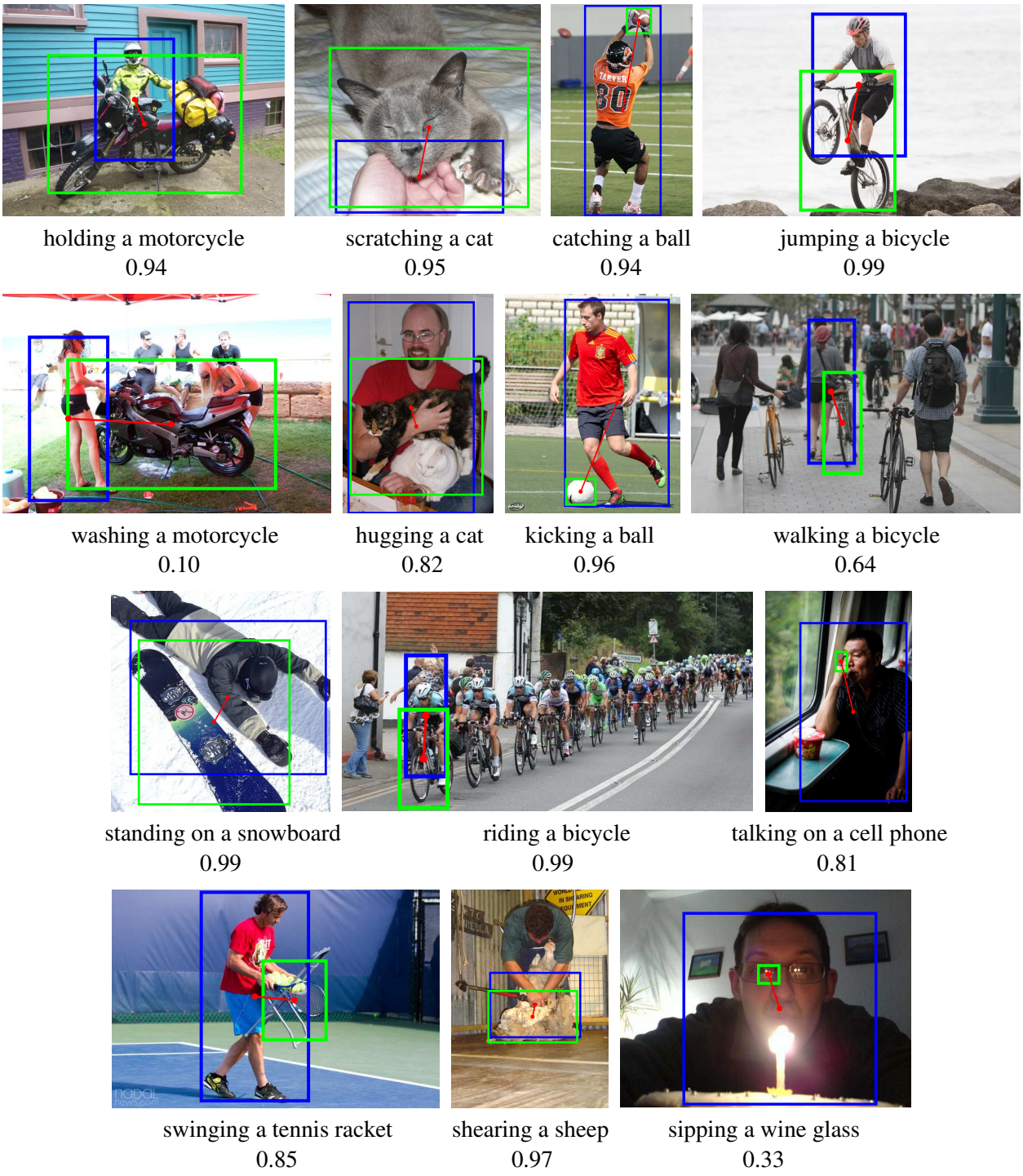


Figure 4.9: Qualitative examples of detections from our HO-RCNN. We show the HOI class and the output probability below each detection. Top two rows: true positives. Bottom two rows: false positives (left/middle/right: incorrect interaction class/inaccurate bounding box/false object detection).

is the Interaction Pattern, a novel DNN input that characterizes the spatial relations between two bounding boxes. Experiments show that HO-RCNN significantly improves the performance of

HOI detection over baseline approaches.

CHAPTER 5

Human Pose Forecasting [1]

5.1 Introduction

Human pose forecasting is the capability of predicting future human body dynamics from visual observations. Human beings are endowed with this great ability. For example, by looking at the left image of Fig. 5.1, we can effortlessly imagine the upcoming body dynamics of the target tennis player, namely a forehand swing, as shown in the right image of Fig. 5.1 Such prediction is made by reasoning on the scene context (i.e. a tennis court), the current body pose of the target (i.e. standing and holding a tennis racket), and our visual experience of a tennis forehand swing.

The ability of forecasting reflects a higher-level intelligence beyond perception and recognition and plays an important role for agents to survive from challenging natural and social environments. In the context of human-robot interactions, such ability is particularly crucial for assistant robots that need to interact with surrounding humans in an efficient and robust manner. Apparently, the abilities of identifying and localizing the action categories [170, 48, 224, 168] after observing an image or video are not sufficient to achieve this goal. For example, when a person throws a ball at a robot, the robot needs to identify the action and forecast the body pose trajectory even before the person finishes so that it can response effectively (either by catching the ball or dodging it).

This chapter presents the first study on human pose forecasting from static images. Our task is to take a single RGB image and output a sequence of future human body poses. Our approach has two key features. First, as opposed to other forecasting tasks that assume a multi-frame input (i.e. videos) [176, 60, 134], our work assumes a single-frame input. Although this assumption increases the learning challenge due to the lack of explicit motion cues, it encourages the algorithm to learn high-level dynamics instead of low-level smoothness. Note that our approach can be trivially extended to take multi-frame inputs as shown later in the methodology section. Second, like most forecasting problems [229, 194, 153, 195, 193], we first represent the forecasted poses in the 2D image space. However, we include an extra component to our approach to further

[1] This chapter is based on a joint work with Jimei Yang, Brian Price, Scott Cohen, and Jia Deng [26].

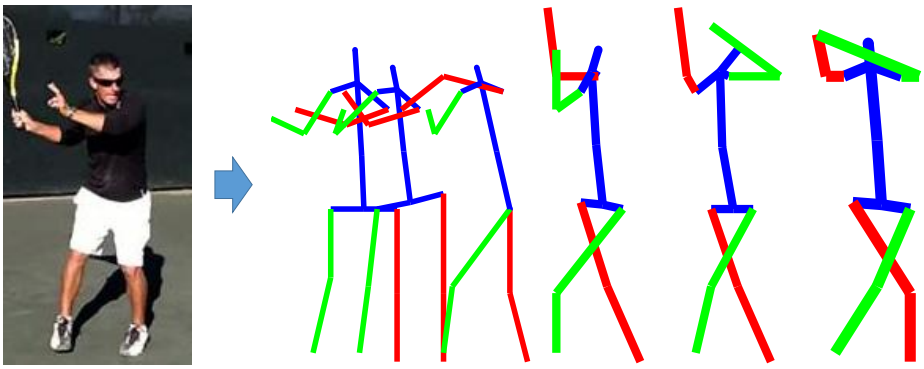


Figure 5.1: Forecasting human dynamics from static images. Left: the input image. Right: the sequence of upcoming poses.

convert each forecasted pose from 2D space to 3D space. Both forecasting and 3D conversion are performed using a deep neural network (DNN). The two networks are integrated into one single unified framework to afford end-to-end training. Since human bodies feature a complex articulated structure, we believe the 3D output is more actionable and useful for future applications (e.g. shape and texture rendering) as we will demonstrate in Sec. 5.4.3.

The main contributions of this chapter are three-fold: (1) We present the first study on single-frame human pose forecasting. This extends the dimension of current studies on human pose modeling from recognition (i.e. pose estimation [184, 143]) to forecasting. The problem of pose forecasting in fact generalizes pose estimation, since to forecast future poses we need to first estimate the observed pose. (2) We propose a novel DNN-based approach to address the problem. Our forecasting network integrates recent advances on single-image human pose estimation and sequence prediction. Experimental results show that our approach outperforms strong baselines on 2D pose forecasting. (3) We propose an extra network to convert the forecasted 2D poses into 3D skeletons. Our 3D recovery network is trained on a vast amount of synthetic examples by leveraging motion capture (MoCap) data. Experimental results show that our approach outperforms two state-of-the-art methods on 3D pose recovery. In a nutshell, we propose a unified framework for 2D pose forecasting and 3D pose recovery. Our *3D Pose Forecasting Network* (3D-PFNet) is trained by leveraging a diverse source of training data, including image and video based human pose datasets and MoCap data. We separately evaluate our 3D-PFNet on 2D pose forecasting and 3D pose recovery, and show competitive results over baselines.

5.2 Related Work

5.2.1 Visual Scene Forecasting

Our work is in line with a series of recent work on single-image visual scene forecasting. These works vary in the predicted target and the output representation. [112] predicts human actions in the form of semantic labels. Some others predict motions of low level image features, such as the optical flow to the next frame [153, 195] or dense trajectories of pixels [229, 193]. A few others attempt to predict the motion trajectories of middle-level image patches [194] or rigid objects [140]. However, these methods do not explicitly output a human body model, thus cannot directly address human pose forecasting. Notably, [60] predicts the future dynamics of a 3D human skeleton from its past motion. Despite its significance, their method can be applied to only 3D skeleton data but not visual inputs. Our work is the first attempt to predict 3D human dynamics from RGB images.

5.2.2 Human Pose Estimation

Our work is closely related to the problem of human pose estimation, which has long been attractive in computer vision. Human bodies are commonly represented by tree-structured skeleton models, where each node is a body joint and the edges capture articulation. The goal is to estimate the 2D joint locations in the observed image [184, 143] or video sequences [146, 68]. Recent work has even taken one step further to directly recover 3D joint locations [119, 223, 180, 50, 28, 160] or body shapes [10] from image observations. While promising, these approaches can only estimate the pose of humans in the observed image or video. Our approach not only estimate the human pose in the observed image, but also forecasts the poses in the upcoming frames. Besides estimation from images or videos, an orthogonal line of research addresses the recovery of 3D body joint locations from their 2D projections [7, 238, 239, 205]. Our work also takes advantage of these approaches to transform the estimated 2D joint locations into 3D space.

5.2.3 Video Frame Synthesis

Two very recent works [208, 192] attempt to synthesize videos from static images by predicting pixels in future frames. This is a highly challenging problem due to the extremely high dimensional output space and the massive variations a scene can transform from a single image. Our work can provide critical assistance to this task by using the predicted human poses as intermediate representation to regularize frame synthesis, e.g. it is easier to synthesize a baseball pitching video from a single photo of a player if we can forecast his body dynamics. In addition to static images, there are also other efforts addressing video prediction from video inputs [176, 134, 58], which can be benefited by our work in the same way.

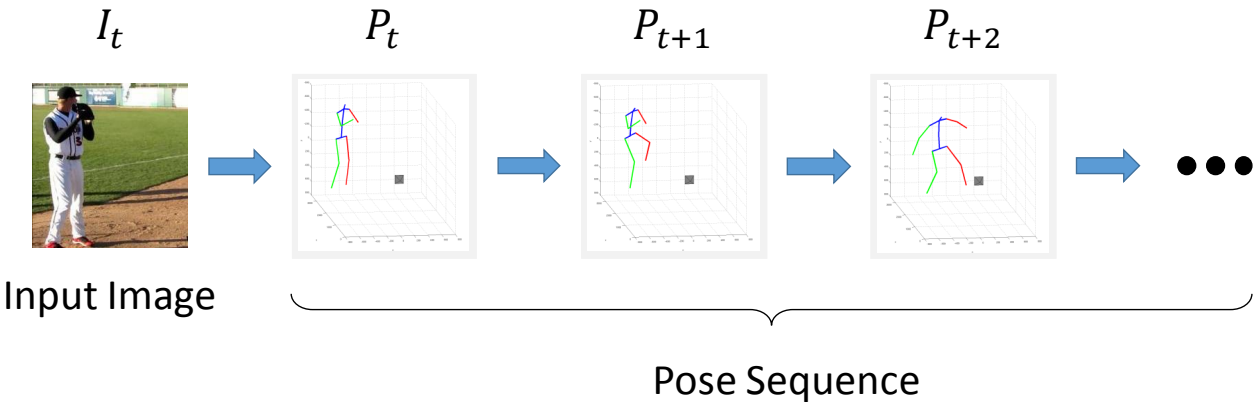


Figure 5.2: The problem of human pose prediction. The input is a single image, and the output is a 3D pose sequence.

5.3 Approach

5.3.1 Problem Statement

The problem studied in this chapter assumes the input to be a single image captured at time t. The output is a sequence of 3D human body skeletons $P = \{P_t, \ldots, P_{t+T}\}$, where $P_i \in \mathbb{R}^{3\times N}$ denotes the predicted skeleton at time i, represented by the 3D locations of N keypoints. See Fig. 5.2 for an illustration of the problem. Note that this formulation generalizes single-frame 3D human pose estimation, which can be viewed as a special case when $T = 0$.

5.3.2 Network Architecture

We propose a deep recurrent network to predict human skeleton sequences (Fig. 5.3). The network is divided into two components: first, a 2D pose sequence generator that takes an input image and sequentially generates 2D body poses, where each pose is represented by heatmaps of keypoints; second, a 3D skeleton converter that converts each 2D pose into a 3D skeleton.

5.3.2.1 2D Pose Sequence Generator

The first step is to generate a 2D body pose sequence from the input image. The task can be decomposed into estimating the body pose in the given frame and predicting the body poses in upcoming frames. We thus leverage recent advances on single-frame human pose estimation as well as sequence prediction. The recently introduced *hourglass* networks [143] have demonstrated state-of-the-art performance on large-scale human pose datasets [8]. We summarize the hourglass architecture as follows: The first half of the hourglass processes the input image with convolution and pooling layers to a set of low resolution feature maps. This resembles conventional ConvNets (and is frequently referred to as "encoder" in generative models). The second half (frequently

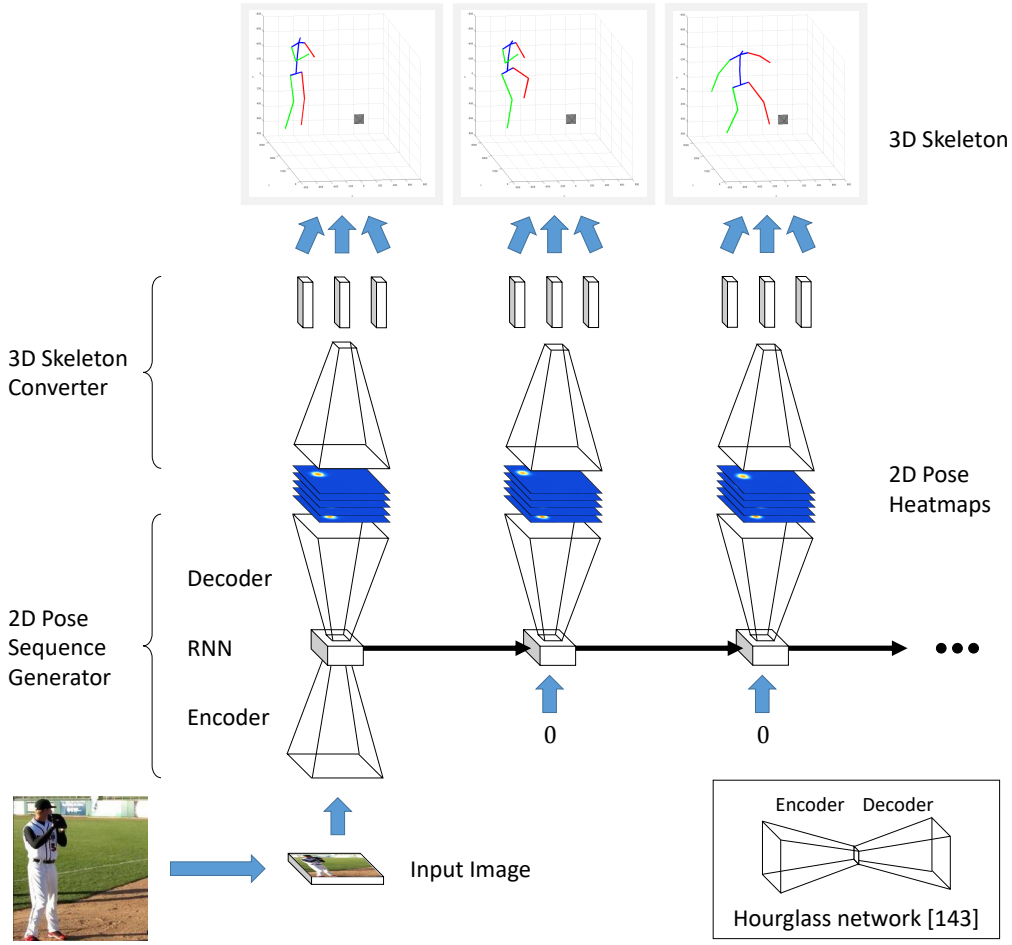


Figure 5.3: A schematic view of the unrolled 3D-PFNet.

referred to as "decoder") then processes the low resolution feature maps with a symmetric set of upsampling and convolution layers to generate dense detection heatmaps for each keypoint at high resolution. A critical issue of this architecture is the loss of high resolution information in the encoder output due to pooling. Thus one key ingredient is to add a "skip connection" before each pooling layer to create a direct path to the counterpart in the decoder. As a result, the hourglass can consolidate features from multiple scales in generating detection outputs.

While achieving promising results on single-frame pose estimation, the hourglass network is incapable of predicting future poses. A straightforward extension is to increase the channel size of its output to jointly generate predictions for future frames [193, 192]. However, the drawback is that a trained network will always predict output for a fixed number of frames. To bypass this constraint, we choose to formulate pose forecasting as a sequence prediction problem by adopting recurrent neural networks (RNNs).

RNNs extend conventional DNNs with feedback loops to enable sequential prediction from internal states driven by both current and pass observations. Our key idea is to introduce an RNN

to the neck of the hourglass, i.e. between the encoder and decoder. We hypothesize that the global pose features encoded in the low resolution feature maps are sufficient to drive the future predictions. We refer to the new network as the *recurrent hourglass* architecture. Fig. 5.3 illustrates the process of generating pose sequence from the unrolled version of the recurrent hourglass network. First, the given image is passed into the encoder to generate low resolution feature maps. These feature maps are then processed by an RNN to update its internal states. Note that the internal states here can be viewed as the "belief" on the current pose. This "belief" is then passed to the decoder to generate pose heatmaps for the input image. To generate pose for the next timestep, this "belief" is fed back to the RNN and then updated to account for the pose change. The updated "belief" is again passed to the decoder to generate heatmaps for the second timestep. This process will repeat, and in the end we will obtain a sequence of 2D pose heatmaps. Since we assume a single-image input, the encoder is used only in the initial frame. Starting from the second frame, the input to RNN is set to zeros. As mentioned earlier, it is natural to extend our model to video inputs by adding an encoder at every timestep.

For the RNN, we adopt the long short-term memory (LSTM) architecture [92] due to its strength in preserving long-term memory. We apply two tricks: First, conventional LSTMs are used in fully-connected architectures. Since the hourglass network is fully convolutional and the encoder output is a feature map, we apply the LSTM convolutionally on each pixel. This is equivalent to replacing the fully-connected layers in LSTM by 1×1 convolution layers. Second, we apply the residual architecture [76] in our RNN to retain a direct path from the encoder to the decoder. As a result, we place less burden on the RNN as it only needs to learn the "changes" in poses. Fig. 5.4 (a) shows the detailed architecture of our recurrent hourglass networks. Note that we also place an RNN on the path of each skip connection of the hourglass.

5.3.2.2 3D Skeleton Converter

The second step is to convert the heatmap sequence into a sequence of 3D skeletons. Many recent works have addressed the problem of recovering 3D skeleton structures from the 2D projection of their keypoints [238, 205]. Zhou et al. [238] assumes the unknown 3D pose can be approximated by a linear combination of a set of predefined basis poses, and propose to minimize reprojection error with a convex relaxation approach. Wu et al. [205] adopts a similar assumption but instead uses a DNN to estimate the linear coefficients and camera parameters. Both methods use a top-down approach by leveraging a set of "prior pose" models. On the contrary, we propose a bottom-up, data driven approach that directly predicts the 3D keypoint locations from local 2D features. We hypothesize that the bottom-up reconstruction can outperform top-down approaches given sufficiently complex models and a vast amount of training data.

We model 3D skeletons and their 2D projection with a perspective projection model. Recall

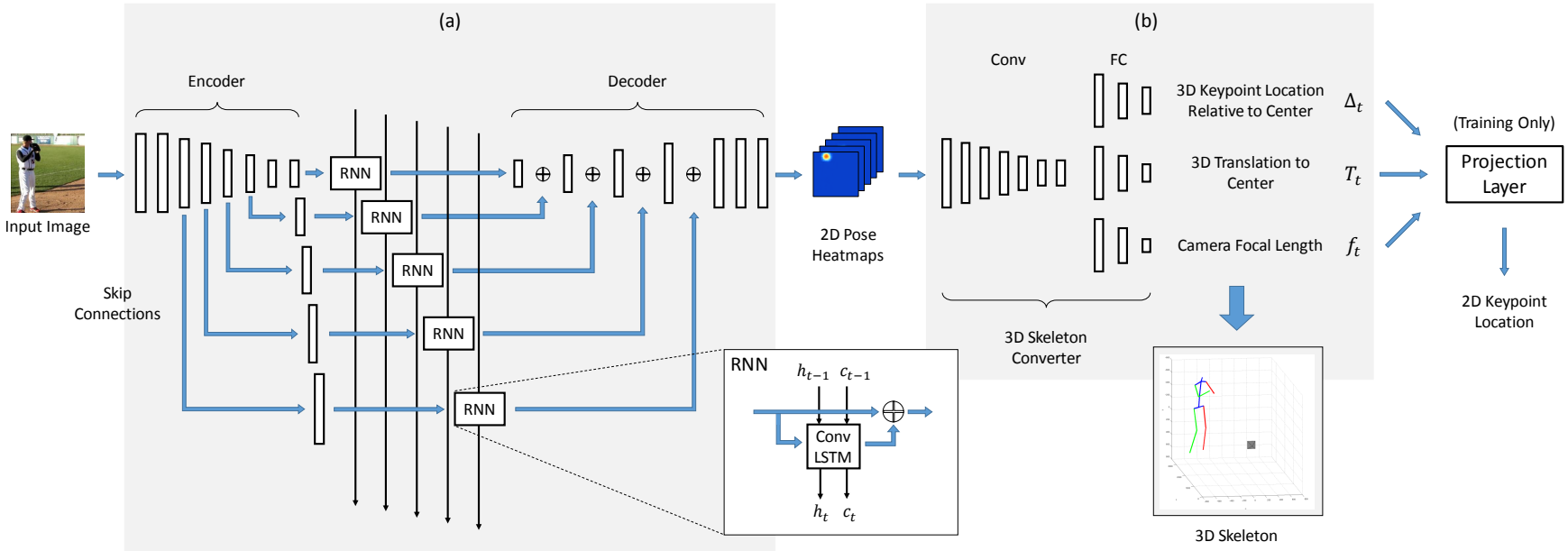


Figure 5.4: Architecture of the 3D-PFNet. (a) The recurrent hourglass architecture for 2D pose forecasting. (b) The 3D skeleton converter.

that a 3D skeleton $P \in \mathbb{R}^{3\times N}$ is represented by N keypoints in the camera coordinate system. We can decompose P by $P = \Delta + T1^T$, where $\Delta \in \mathbb{R}^{3\times N}$ represents the relative position of the N keypoints to their center in 3D, and $T \in \mathbb{R}^{3\times 1}$ represents the translation to the center. Let f be the camera focal length and assume the principal point is at the image center. The goal of our 3D skeleton converter is to estimate $\{\Delta, T, f\}$ from the observed 2D heatmaps. Fig. 5.4 (b) details the architecture. The heatmaps generated at each timestep are first processed by another encoder. Now instead of connecting to a decoder, the encoder output is forwarded to three different branches. Each branch consists of three fully-connected layers, and the three branches will output Δ, T, and f, respectively. Note that estimating camera parameters is unnecessary if we have ground-truth 3D keypoint annotations to train our network. However, 3D pose data is hard to collect and thus are often unavailable in in-the-wild human pose datasets. With the estimated camera parameters, we can apply a projection layer [205] at the output of the network to project 3D keypoints to 2D, and measure the loss on reprojection error for training.

5.3.3 Training Strategy

Our 3D-PFNet is composed of multiple sub-networks. Different sub-networks serve different sub-tasks and thus can exploit different sources of training data. We therefore adopt a three-step, task-specific training strategy.

1. **Hourglass:** The hourglass network (i.e. encoder and decoder) serves the task of single-frame 2D pose estimation. We therefore pre-train the hourglass network by leveraging large human pose datasets that provide 2D body joint annotations. We follow the training setup in [143] and apply a Mean Squared Error (MSE) loss for the predicted and ground-truth heatmaps.

2. **3D Skeleton Converter:** Training the 3D skeleton converter requires correspondences between 2D heatmaps and 3D ground truth of $\{\Delta, T, f\}$. We exploit the ground-truth 3D human poses from motion capture (MoCap) data. We synthesize training samples using a technique similar to [205]: First, we randomly sample a 3D pose and camera parameters (i.e. focal length, rotation, and translation). We then project the 3D keypoints to 2D coordinates using the sampled camera parameters, followed by constructing the corresponding heatmaps. This provides us with a training set that is diverse in both human poses and camera viewpoints. We apply an MSE loss for each output of Δ, T, and f, and an equal weighting to compute the total loss.

3. **Full Network:** Finally, we train the full network (i.e. hourglass + RNNs + 3D skeleton converter) using static images and their corresponding pose sequences. To ease the training of LSTM, we apply curriculum learning similar to [211]: We start training the full network

with pose sequences of length 2. Once the training converges, we increase the sequence length to 4 and resume the training. We repeat doubling the sequence length whenever training converges. We train the network with two sources of losses: The first source is the heatmap loss used for training the hourglass. Since we assume the 3D ground truths are unavailable in image and video datasets, we cannot apply loss directly on Δ, T, and f. We instead apply a projection layer as mentioned earlier and adopt an MSE loss on 2D keypoint locations. Note that replacing 3D loss with projection loss might diverge the training and output implausible 3D body poses, since a particular 2D pose can be mapped from multiple possible 3D configurations. We therefore initialize the 3D converter network with weights learned from the synthetic data, and keep the weights fixed during the training of the full network.

5.4 Experiments

We evaluate our 3D-PFNet on two tasks: (1) *2D pose forecasting* and (2) *3D pose recovery*.

5.4.1 2D Pose Forecasting

5.4.1.1 Dataset

We evaluate pose forecasting in 2D using the Penn Action dataset [234]. Penn Action contains 2326 video sequences (1258 for training and 1068 for test) covering 15 sports action categories. Each video frame is annotated with a human bounding box along with the locations and visibility of 13 body joints. Note that we do not evaluate our forecasted 3D poses due to the lack of 3D annotations in Penn Action. During training, we also leverage two other datasets: MPII Human Pose (MPII) [8] and Human3.6M [87]. MPII is a large-scale benchmark for single-frame human pose estimation. Human3.6M consists of videos of acting individuals captured in a controlled environment. Each frame is provided with the calibrated camera parameters and the 3D human pose acquired from MoCap devices.

5.4.1.2 Evaluation Protocal

We preprocess Penn Action with two steps: First, since our focus is not on human detection, we crop each video frame to focus roughly around the human region: for each video sequence, we crop every frame using the tight box that bounds the human bounding box across all frames. Second, we do not assume the input image is always the starting frame of each video (i.e. we should be able to forecast poses not only from the beginning of a tennis forehand swing, but also from the middle or even shortly before the action finishes). Thus for a video with K frames, we generate

Figure 5.5: Sample sequences of the processed Penn Action dataset. The ction classes are: baseball swing, bench press, golf swing, jumping jacks, ull ups, and tennis serve.

K sequences by varying the starting frame. Besides, since adjacent frames contain similar poses, we skip frames when generating sequences. The number of frames skipped is video-dependent: Given a sampled starting frame, we always generate a sequence of length 16, where we skip every $(K-1)/15$ frames in the raw video sequence after the sampled starting frame. This is to ensure that our forecasted output can "finish" each action in a predicted sequence of length 16. Note that once we surpass the end frame of a video, we will repeat the last frame collected until we obtain 16 frames. This is to force the forecasting to learn to "stop" and remain at the ending pose once an action has completed. Fig. 5.5 shows sample sequences of our processed Penn Action.

To evaluate the forecasted pose, we adopt the standard Percentage of Correct Keypoints (PCK) metric [8] from 2D pose estimation. PCK measures the accuracy of keypoint localization by considering a predicted keypoint correct if it falls within certain normalized distance of the ground truth. This distance is normalized typically based on the size of the full body bounding box [214] or the head bounding box [8]. Since we have already cropped the frames based on full body bounding boxes, we normalize the distance by $\max(h, w)$ pixels, where h and w are the height and width of the cropped image. We ignore invisible joints, and compute PCK separately for each of the 16 timesteps on the test sequences.

5.4.1.3 Implementation Details

We use Torch7 [33] for our experiments. In all training, we use rmsprop for optimization. We train our 3D-PFNet in three steps as described in Sec. 5.3.3. First, we train the hourglass for single-

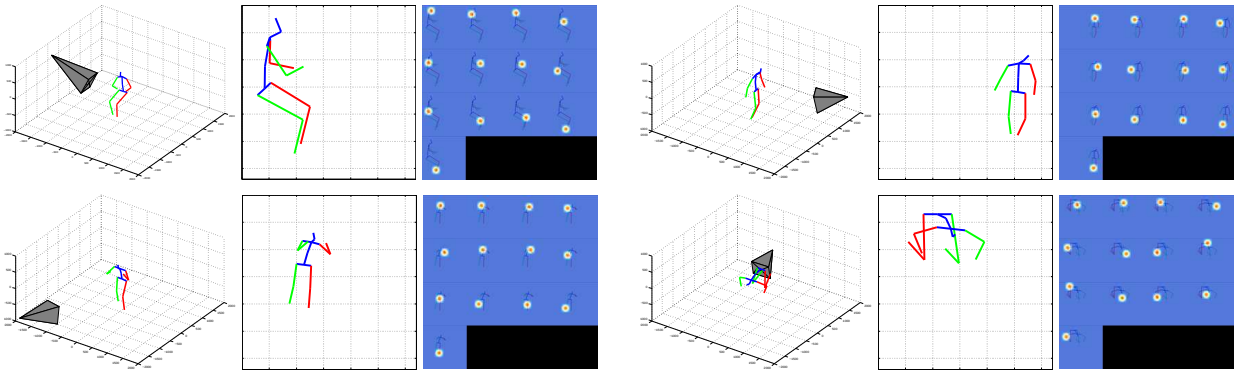


Figure 5.6: Samples of the synthetic data for training 3D skeleton converter. Each triplet consists of (1) the sampled 3D pose and camera in world coordinates, (2) the 2D projection, and (3) the converted heatmaps for 13 keypoints.

frame pose estimation by pre-training on MPII and fine-tuning on the preprocessed Penn Action. For both datasets, we partition a subset of the training set for validation. Second, we train the 3D skeleton converter using Human3.6M. Note that the image data in Human3.6M are unused here, since we only need 3D pose data for synthesizing camera parameters and 2D heatmaps. Following the standard data split in [87], we use poses of 5 subjects (S1, S7, S8, S9, S11) for training and 2 subjects (S5, S6) for validation. Fig. 5.6 shows samples of our synthesized training data. We use mini-batches of size 64 and a learning rate of 0.001. Finally, we train the full 3D-PFNet on the preprocessed Penn Action. We apply the curriculum learning scheme until convergence at sequence length 16. At test time, we always generate pose sequences of length 16.

5.4.1.4 Baselines

Since there are no prior approaches for pose forecasting, we devise our own baselines for comparison. We consider three baselines based on *nearest neighbor (NN)*. (1) *NN-all*: Given a test image, we first estimate the current human pose with an hourglass network and find the NN pose in the training images. We then transform the sequence of the NN pose to the test image as output. To measure distance between two poses (each represented by 13 2D keypoints), we first normalize the keypoints of each pose to have zero mean and unit maximum length from the center. We define distance by the MSE between two normalized poses. Since a ground-truth pose might contain invisible keypoints, we compute MSE only on the visible keypoints. Given the NN, we transform the associated sequence for the test image by reversing the normalization. (2) *NN-CaffeNet*: We hypothesize that the NN results can be improved by leveraging scene contexts. We therefore pre-filter the training set to keep only images with scene background similar to the test image before applying NN-all. We compute the Euclidean distance on the CaffeNet feature [89], and select the filtering threshold using a validation set. (3) *NN-oracle*: We exploit ground-truth action labels

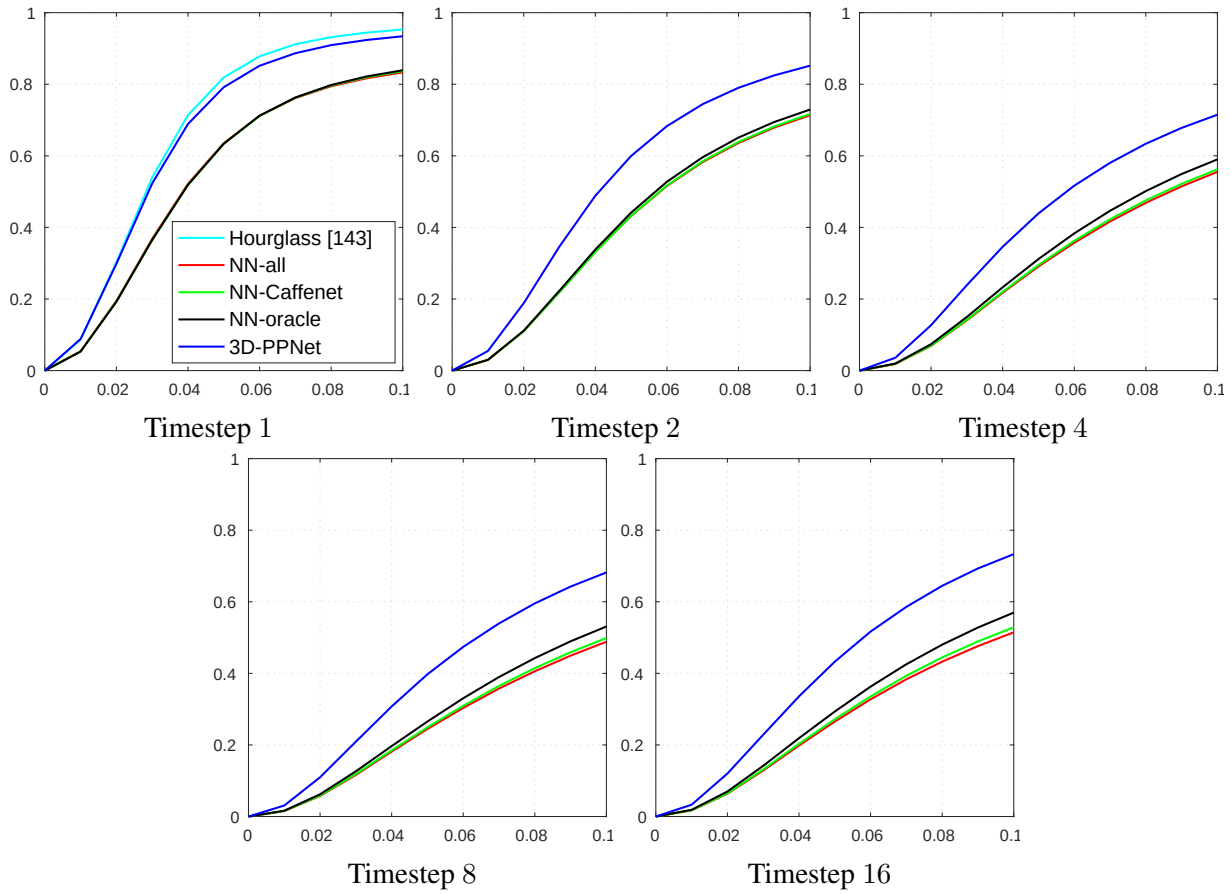


Figure 5.7: PCK curves at different timesteps. The x-axis is the distance threshold and the y-axis is the PCK value. The hourglass network [143] only estimates the current pose in timestep 1. Our 3D-PFNet outperforms all three NN baselines for all timesteps.

to keep only the training images with the same action category as the test image before applying NN-all. Note that this is a strong baseline since our method does not use ground-truth action labels.

5.4.1.5 Results

Fig. 5.7 shows the PCK curves of our approach and the baselines at different timesteps (timestep 1 corresponds to the current frame). For all approaches, the PCK value decreases as timestep increases, since prediction becomes more challenging due to increasing ambiguity as we move further from the current observation. We also report the result of the hourglass network used for our 3D-PFNet. Since the hourglass network can only estimate the current pose, we only show its PCK curve in timestep 1. The three NN baselines achieve similar performance at timestep 1. As the timestep increases, NN-CaffNet gradually outperforms NN-all, verifying our hypothesis that scene contexts can be used to reject irrelevant candidates and improve NN results. Similarly, NN-oracle gradually outperforms NN-CaffeNet, since the ground-truth action labels can improve

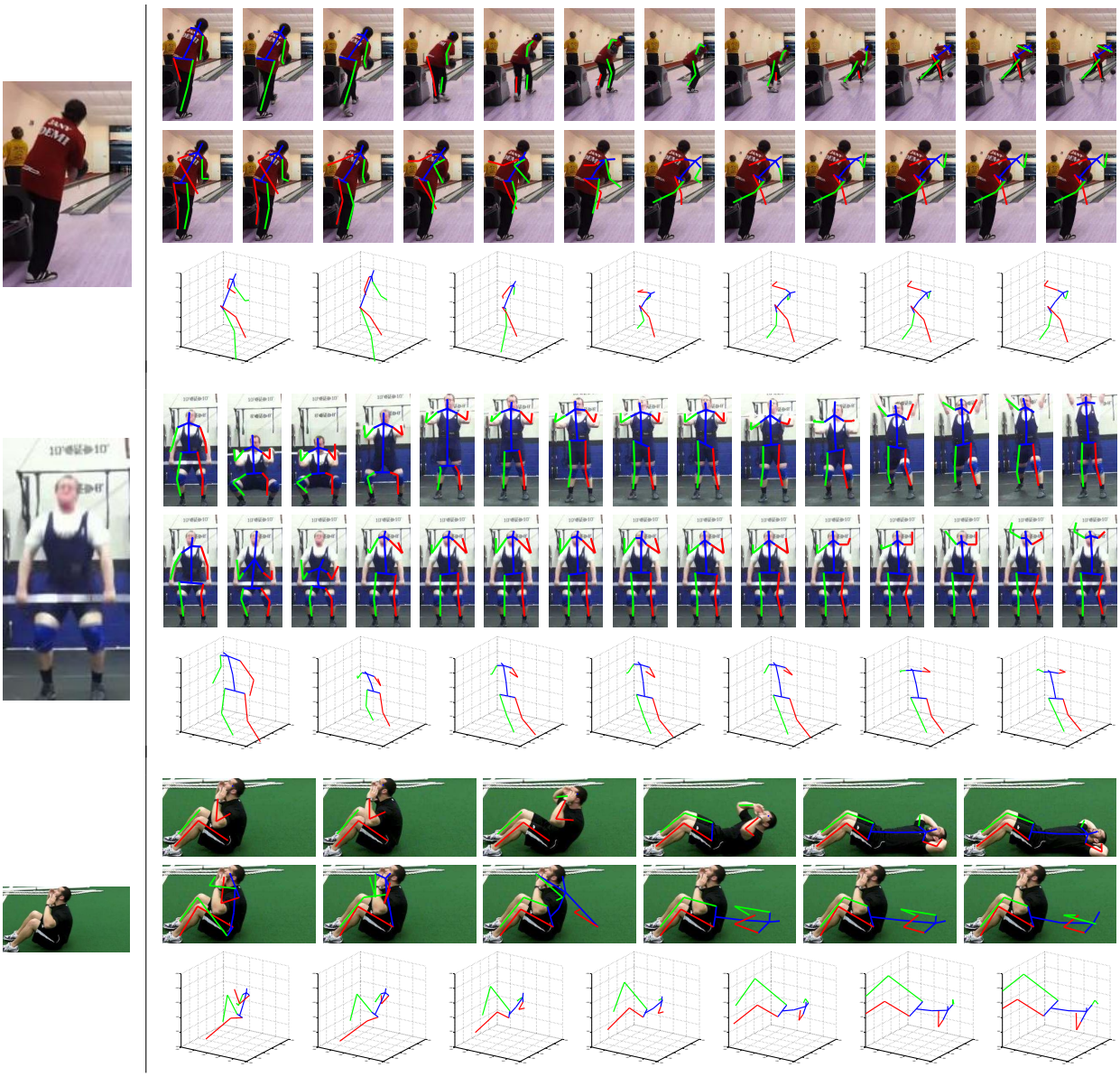


Figure 5.8: Qualitative results of pose forecasting. The left column shows the input images. For each input image, we show in the right column the sequence of ground-truth frame and pose (top) and our forecasted pose sequence in 2D (middle) and 3D (bottom). Note that some keypoints are not shown since they are labeled as invisible in the ground-truth poses.

the candidate set further. Finally, our 3D-PFNet outperforms all three baselines by significant margins. Fig. 5.8 shows qualitative examples of the poses forecasted by our 3D-PFNet. [2] Our 3D-PFNet can predict reasonable pose sequences in both 2D and 3D space. Fig. 5.9 shows failure cases of the NN baselines. The retrieved sequence of NN-all (top) is inconsistent with the context (i.e. a bowling alley) when the NN pose is from a different action class (i.e. baseball swing). By

[2] Also see http://www.umich.edu/~ywchao/image-play/.

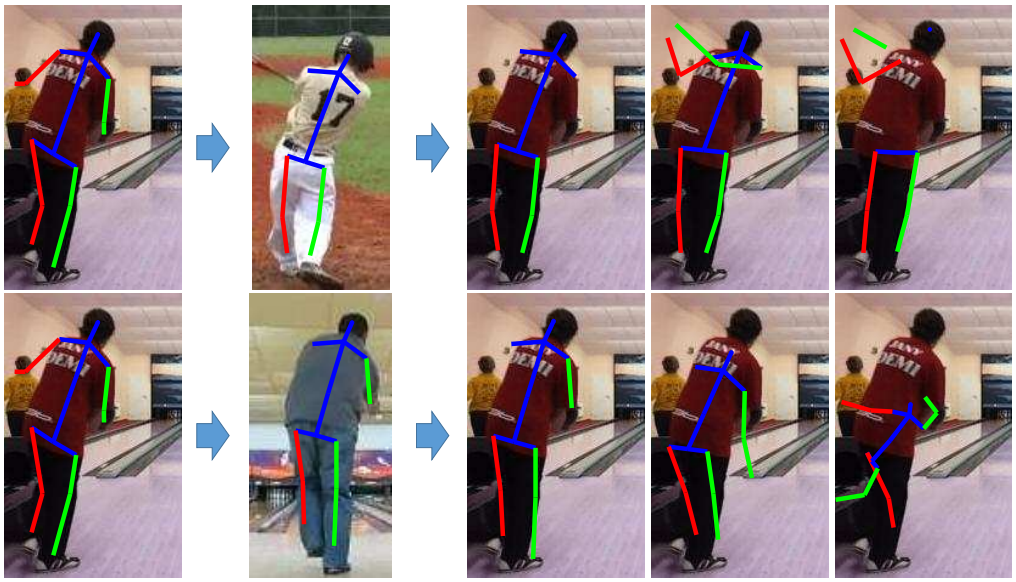


Figure 5.9: Failure cases of the NN baselines. Top: NN-all. Bottom: NN-oracle. Each row shows the input image with the estimated pose, the NN pose in the training set, and the transformed pose sequence of the NN pose on the input image.

Timestep #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Hourglass [143]	**81.9**	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
NN-all	63.5	43.2	33.8	29.1	26.9	25.8	24.8	24.5	24.4	24.5	24.7	25.0	25.5	26.0	26.5	26.5
NN-CaffeNet	63.4	43.3	34.1	29.5	27.3	26.2	25.3	24.9	24.9	25.0	25.3	25.6	26.1	26.7	27.1	27.2
NN-oracle	63.4	44.1	35.5	31.2	29.1	28.0	27.0	26.5	26.5	26.8	27.3	27.6	28.1	28.8	29.2	29.3
3D-PFNet	79.2	**60.0**	**49.0**	**43.9**	**41.5**	**40.3**	**39.8**	**39.7**	**40.1**	**40.5**	**41.1**	**41.6**	**42.3**	**42.9**	**43.2**	**43.3**

Table 5.1: PCK values (%) with threshold 0.5 (PCK@0.05) for timestep 1 to 16.

exploiting ground-truth action labels, NN-oracle (bottom) is able to retrieve a similar pose in the same context. However, the retrieved sequence still fails due to a small error in pose alignment, i.e. the person should be moving slightly toward the right rather than straight ahead. We believe the internal feature representation learned by our 3D-PFNet can better align human poses in the given context and thus can generate more accurate predictions. Tab. 5.1 reports the PCK with threshold 0.05 (PCK@0.05) for all 16 timesteps. Note that all PCK values stop decreasing after timestep 8. This is due to the subset of test sequences with repetitive ending frames, since prediction is easier for those cases as we only need to learn to stop and repeat the last predicted pose.

Tab. 5.2 shows the PCK and the number of training videos of each action class. We see that actions with holistic joint motions (e.g. baseball pitch) are more challenging for pose forecasting and thus have lower PCK values even with more training samples, while actions with only partial joint motions (e.g. jump rope) are the opposite.

Timestep #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	# Tr
Baseball pitch	79.7	51.2	37.4	30.3	26.3	23.6	22.2	21.5	20.8	20.6	20.5	20.7	20.8	20.7	20.6	20.5	94
Baseball swing	81.2	69.0	54.9	46.7	42.3	40.2	39.1	38.7	38.8	38.9	38.7	38.9	39.0	38.8	38.8	38.7	104
Bench press	69.1	60.6	52.6	50.1	48.8	48.7	48.9	49.3	49.9	50.5	51.3	52.1	52.9	53.6	54.1	54.3	63
Bowl	68.8	53.1	41.1	34.9	31.7	30.0	28.9	28.4	27.7	27.3	27.0	26.9	26.9	27.0	26.9	27.0	123
Clean and jerk	87.5	60.1	52.7	47.9	44.6	41.6	39.9	38.5	38.0	37.5	37.1	36.8	36.9	37.0	37.1	37.1	39
Golf swing	82.1	68.7	59.4	54.2	51.6	50.3	49.8	49.3	48.6	47.5	47.3	47.6	48.0	48.0	47.8	47.6	81
Jump rope	83.6	69.4	60.6	61.1	65.4	69.2	65.6	61.9	62.2	64.9	66.1	64.6	64.2	65.6	67.2	67.6	36
Jumping jacks	85.0	63.9	47.1	41.3	40.7	42.9	46.7	50.0	52.6	53.9	55.4	57.9	60.5	62.9	64.9	65.5	51
Pullup	81.4	65.7	50.9	44.3	42.1	42.3	43.4	44.8	46.7	48.8	50.8	52.5	54.4	55.7	56.4	56.5	89
Pushup	73.3	65.5	57.5	53.1	51.4	51.3	51.9	53.2	54.9	56.6	58.4	60.1	61.6	62.7	63.2	63.2	94
Situp	67.1	48.0	41.6	38.9	37.6	37.1	37.4	38.0	39.0	39.6	40.4	41.2	41.8	42.3	42.6	42.8	45
Squat	81.3	58.4	46.1	42.3	40.8	41.1	42.3	43.7	45.5	47.4	49.3	51.2	53.0	54.8	56.0	56.0	104
Strum guitar	62.4	61.6	61.5	61.2	61.1	61.6	61.1	60.7	60.3	60.2	59.7	59.2	58.6	58.5	58.4	58.3	42
Tennis forehand	80.9	59.3	40.8	31.7	27.4	24.7	22.9	22.0	21.0	20.5	20.1	19.9	19.8	19.7	19.7	19.6	73
Tennis serve	78.8	56.4	41.3	34.1	29.5	26.4	24.3	22.8	21.6	20.7	20.3	20.0	20.0	20.3	20.3	20.2	104

Table 5.2: PCK@0.05 of 3D-PFNet on individual action classes.

5.4.2 3D Pose Recovery

We separately evaluate the task of per-frame 3D skeleton recovery from 2D heatmaps on Human3.6M [87].

5.4.2.1 Setup

We use the same data split as in training 3D-PFNet. However, we use video frames and generate heatmaps from hourglass rather than using synthetic data. For evaluation, we construct a validation set of 16150 images by sampling every 40 frames from all sequences and cameras of S5 and S6. Each frame is cropped with the tightest window that encloses the person bounding box while keeping the principal point at image center. We evaluate the predicted 3D keypoint positions relative to their center (i.e. Δ) with mean per joint position error (MPJPE) proposed in [87]. For training, we first fine-tune the hourglass on Human3.6M. We initialize the 3D skeleton converter with weights trained on synthetic data, and further train it with heatmaps from the hourglass.

5.4.2.2 Baselines

We compare our 3D skeleton converter with two top-down approaches: the convex optimization based approach (Convex) proposed by Zhou et al. [238] and SMPLify [10]. Since Convex assumes a weak perspective camera model, it can only estimate keypoint positions relative to their center up to a scaling factor. To generate poses with absolute scale, we first learn a prior on the length of human body limbs using the training data in Human3.6M, and scale their output pose to minimize

	Head	R.Sho	L.Sho	R.Elb	L.Elb	R.Wri	L.Wri
Convex [238]	145.3	123.5	122.8	139.1	129.5	162.2	153.0
SMPLify [10]	132.3	117.4	119.3	149.6	149.5	204.3	192.8
Ours	**72.3**	**64.7**	**63.5**	**93.9**	**88.8**	**135.1**	**124.2**
	R.Hip	L.Hip	R.Kne	L.Kne	R.Ank	L.Ank	Avg
Convex [238]	115.2	111.8	172.1	171.7	257.4	258.5	158.6
SMPLify [10]	140.9	124.0	131.9	135.3	202.3	213.6	154.9
Ours	**59.1**	**57.5**	**75.7**	**76.5**	**113.6**	**113.4**	**87.6**

Table 5.3: Mean per joint position errors (mm) on Human3.6M. Our 3D converter achieves a lower error than the baselines on all joints.

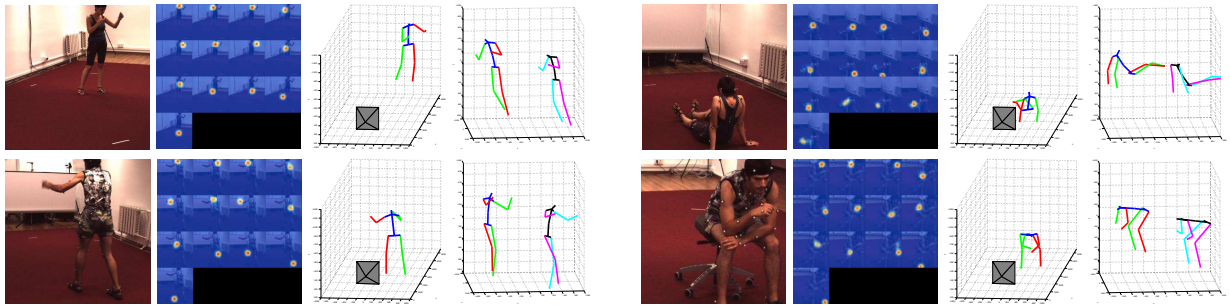


Figure 5.10: Qualitative results of 3D pose recovery. Each sample shows the input image, the heatmaps output of the hourglass, the estimated 3D pose and camera, and a side-by-side comparison with the ground-truth pose (colored by black, cyan, and magenta).

the error between the predicted limb lengths and the prior. Besides, since Convex takes input of 2D keypoint coordinates rather than heatmaps, we sample 2D coordinates for each keypoint by searching for the maximum response in the heatmap. We also re-train the pose dictionary of Convex using the same training set of Human3.6M.

5.4.2.3 Results

Tab. 5.3 shows the comparison of our approach against the baselines on 13 body joints. Our 3D skeleton converter achieves a lower error on all 13 body joints by a significant margin. The improvement over Convex is especially significant on the keypoints of knees and ankles (e.g. for left knee, from 171.7 to 76.5mm, and for left ankle, from 258.5 to 113.4mm). As pointed out in [205], Zhou et al.'s method assumes the input keypoint coordinates to be clean, which is not true for the hourglass output. Our approach, by training on heatmaps, can be adjusted to noisy input. Furthermore, our DNN-based, bottom-up approach, without using any pose priors, enjoys advantages over two top-down baselines, by learning to directly regress the 3D keypoint positions with a sufficiently complex model and a vast amount of training data. We show qualitative examples of our reconstructed 3D poses as well as the estimated camera poses in Fig. 5.10.

5.4.3 Human Character Rendering

We demonstrate one potential application of 3D pose forecasting by rendering human characters from 3D skeletal poses. We use the public code provided by Chen et al. [28]: We first produce a 3D human shape model from each 3D skeletal pose using SCAPE. We then transfer skin and clothing textures to the 3D human model. Finally, the 3D model is rendered and overlaid on the person's projected bounding box in the input image. Fig. 5.11 shows the rendered human characters, both textureless and textured, for the qualitative results shown in the Fig. 5.8. We believe the capability of pose forecasting with 3D human rendering may trigger further applications in augmented reality.

5.4.4 Additional Qualitative Results

We show additional qualitative examples of the forecasted poses in Fig. 5.12, 5.13, and 5.14. Note that the rendered human model also improves the interpretability of the output 3D poses over skeletons. The second example in Fig. 5.14 shows a failure case of 3D pose recovery. While the forecasted motion of the tennis serve looks plausible in 2D (row 2), the recovered 3D poses are unrealistic in their body configurations (row 4 and 5), which may be difficult to perceive in the visualizations of 3D skeletons (row 3). All qualitative results can also be viewed as videos at http://www.umich.edu/~ywchao/image-play/.

5.5 Summary

We presents the first study on forecasting human dynamics from static images. Our proposed 3D Pose Forecasting Network (3D-PFNet) integrates recent advances on single-image human pose estimation and sequence prediction, and further converts the 2D predictions into 3D space. We train the 3D-PFNet using a three-step training strategy to leverage a diverse source of training data, including image and video based human pose datasets and 3D MoCap data. We demonstrate competitive performance of our 3D-PFNet on 2D pose forecasting and 3D pose recovery through quantitative and qualitative results.

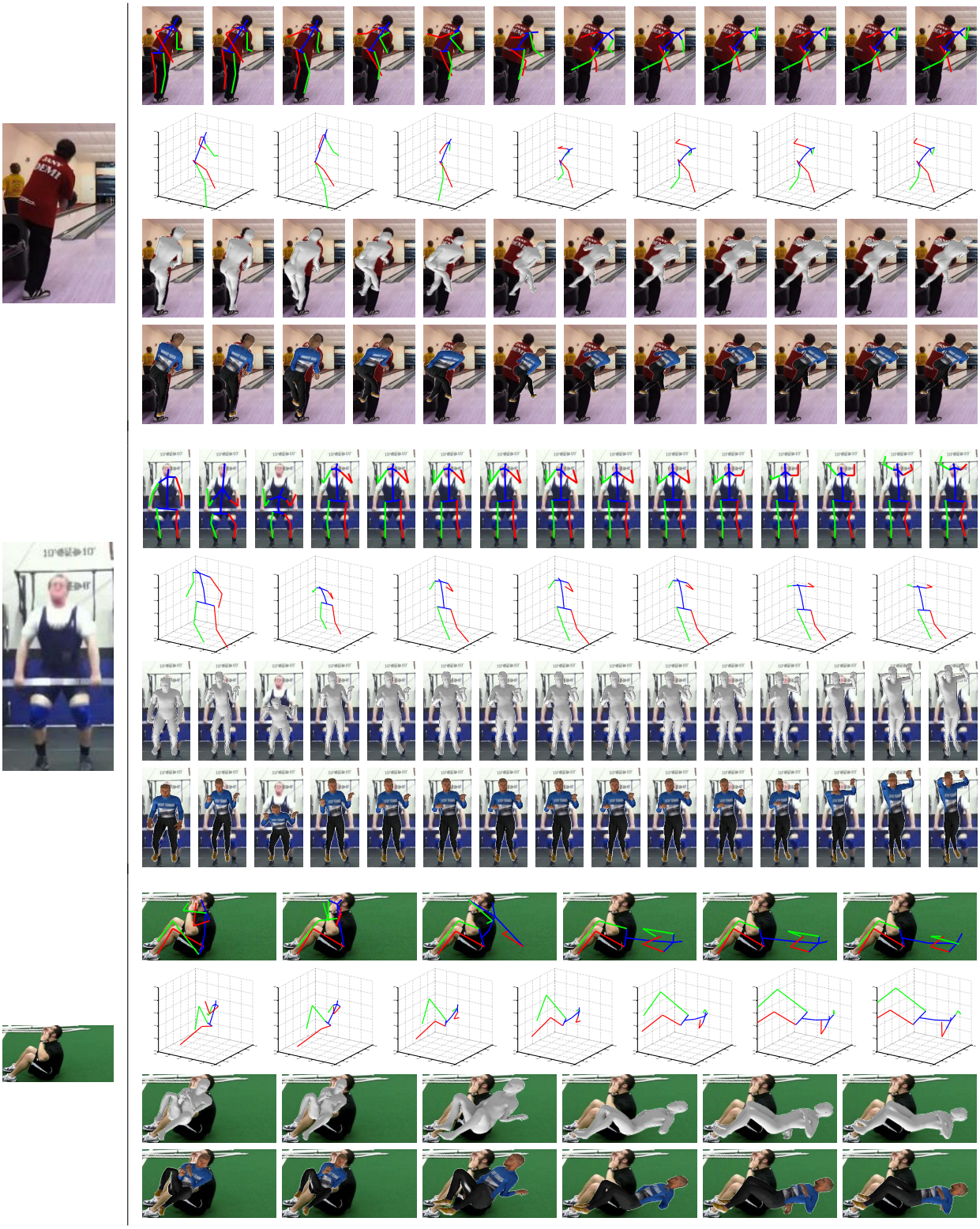


Figure 5.11: Rendering human characters from the forecasted 3D skeletons. The left column shows the input images. For each input image, we show in the right column our forecasted pose sequence in 2D (row 1) and 3D (row 2), and the rendered human body without texture (row 3) and with skin and cloth textures (row 4). We use the rendering code provided by [28].

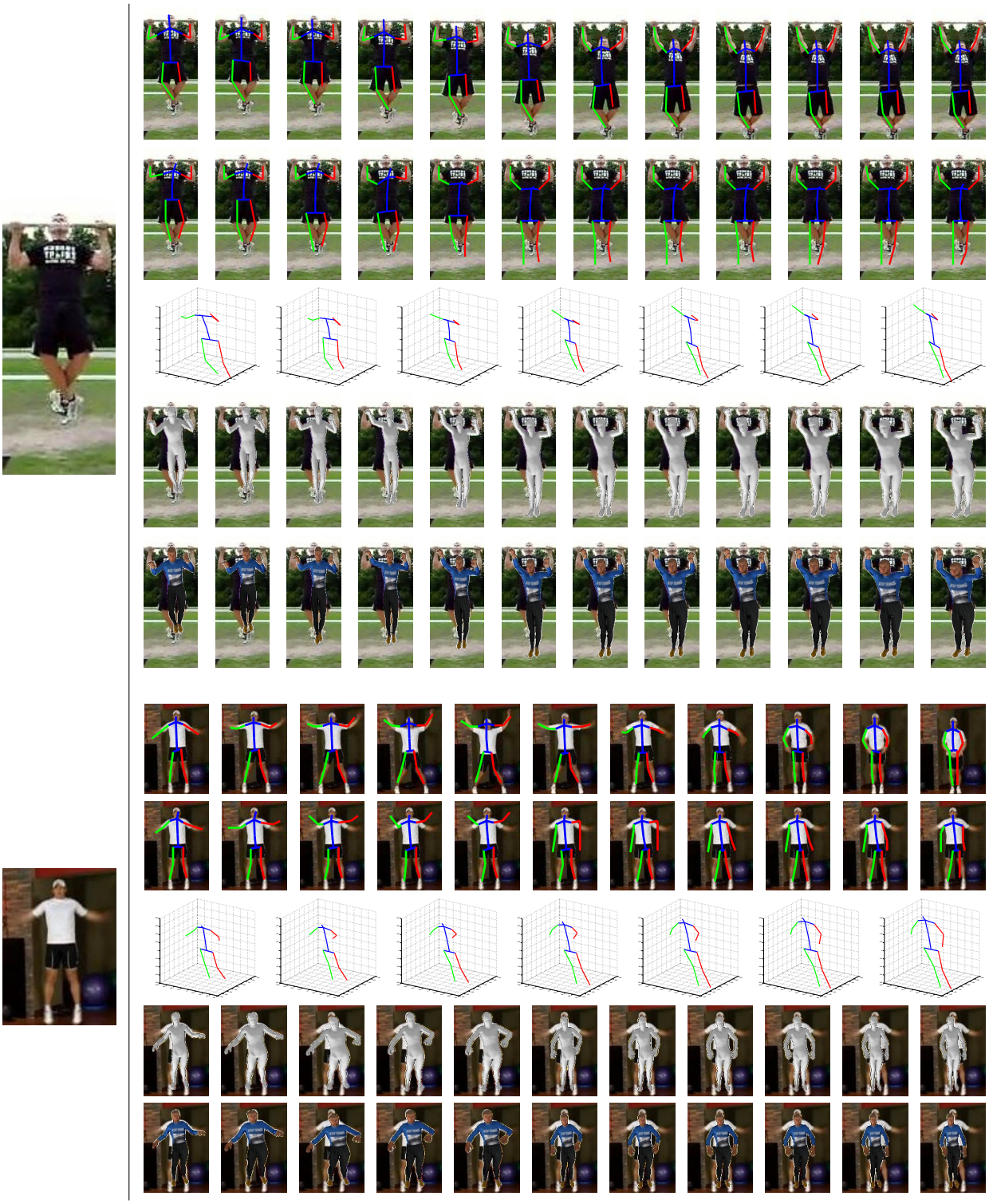


Figure 5.12: Additional qualitative results of pose forecasting. The left column shows the input images. For each input image, we show in the right column the sequence of ground-truth frame and pose (row 1), our forecasted pose sequence in 2D (row 2) and 3D (row 3), and the rendered human body without texture (row 4) and with skin and cloth textures (row 5).

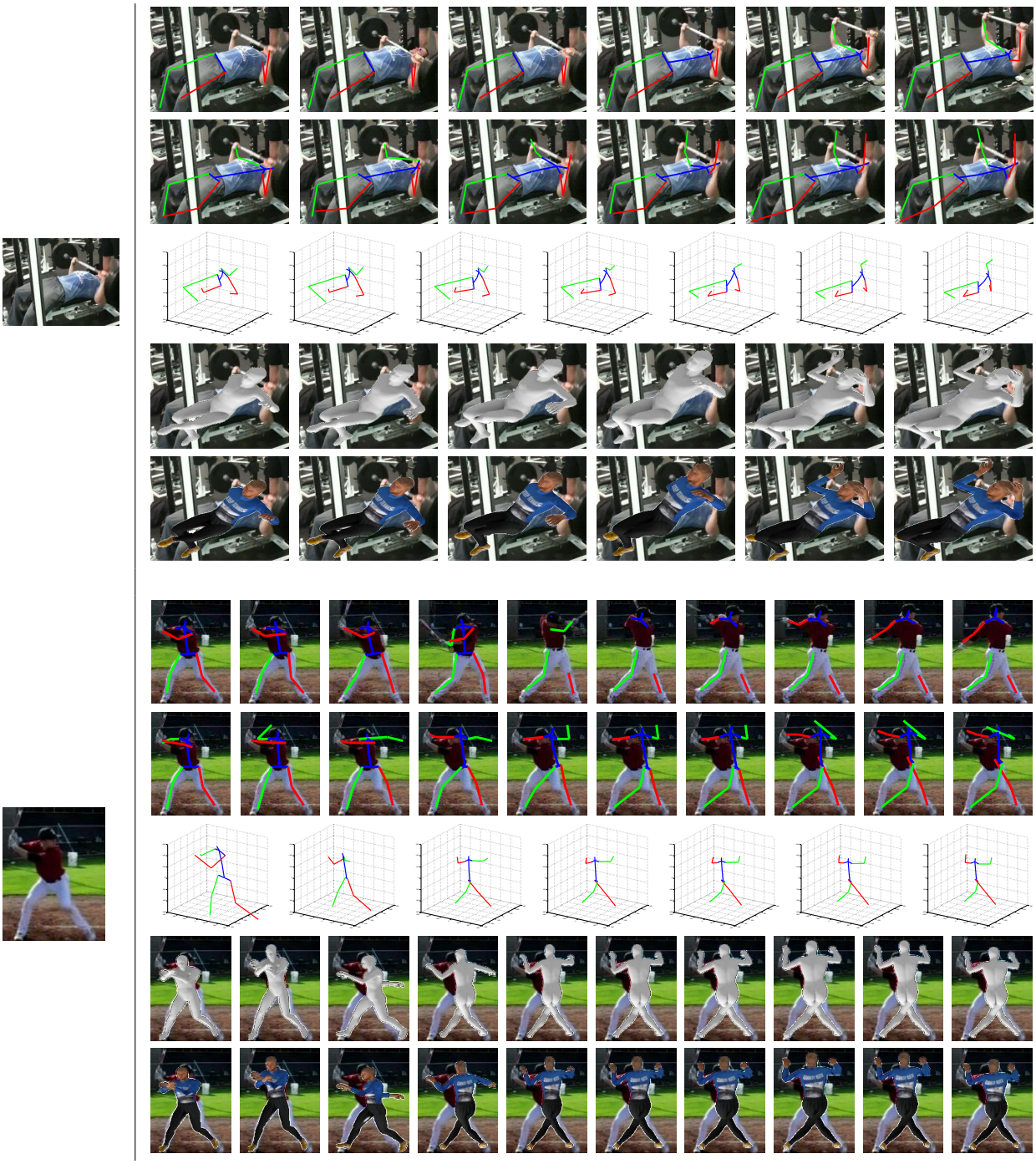


Figure 5.13: Additional qualitative results of pose forecasting. The left column shows the input images. For each input image, we show in the right column the sequence of ground-truth frame and pose (row 1), our forecasted pose sequence in 2D (row 2) and 3D (row 3), and the rendered human body without texture (row 4) and with skin and cloth textures (row 5).

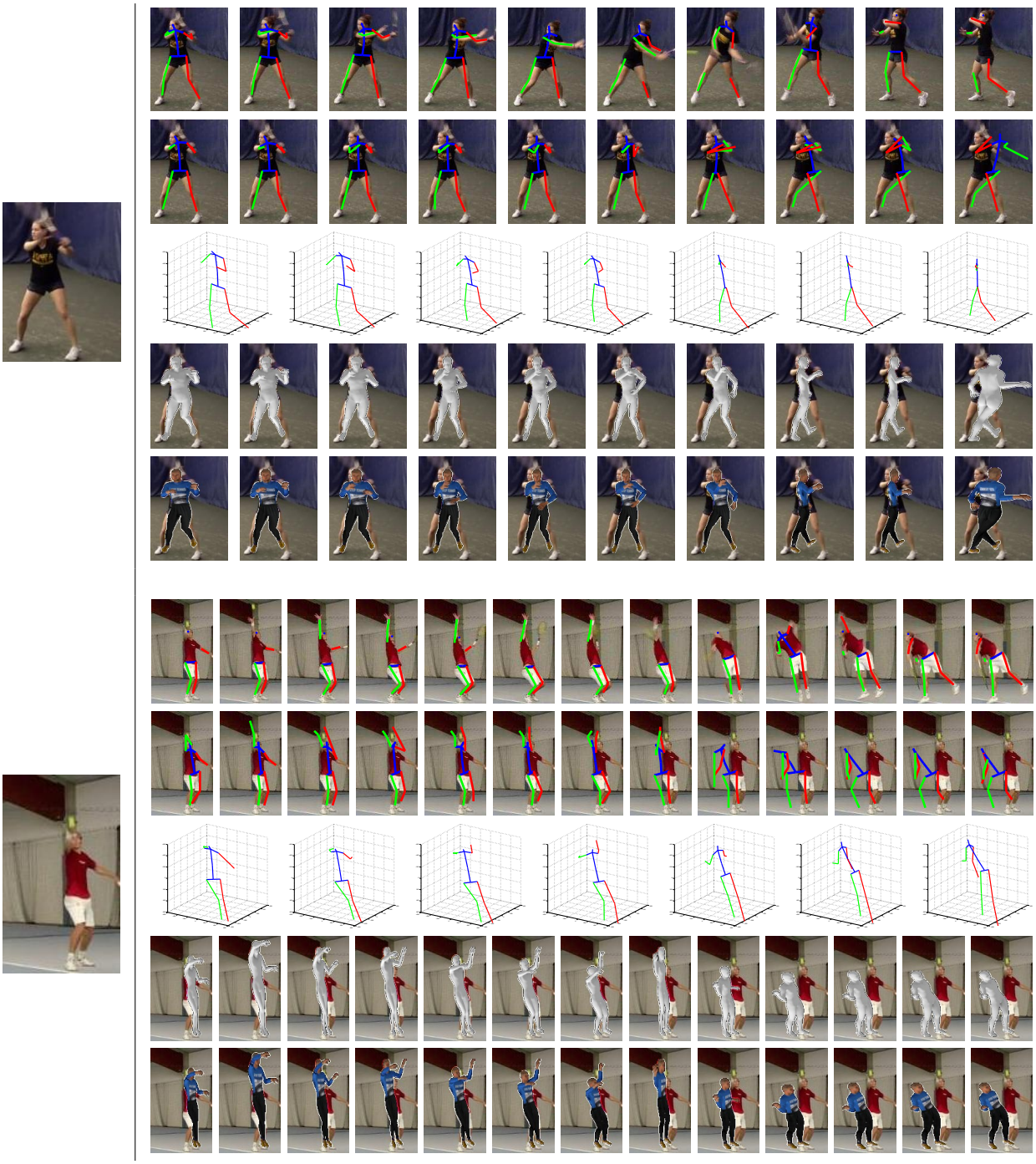


Figure 5.14: Additional qualitative results of pose forecasting. The left column shows the input images. For each input image, we show in the right column the sequence of ground-truth frame and pose (row 1), our forecasted pose sequence in 2D (row 2) and 3D (row 3), and the rendered human body without texture (row 4) and with skin and cloth textures (row 5).

CHAPTER 6

Synthesizing Motion of Human-Object Interactions [1]

6.1 Introduction

The capability of synthesizing (or predicting) human-environment interactions is an important basis for building assistive and socially-aware robots. It enables collision avoidance, e.g. when a robot standing in front of a fridge observes an approaching person, it should immediately move aside by foreseeing that the person will soon reach and open the fridge door. It also expedites the completion of assistive tasks, e.g. when a person is about to sit down at a table, the coffee delivering robot should foresee the sitting posture and navigate to the side of the chair preemptively to hand over the coffee.

Motion capture (mocap) data, which offers high quality recordings of the articulated body pose, has provided a crucial resource for synthesizing human motions. Kinematics based approaches have recently achieved significant progress on problems such as motion synthesis and prediction, due to the marriage of large mocap datasets and deep learning algorithms [60, 88, 80, 63, 14, 133, 79, 240, 117, 70, 71, 210]. However, the lack of physical interpretability in the synthesized motion has been a critical limitation of these approaches. Such limitation becomes significant when it comes to motions that involves substantial human-object or human-human interactions. Due to the lack of ability in modeling the physical interactions, the synthesized motions are often physically unrealistic (e.g. body parts going through physical obstacles or not reacting to collision). This constrains the application of these approaches to mostly non-interactive motions, such as walking and jumping.

Recent progress on physics based character animation in the graphics community has shown impressive breakthroughs [151, 150, 152]. These approaches, by imitating mocap examples through deep reinforcement learning, are able to synthesize realistic motions in a physics simulated environment. Therefore, they achieve a better generalization performance for motions that involve substantial interactions due to the ability of adapting to different physical contexts (e.g. walking

[1] This chapter is based on a joint work with Jimei Yang, Weifeng Chen, and Jia Deng [25].

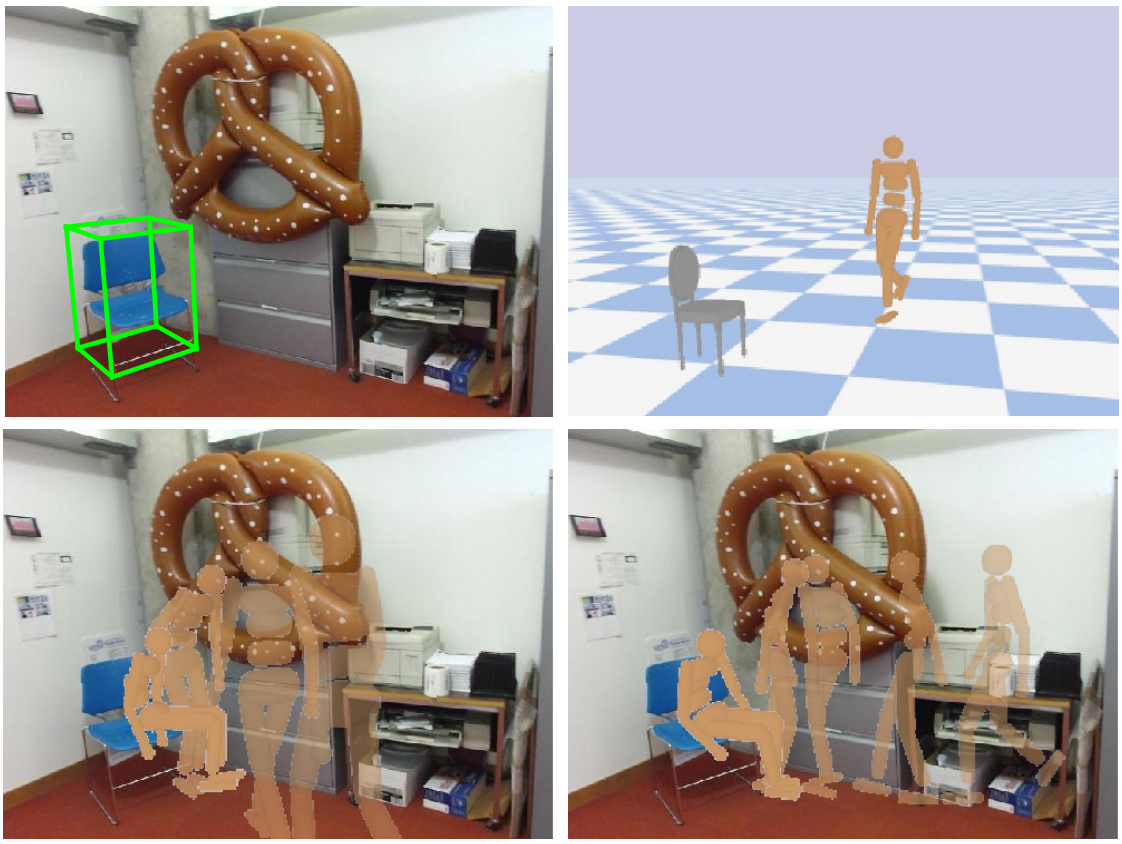


Figure 6.1: Synthesizing the motion of sitting. Top left: an image and a 3D chair detection. Top right: a physics simuliated environment for learning human-chair interactions. Bottom: synthesized sitting motions for the given image.

on uneven terrain and stunt performance under obstacle disturbance). While they can produce realistic motions, a single model is trained for the performance of a single task with a distinct motion pattern (often time from a single mocap clip). Therefore they may fail when it comes to higher-level interactive tasks that require flexible motion patterns. Take the example of a person sitting down on a chair. A person can start in any relative location and orientation with respect to the chair (Fig. 6.1). Imitating a fixed motion sequence (e.g. turn left and sit) may fail to generalize to different human-chair configurations.

In this chapter, we focus on one particular high-level task of sitting onto a chair. As mentioned, there are many possible human-chair configurations and different configurations may require different sequences of actions to achieve the goal. For example, if the human is facing the chair, it needs to walk, turn either left or right and sit; if the human is beyond the chair, it needs to walk, side-walk and sit. We propose a hierarchical reinforcement learning (RL) framework to address the challenge of generalization. Our key idea is the use of hierarchical control: (1) we assume the main task performed by a human (e.g. sit on a chair) can be decomposed into several subtasks (e.g. walk, turn, sit, etc.), where the motion of each subtask can be reliably learned from the mocap data,

and (2) we train a meta controller using RL which can execute the subtasks properly to "complete" the main task from the observed configuration. Such strategy is in line with the observation that humans have a repertoire of motion skills, and different subset of skills is selected and executed when encountering different high-level tasks.

Our contributions are four folds: (1) we extend the prior work on physics based character skill imitation to the context of higher-level interactive tasks—sitting onto a chair; (2) we propose a hierarchical control model that learns a set of subtask controllers, each imitating the motion of a simple, reusable skill, and use a higher-level meta controller to complete the main high-level task by properly executing these subtasks in a sequence; (3) we experimentally demonstrate the strength of the hierarchical approach over single level approaches; (4) we also show at the end of the chapter that our approach can be applied to synthesize motion in living space with the help of 3D scene reconstruction.

6.2 Related Work

6.2.1 Kinematics-based Models

Kinematic modeling of human motions has a substantial body of literature in both vision and graphics domains. Conventional methods such as motion graphs [105] require a large corpus of mocap data and face challenges in generalizing to new behaviors in new context. Recent progress in deep learning enables researchers to explore more efficient algorithms to model human motions, again, from large-scale mocap data. In the vision community, the problem of concern is often motion prediction [60, 88, 63, 14, 133, 240, 117, 70, 71, 210, 189], where a sequence of mocap poses is given as historical observation, and the goal is to predict the upcoming poses. More recent work has even started to predict motions directly from image input [26, 196, 222]. In the graphics community, the focus has been primarily on motion synthesis, which aims to synthesis realistic motions from mocap examples [209, 6, 80, 79]. Regardless of the focus, this class of approaches still face the challenge of generalization due to the lack of physical plausibility in the synthesized motion, e.g. foot sliding and obstacle penetrations.

6.2.2 Physics-based Models

Physics simulated character animation has a long history in computer graphics [127, 125, 151, 126, 150, 32, 152]. Our work is most relevant to the recent work by Peng et al. [151, 150], which achieved impressive results on mocap-based character skill imitation using deep reinforcement learning. They demonstrated robust and realistic looking motions from a virtual character on a wide array of skills including locomotion and acrobatic motions. Notably, they have shown some

progress on high-level tasks (e.g. navigating on irregular terrain [151]) also using a hierarchical model. However, their main task (i.e. locomotion) is simpler, and requires only a single subtask (i.e. walk). We address a more complex high-level task (i.e. sitting onto a chair), and require the execution of a collection of subtasks (i.e. walk, turn, and sit). A very recent work [32] also addresses motion synthesis using hierarchical control, but they focus on a different type of motion (i.e. dressing).

Note that the RL community has also recently witnessed increasing interests in learning humanoid control in physics simulated environments [77, 136]. However, the work in this domain is more geared towards the learning aspects and focuses less on the realisticness of motion.

6.2.3 Hierarchical Reinforcement Learning

Our model is inspired by a series of recent work on hierarchical control in deep reinforcement learning [78, 110, 181]. Although in different contexts, they share the same challenge that the tasks of concern have high-dimensional action space, but can be decomposed into simpler, reusable subtasks. Such decomposition may even help in generalizing to new high-level tasks due to the shared subtasks. Note that Peng et al. [151] also use a hierarchical RL model for character locomotion. However, they consider only one subtask in the lower level, while our main task requires multiple subtasks.

6.2.4 Object Affordances

Our work is also connected to the learning of object affordances in the vision domain. Affordances express the functionality of objects and how humans can interact with them. Previous work has attempted to detect affordances of a scene, represented as a set of plausible human poses, by training large videos corpora [40, 242, 203]. In terms of sitting onto a chair, rather than learning the motion from watching numerous video examples, we learn the motion in a physics simulated environment using limited mocap examples and reinforcement learning. Another interesting work also detects affordances using mocap data [74], but focuses only on static pose rather than motion.

6.3 Overview

Our main task is the following: given a chair model in the 3D space and a skeletal pose of a human, generate a sequence of skeletal poses that describes the motion of the target human sitting down on the chair starting from the given pose (Fig. 6.1). Our system builds upon a physics simulated environment, which contains an articulated structured humanoid and a rigid body chair model. Each joint of the humanoid (except the root) can receive a control signal and produce

dynamics from the physics simulation. The goal is to learn a policy that controls the humanoid to successfully sit on the chair.

Fig. 6.2 (left) illustrates the hierarchical architecture of our policy. At the lower level is a set of subtask controllers, each responsible for generating the control input of executing a particular subtask. As illustrated in Fig. 6.2 (right), we consider four subtasks: *walk*, *left turn*, *right turn*, and *sit*. [2] Since our goal is to synthesize realistic motions, the subtask policies are trained with associated mocap data to imitate real human motions. At the higher level, a meta controller is responsible for controlling the execution of subtasks at each timestep to ultimately accomplish the main task. The subtask controllers and meta controller generate control input at different timescales—60 Hz for the former and 2Hz for the latter. The physics simulation runs at 240 Hz. Each subtask as well as the meta controlling task is formulated as a separate reinforcement learning problem (detailed in Sec. 6.4 and 6.5). We leverage recent progress in deep RL and approximate each policy using a neural network.

6.4 Subtask Controller

A subtask controller is a policy network $\pi(a_t|s_t)$ that maps the state vector s_t to an action a_t at each timestep t. The state representation s is extracted from the current configuration of the simulation environment, and may vary for different subtasks. For example, *turn* requires only proprioceptive information of the humanoid, while *sit* requires not only such information, but also the spatial configuration of the chair with respect to the humanoid. For all subtasks, the action a is the control signal for the humanoid joints. We use a humanoid model with 21 degree of freedom, i.e. $a \in \mathbb{R}^{21}$. The network architecture is fixed for all the subtask policies: we use a multi-layer perceptron with two hidden layers of size 64. The output of the network parameterizes the probability distribution of a, which is modeled by a Gaussian distribution with a fixed diagonal covariance matrix, i.e. $\pi(a|s) = \mathcal{N}(\mu(s), \Sigma)$ and $\Sigma = diag(\{\sigma_i\})$. We can then generate a_t at each timestep by sampling from $\pi(a_t|s_t)$.

Each subtask is formulated as an independent RL problem. At timestep t, the state s_t extracted from the simulation environment is given to the policy network and output an action a_t. The action a_t is then fed back to the simulation environment, which then generates the state s_{t+1} at the next timestep as well as a reward signal r_t. The design of the reward function is crucial and plays a key role in shaping the style of humanoid's motion. A heuristically crafted reward may yield a task achieving policy, but may result in unnatural looking motions and behaviors [77]. Inspired by a recent work on mocap-based character skill imitation [150], we formulate each subtask reward

[2] Here we consider 180 degree in-place *turns*, which should be distinguished from moderate angled steering during walking. The *sit* subtask is also an in-place motion and should be distinguished from the main sitting task that involves locomotion.

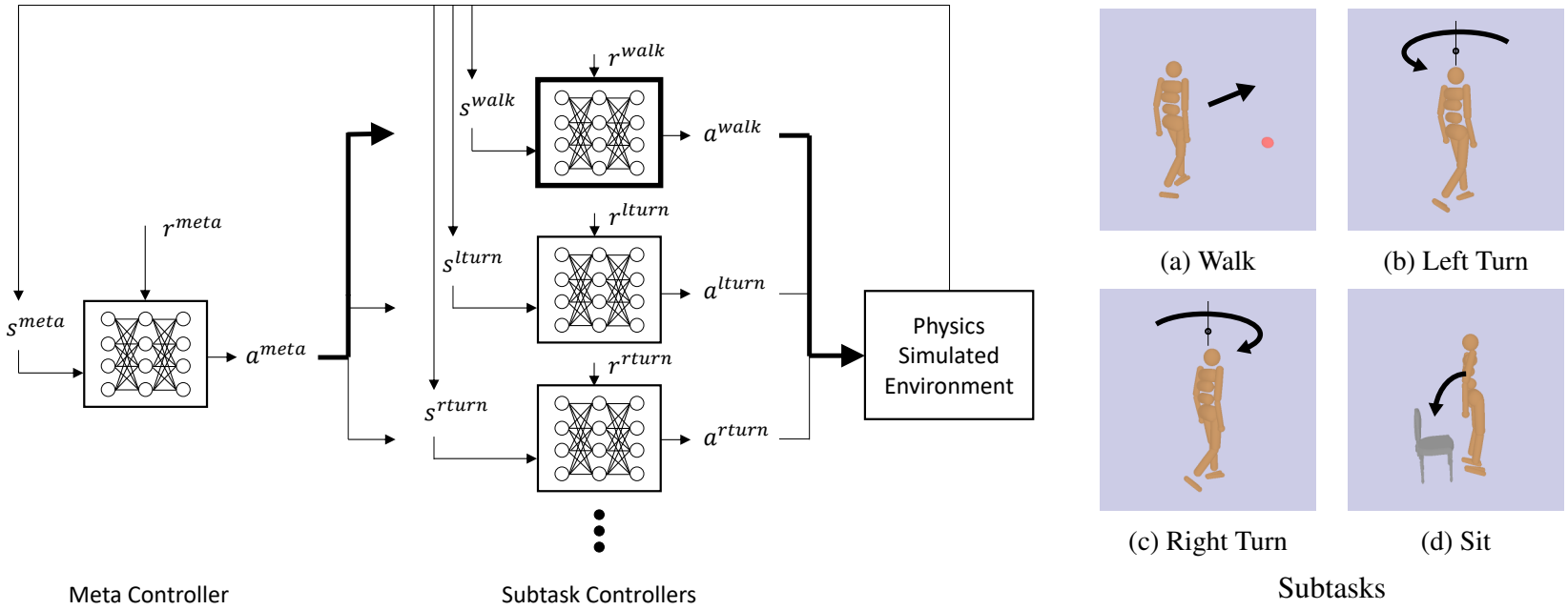


Figure 6.2: Left: Overview of the hierarchical system. Right: Illustration of the subtasks.

function by a sum of two terms that simultaneously encourages the imitation of the mocap-based reference motion and the achievement of task objectives:

$$r^{sub} = r^S + r^G. \tag{6.1}$$

Here r^S and r^G account for the similarly to the reference motion and the achievement of subtask goals, respectively. We use a consistent similarity reward r^S over all subtasks:

$$r^S = \omega^p r^p + \omega^v r^v, \tag{6.2}$$

where r^p and r^v encourage the similarity of local joint angles q_j and velocities $\dot{q}_j$ between the humanoid and the reference motion, and ω^p and ω^v are the respective weights. Specifically,

$$\begin{aligned} r^p &= \exp\left(-\alpha^p \sum_j (d(q_j, \hat{q}_j)^2\right) \\ r^v &= \exp\left(-\alpha^v \sum_j (\dot{q}_j - \hat{\dot{q}}_j)^2\right). \end{aligned} \tag{6.3}$$

Note that $d(\cdot, \cdot)$ computes the angular difference between two angles. We empirically set $\omega^p = 0.5$, $\omega^v = 0.05$, $\alpha^p = 1$, and $\alpha^v = 10$.

Next we detail the state representation s and task objective reward r^G for each of the four subtasks.

6.4.1 Walk

The state $s^{walk} \in \mathbb{R}^{52}$ consists of a 50-d proprioceptive feature and a 2-d goal feature that specifies an intermediate walking target. The proprioceptive feature includes the local joint angles and velocities, the height and linear velocity of the root (torso in our humanoid) as well as its pitch and roll angles, and a 2-d binary vector indicating the contact of each foot with the ground (see Fig. 6.3 for illustration). Instead of walking in random directions, target-directed locomotion [6] is crucial for accomplishing high-level tasks. Assume a target is given, represented by a 2D point on the ground plane (e.g. the red dot in Fig. 6.3), the 2-d goal feature is given by $[\sin(\psi), \cos(\psi)]^\top$, where ψ is the azimuth angle to the target in the humanoid centric coordinates. The generation of targets will be detailed in the meta controller section.

We observe that directly training a target directed walking policy with mocap samples can be challenging. Therefore we adopt a two-step training strategy with distinct task objective rewards. In the first step, we encourage similar steering patterns to the reference motion, i.e. the linear

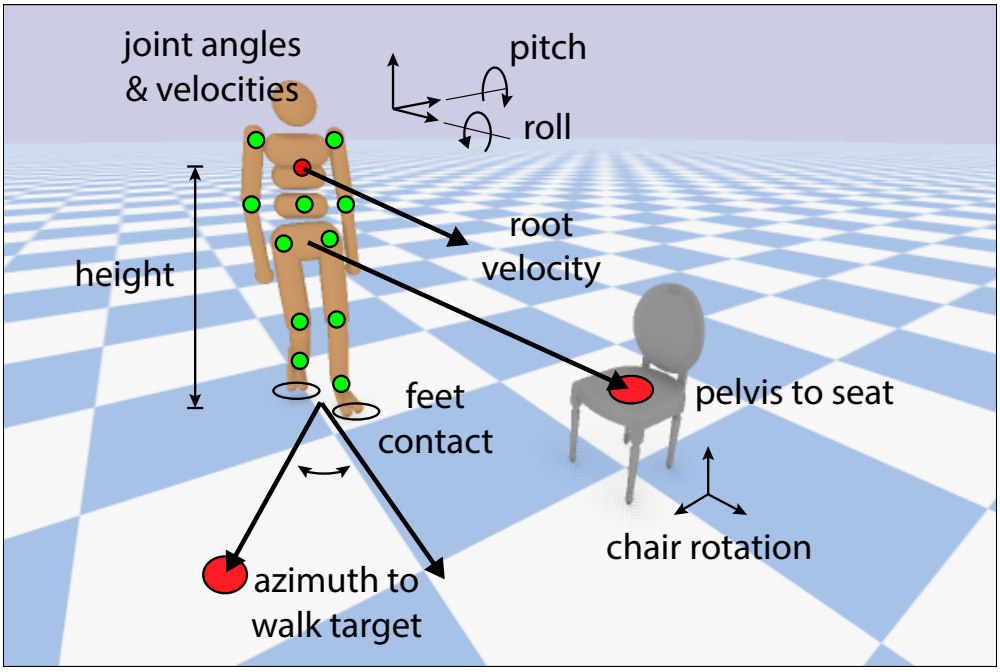


Figure 6.3: Humanoid and chair state representation. The red dot on the humanoid denotes the root, and the greed dots denote the non-root joints. The red dot on the ground denotes the walk target, while the one on the chair denotes the center of the seat surface.

velocity of the root $v \in \mathbb{R}^3$ should be similar between the humanoid and reference motion:

$$r^G = 0.5 \cdot \exp\left(-10 \cdot \sum_i (v_i - \hat{v}_i)^2\right). \tag{6.4}$$

In the second step, we encourage the progress of moving towards the target:

$$r^G = 0.1 \cdot \frac{1}{1 + \exp(10 \cdot V^{walk})}, \tag{6.5}$$

where $V^{walk} = (D^{walk}_{t+1} - D^{walk}_t)/\delta t$, D^{walk}_t denotes the horizontal distance between the humanoid's root and the target at timestep t, and δt is the length of the timestep.

6.4.2 Left/Right Turn

The state s^{lturn}, $s^{rturn} \in \mathbb{R}^{52}$ share the same representation containing the 50-d proprioceptive feature used in the walk subtask. The task objective reward encourages the rotation of the root to be matched between the humanoid and reference motion:

$$r^G = 0.1 \cdot \exp\left(-10 \cdot \sum_i d(\theta_i, \hat{\theta}_i)^2\right), \tag{6.6}$$

where $\theta \in \mathbb{R}^3$ consists of the pitch, yaw, and roll angles of the root.

6.4.3 Sit

The sit subtask assumes that the humanoid sets out by standing roughly in the front area of the chair and facing away from the chair. The goal is simply to lower the body and be seated. Different from walk and turn, the state for sit should capture the spatial information of the chair. Our state $s^{sit} \in \mathbb{R}^{57}$ consists of the same 50-d proprioceptive feature used in walk and turn, as well as a 7-d feature describing the state of the chair in the humanoid centric coordinates. The 7-d chair state includes the displacement vector from the pelvis to the center of the seat surface, and the rotation of the chair in the humanoid centric coordinates, represented as a quaternion (see Fig. 6.3 for illustration). The task objective reward encourages the pelvis to move towards the center of the seat surface:

$$r^G = 0.5 \cdot (-V^{sit)}, \tag{6.7}$$

where $V^{sit} = (D^{sit}_{t+1} - D^{sit}_t)/\delta t$ and D^{sit}_t is the 3D distance between the pelvis and the center of the chair's seat surface.

6.5 Meta Controller

The meta controller is a policy network with the same architecture as the subtask controllers. Since the goal now is to navigate the humanoid to successfully sit on the chair, the input state s^{meta} should also encode the spatial information of the chair. We use the same 57-d state representation from the sit subtask, which contains both proprioceptive and chair information. Different from the subtask controllers, where the output action directly controls the humanoid joints, the output action a^{meta} controls the execution of subtasks. Specifically, $a^{meta} = \{a^{switch}, a^{target}\}$ consists of two components. $a^{switch} \in \{\textit{walk}, \textit{left turn}, \textit{right turn}, \textit{sit}\}$ is a discrete output which at each timestep picks a single subtask out of four to execute. $a^{target} \in \mathbb{R}^2$ specifies the 2D target for the walk subtask, which is used to compute the goal state in s^{walk}. Note that a^{target} is only used when the walk subtask is picked for execution. The output of the policy network parameterizes the probability distributions of both a^{switch} and a^{target}, where a^{switch} is modeled by a categorical distribution as standard classification problems and a^{target} is modeled by a Gaussian distribution following the subtask actions.

The meta controlling task is also formulated as an independent RL problem. At timestep t, the policy network takes the state s^{meta}_t provided by the simulation environment and output an action a^{meta}_t. a^{meta}_t then triggers one specific subtask controller to generate the control signals for the humanoid. The simulation environment takes the control signal and returns the state s^{meta}_{t+1}

at the next timestep and also a reward r_t^{meta}. In contrast to the subtask reward, which accounts for both the similarity to the reference motion and task objectives, the reward for meta controller should only be providing feedback on the main task. We adopt a reward function that encourages a specific body part to move towards and be in contact with a specific physical object:

$$r^{meta} = \begin{cases} 1 & \text{if } z_{\text{contact}} = 1 \\ 0.5 \cdot (-V^{sit}) & \text{otherwise.} \end{cases} \tag{6.8}$$

For our main task, z_{contact} determines if the pelvis is in physical contact with the seat surface, which can be detected by the physics simulator, and V^{sit} is defined as in the reward of the sit subtask.

6.6 Training

Since the subtasks and the meta controlling task are formulated as independent RL problems, they can be individually trained using standard RL algorithms. The full training pipeline is divided into two stages: (1) training each subtask controllers separately and (2) training the meta controller given the trained subtask controllers. All controllers are trained in a standard actor-critic framework using the proximal policy optimization (PPO) algorithm [165].

6.6.1 Subtask Controller

The training of the subtask controllers is also divided into two stages. First, we start each episode by initializing the pose of the humanoid to the first frame of the reference motion, and train the humanoid to complete the subtask by imitating the following frames. We apply the early termination strategy [150]: we terminate the episode immediately if the height of the root falls below 0.78 meters for *walk* and *turn*, and 0.54 meters for *sit*. These thresholds are chosen according to the height of the humanoid. For *turn*, the episode is also terminated when the root yaw angle differs from the reference motion for more than $45°$. For *walk*, we adopt the two-step training strategy mentioned in Sec. 6.4. In target-directed walking, we randomly sample a new 2D target in the frontal region of the humanoid every 2.5 seconds or when the target is reached. For *sit*, the chair is placed at a fixed location behind the humanoid, and we use reference state initialization [150] to facilitate training.

The training above enables the humanoid to perform subtasks from the initial pose of the reference motion, but does not guarantee successful transitions between subtasks (e.g. *walk*→*turn*), which is critical for the main task. Therefore in the second stage, we fine-tune the controllers by setting the initial pose to a sampled ending pose of another subtask, similar to the policy sequencing method in [32]. For *turn* and *sit*, we sample the initialize pose from the ending pose of *walk*

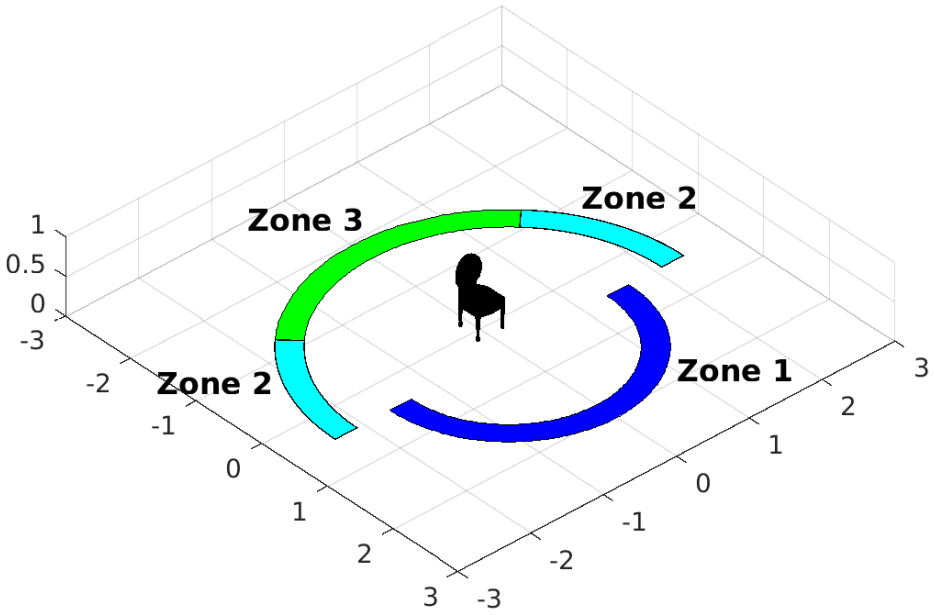


Figure 6.4: Curriculum learning for the meta controller. Training is started from easier states (Zone 1), and then moved to more challenging states (Zone 2 and 3).

and *turn*, respectively.

6.6.2 Meta Controller

Our goal is to enable the humanoid to sit down regardless of where it starts in the environment. Note that the difficulty of this task is highly dependent on the initial state: when the humanoid is already facing the seat, it only needs to turn and sit, while if it is standing behind the chair, it needs to walk around the chair first and then sit down. Training can be challenging if starting from a difficult state, since the humanoid needs to execute a long sequence of correct actions by chance to receive the final reward for sitting down. To facilitate training, we propose a multi-step training strategy inspired by curriculum learning [230]. The idea is to begin the training from easier states, and progressively increase the task difficulty when the training converges. As illustrated in Fig. 6.4, we begin by initializing the humanoid only in the frontal side of the chair (Zone 1). Once trained, we change the initial position to the lateral sides (Zone 2). And finally, we start the humanoid from the rear side (Zone 3).

Subtask	Subject #	Trial #
Walk	8	1, 4
L/R Turn	69	13
Sit	143	18

Table 6.1: Mocap clips adopted from the CMU database [3].

	Subtasks	Meta Task
`nsteps`	8192	64
`nminibatches`	32	8
`noptepochs`	4	2
`lr`	1×10^{-4}	1×10^{-4}

Table 6.2: Hyperparamters for PPO training.

6.7 Results

6.7.1 Reference Motion

We obtain mocap examples from the CMU Graphics Lab Motion Capture Database [3]. Tab. 6.1 shows the mocap clips we used for each subtask. To obtain the reference motion, we extract the relevant motion segments in the mocap clips and retarget the motion to our humanoid model. We use a 21-DoF humanoid model provided by the Bullet Physics SDK [2]. Motion retargeting is performed using a Jacobian-based inverse kinematics method developed by [80].

6.7.2 Implementation Details

Our simulation environment is based on OpenAI Roboschool [165, 4], which uses the Bullet physics engine [2]. We use a randomly selected chair model from ShapeNet [20]. The PPO algorithm for training the policy networks is based on the implementation in OpenAI Baselines [46]. Tab. 6.2 shows the hyerparamters we used for training. Note that the training of the subtask controllers is challenging and can take up to several days due to the high dimensional (i.e. 21-DoF) action space.

6.7.3 Subtask

First we show qualitative results of the individual subtask controllers trained using their corresponding reference motions. Each row in Fig. 6.5 shows the humanoid performance of one particular subtask. The humanoid can successfully walk in one direction (row 1), following a target (row 2), turn in place both leftward (row 3) and rightward (row 4), and sit on a chair (row 5).

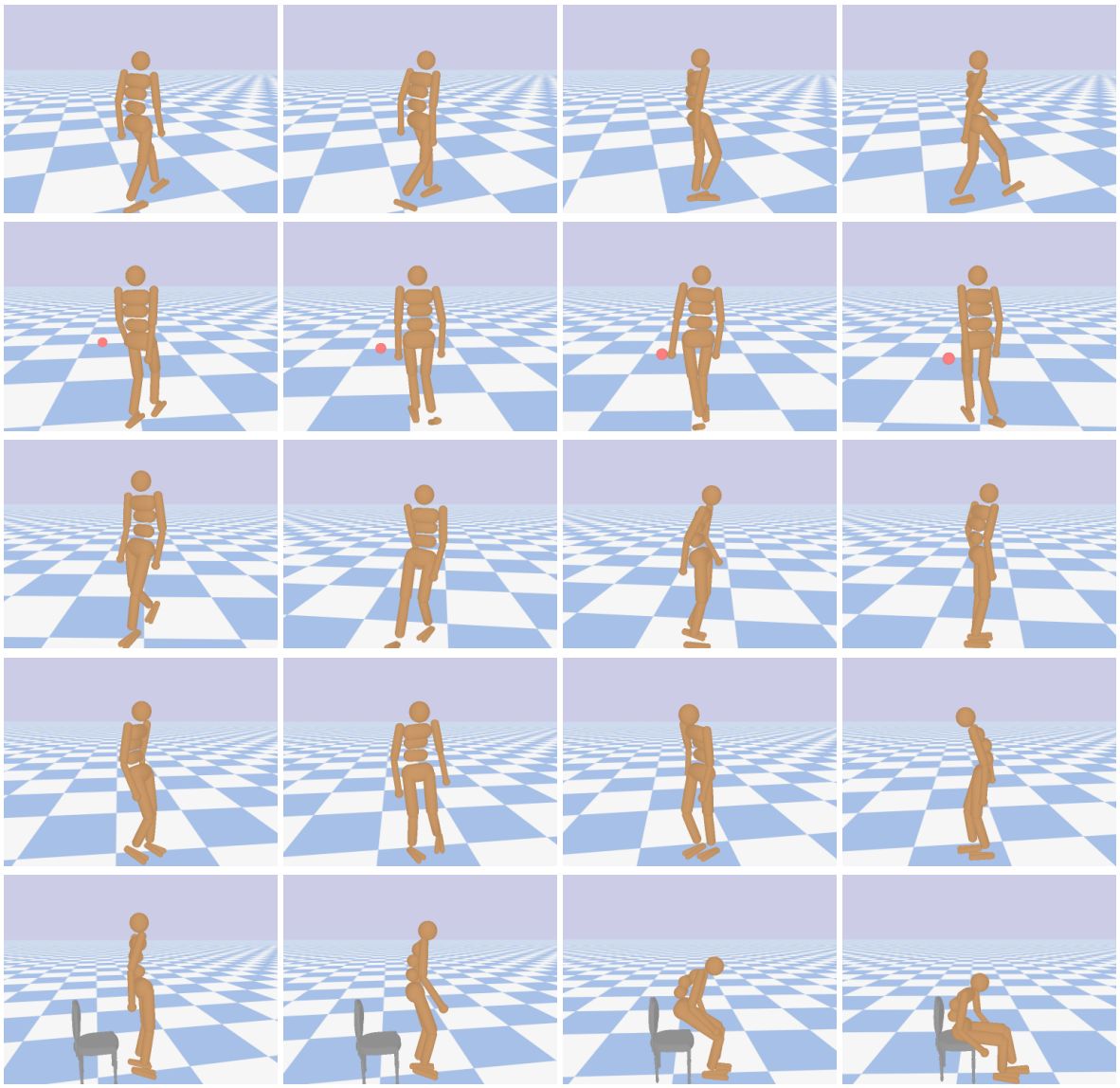


Figure 6.5: The humanoid trained for each subtasks. From top to bottom row: forward walking, target directed walking, left turn, right turn, and sit.

6.7.4 Evaluation of Main Task

Given the trained subtask controllers, we next evaluate our main task. We consider two metrics for quantitative evaluation: (1) *success rate* and (2) *mean minimum distance*. The success rate requires a definition of the success of the task. We declare a success whenever the pelvis of the humanoid has been continuously in physical contact with the seat surface for the past three seconds. We report the success rate over 1000 trials by initializing the humanoid at random positions with respect to the chair. The success rate evaluates the task completion with a hard constraint and does not reveal the progress when the humanoid fails. Therefore we also evaluate with the per-trial minimum distance (in meters) between the pelvis and the center of the seat surface and report the

	Succ Rate (%)	Mean Min Dist (m)
Kinematics	–	1.268 ± 0.492
Physics	0.0	1.175 ± 0.233
Ours	**31.4**	$\mathbf{0.330 \pm 0.231}$

Table 6.3: Comparison between our hierarchical approach and non-hierarchical baselines in the Easy setting.

mean over the 1000 trials.

We observe that the task can be extremely challenging when the initial position of the humanoid is unconstrained (i.e. can start from any position with respect to the chair). To better analyze the performance, we consider two different settings of initialization: (1) *Easy* and (2) *Hard*. In the Easy setting, the humanoid is initialized from roughly 2 meters away on the frontal half-plane of the chair (i.e. Zone 1 in Fig. 6.4), with an orientation roughly towards the chair. The task is expected to be completed by simply walking forward, turning around, and sitting down. In the Hard setting, we initialize the humanoid again from roughly 2 meters away but in the lateral and rear sides of the chair (i.e. Zone 2 and 3 in Fig. 6.4). The humanoid needs to walk around the chair to its front side to sit down successfully.

6.7.5 Easy Setting

We compare with two non-hierarchical (i.e. single-level) baselines in the Easy setting. The first one is a *kinematics* based approach: we select a mocap clip containing a holistic motion sequence that consecutively performs walking, turning, and sitting on a chair. We extract the motion sequence and retarget to the humanoid model. When a trial begins, we align the motion sequence to the humanoid's orientation by aligning the yaw angle of the root. We then use the following frames of the motion sequence as the kinematic trajectory of the trial. Note that this method is purely kinematic based and cannot model any physical interactions (e.g. contact) between the humanoid and the chair. The second baseline extends the first one to a *physics* based approach: we use the extracted motion sequence as reference and train a single controller to imitate the motion. This is similar to training the subtask controller except now the subtask is holistic (i.e. containing walk, turn, and sit in one reference motion). Both baselines are considered non-hierarchical as neither performs task decomposition.

Tab. 6.3 shows the quantitative comparison of our approach with the baselines. The success rate is not provided for the kinematics baseline since we cannot detect physical contact between the humanoid and the chair. However, its 1.268 mean minimum distance suggests that the humanoid on average remains far from the chair throughout the trials. For the physics baseline, we observe a similar mean minimum distance (i.e. 1.175), and the 0% success rate is not surprising given that

	Succ Rate (%)	Mean Min Dist (m)
Zone 1	31.4	0.330 ± 0.231
Zone 2	10.7	0.521 ± 0.322
Zone 3 w/o CL	4.7	0.504 ± 0.233
Zone 3 w/ CL	5.7	0.504 ± 0.244

Table 6.4: Comparison of the Easy and Hard settings. The proposed curriculum learning strategy improves the training outcome.

the humanoid cannot even get close to the chair. Qualitative results are shown in Fig. 6.6. We can see that for the kinematics baseline (row 1), the behavior of the humanoid does not even follow physics rules (i.e sitting in air). For the physics baseline (row 2), although following physics rules (i.e. falling on the ground eventually), the humanoid still fails in approaching the chair. These holistic baselines do not perform well since they simply imitate the mocap example and repeat the same motion pattern regardless of their starting position in space.

Our approach performs significantly better on both evaluation metrics, suggesting that our hierarchical model, by breaking motions into reusable subtasks and learning to execute them, can achieve better generalization. As shown in row 3 and 4 of Fig. 6.6, our method can pick different subtasks in difference scenarios, e.g. *walk*→*right turn*→*sit* when approaching from the south side (row 3), and *walk*→*left turn*→*sit* when approaching from the north side (row 4). Row 5 shows a failure case, where the humanoid is stuck in the state of directly confronting the chair. We note that building a more diverse skill set for the subtasks (e.g. backward walk) might help resolving cases like this, and is an important future direction.

6.7.6 Hard Setting

We now increase the task difficulty by initializing the humanoid in the lateral and rear sides of the chair (i.e. Zone 2 and 3 in Fig. 6.4). Tab. 6.4 compares the success rate and mean minimum distance of different initialization zones. By moving the initial position from Zone 1 to Zone 2, we observe a significant drop in the success rate (i.e. from 31.4% to 10.7%) and an increase in the mean minimum distance (i.e. from 0.330 to 0.521). The success rate drops further when we move the initial position to Zone 3. However, compared to training from scratch, we observe a slightly higher success rate when we adopt the proposed curriculum learning strategy (i.e. 5.7% for w/ CL versus 4.7% for w/o CL). This suggests that a carefully tailored curriculum strategy can improve the training outcome of a challenging task. Fig. 6.7 shows two successful examples of the humanoids starting from the back side of the chair. The humanoid manages to navigate to the front of the chair and sit down. Interestingly, the humanoid learns a slightly different motion strategy (e.g. *walk*→*sit* without *turn*) compared to starting from the frontal side (row 3 and 4 in Fig. 6.6).

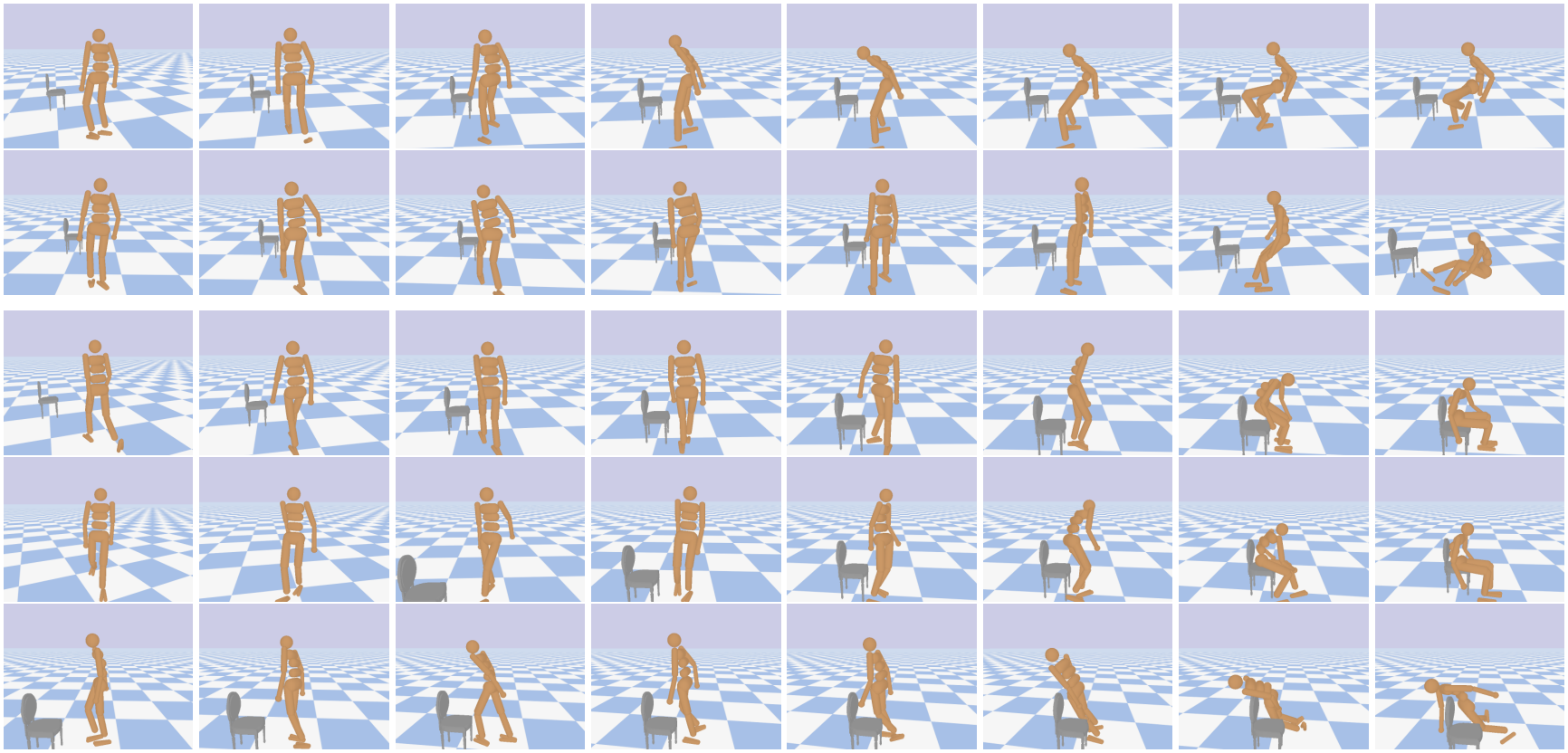


Figure 6.6: Qualitative comparison of our approach and the baselines. Ro w 1 and 2 show failure cases from the kinematics and physics baselines. The fo rmer violates physics rules (i.e. sitting in air), and both do not generalize to new human-chair spatial configurations. Row 3 to 5 show successful (3 and 4) an d failure (5) cases of our approach.

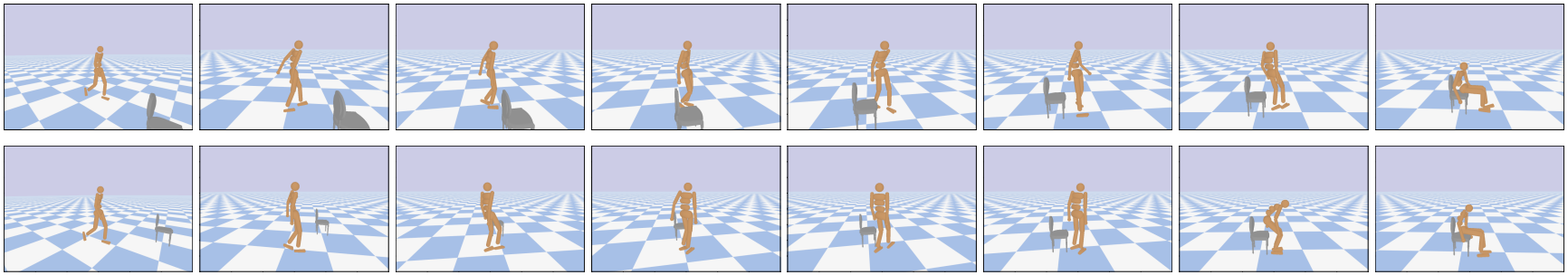


Figure 6.7: Qualitative results from the Hard setting. The humanoid can successfully sit down when starting from the back side of the chair.

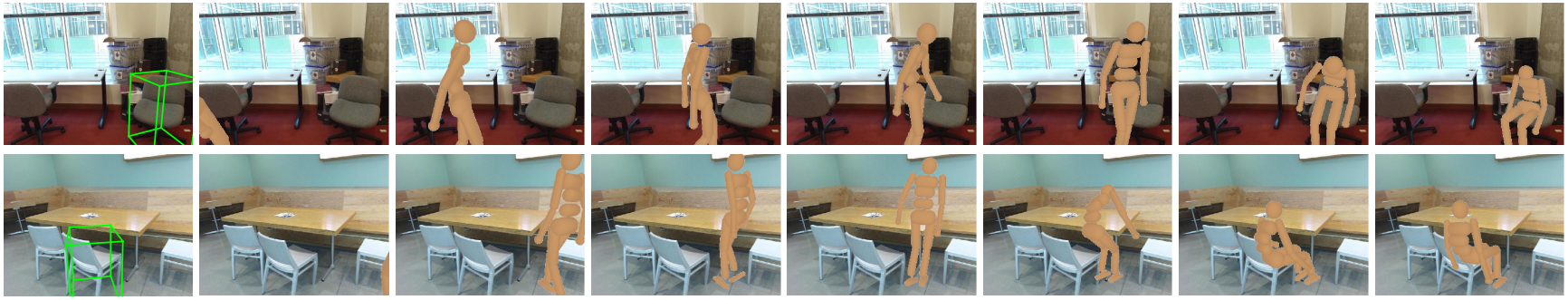


Figure 6.8: Synthesizing sitting motions from a single image. The first column shows the 3D reconstruction output from [84].

6.8 Motion Synthesis in Human Living Space

We demonstrate a vision-based application of our approach by synthesizing sitting motions from a single RGB image that depicts human living space with chairs. First, we reconstruct the 3D configuration of the scene using the method proposed by Huang et al. [84]. We then align the observed scene with the simulated environment using the detected chair and its estimated 3D position and orientation. This enables us to transfer the synthesized sitting motion to the observed scene. Fig. 6.8 shows two image examples rendered with synthesized humanoid motion. Note that the motion looks physically plausible in these examples, while in general this is not always to case, since we do not model the other objects (e.g. tables) in the scene. An interesting future direction is to learn the motion by simulating a more realistic environment with cluttered objects. Another interesting application is to synthesize motions based on observed humans in the image. This is possible given the recent advance on extracting 3D human pose from a single image [152].

6.9 Summary

We study motion synthesis of one particular high-level task—sitting onto a chair. We propose a hierarchical reinforcement learning framework, which relies on a collection of subtask controllers trained to imitate reusable mocap motions, and a meta controller that properly executes the subtasks to complete the main task. We experimentally demonstrate the strength of the hierarchical approach over single level approaches, and also show that our approach can be applied to motion prediction given image input.

CHAPTER 7

Temporal Action Localization [1]

7.1 Introduction

Previous chapters have focused on action understanding from a static image input. However, a single image frame may not always contain enough information for explaining the observed actions. For example, given only an image of a person holding a door knob, it is difficult to tell whether he is "opening" or "closing" the door. In contrast, video observations (i.e. frame sequences) can provide temporal information that is substantial for understanding the action. Action understanding in video is also of practical importance due to the increasing prevalence of video recording devices such as smart phones. In this chapter we specifically tackle the challenges in video-based action understanding.

Action understanding in video is conventionally studied in the setup of action classification [198, 170, 144], where the goal is to perform forced-choice classification of a temporally trimmed video clip into one of several action classes. Despite fruitful progress, this classification setup is unrealistic, because real-world videos are usually untrimmed and the actions of interest are typically embedded in a background of irrelevant activities. Recent research attention has gradually shifted to temporal action localization in untrimmed video [94, 147, 199], where the task is to not only identify the action class, but also detect the start and end time of each action instance. Improvements in temporal action localization can drive progress on a large number of important topics ranging from immediate applications, such as extracting highlights in sports video, to higher-level tasks, such as automatic video captioning.

Temporal action localization, like object detection, falls under the umbrella of visual detection problems. While object detection aims to produce spatial bounding boxes in a 2D image, temporal action localization aims to produce temporal segments in a 1D sequence of frames. As a result, many approaches to action localization have drawn inspiration from advances in object

[1] This chapter is based on a joint work with Sudheendra Vijayanarasimhan, Bryan Seybold, David A. Ross, Jia Deng, and Rahul Sukthankar [22].

detection. A successful example is the use of region-based detectors [66, 65, 158]. These methods first generate a collection of class-agnostic region proposals from the full image, and go through each proposal to classify its object class. To detect actions, one can follow this paradigm by first generating segment proposals from the full video, followed by classifying each proposal.

Among region-based detectors, Faster R-CNN [158] has been widely adopted in object detection due to its competitive detection accuracy on public benchmarks [123, 52]. The core idea is to leverage the immense capacity of deep neural networks (DNNs) to power the two processes of proposal generation and object classification. Given its success in object detection in images, there is considerable interest in employing Faster R-CNN for temporal action localization in video. However, such a domain shift introduces several challenges. We review the issues of Faster R-CNN in the action localization domain, and redesign the architecture to specifically address them. We focus on the following:

1. **How to handle large variations in action durations?** The temporal extent of actions varies dramatically compared to the size of objects in an image—from a fraction of a second to minutes. However, Faster R-CNN evaluates different scales of candidate proposals (i.e., anchors) based on a shared feature representation, which may not capture relevant information due to a misalignment between the temporal scope of the feature (i.e. receptive field) and the span of the anchor. We propose to enforce such alignment using a multi-tower network and dilated temporal convolutions.

2. **How to utilize temporal context?** The moments preceding and following an action instance contain critical information for localization and classification (arguably more so than the spatial context of an object). A naive application of Faster R-CNN would fail to exploit this temporal context. We propose to explicitly encode temporal context by extending the receptive fields in proposal generation and action classification.

3. **How best to fuse multi-stream features?** State-of-the-art action classification results are mostly achieved by fusing RGB and optical flow based features. However, there has been limited work in exploring such feature fusion for Faster R-CNN. We propose a late fusion scheme and empirically demonstrate its edge over the common early fusion scheme.

Our contributions are twofold: (1) we introduce the Temporal Action Localization Network (TAL-Net), which is a new approach for action localization in video based on Faster R-CNN; (2) we achieve state-of-the-art performance on both action proposal and localization on the THUMOS'14 detection benchmark [90], along with competitive performance on the ActivityNet dataset [16].

7.2 Related Work

7.2.1 Action Recognition

Action recognition is conventionally formulated as a classification problem. The input is a video that has been temporally trimmed to contain a specific action of interest, and the goal is to classify the action. Tremendous progress has recently been made due to the introduction of large datasets and the developments on deep neural networks [170, 144, 186, 201, 19, 55]. However, the assumption of trimmed input limits the application of these approaches in real scenarios, where the videos are usually untrimmed and may contain irrelevant backgrounds.

7.2.2 Temporal Action Localization

Temporal action localization assumes the input to be a long, untrimmed video, and aims to identify the start and end times as well as the action label for each action instance in the video. The problem has recently received significant research attention due to its potential application in video data analysis. Below we review the relevant work on this problem.

Early approaches address the task by applying temporal sliding windows followed by SVM classifiers to classify the action within each window [94, 147, 199, 145, 227]. They typically extract improved dense trajectory [198] or pre-trained DNN features, and globally pool these features within each window to obtain the input for the SVM classifiers. Instead of global pooling, Yuan et al. [227] proposed a multi-scale pooling scheme to capture features at multiple resolutions. However, these approaches might be computationally inefficient, because one needs to apply each action classifier exhaustively on windows of different sizes at different temporal locations throughout the entire video.

Another line of work generates frame-wise or snippet-wise action labels, and uses these labels to define the temporal boundaries of actions [130, 171, 38, 116, 228, 81]. One major challenge here is to enable temporal contextual reasoning in predicting the individual labels. Lea et al. [116] proposed novel temporal convolutional architectures to capture long-range temporal dependencies, while others [130, 171, 38] use recurrent neural networks. A few other methods add a separate contextual reasoning stage on top of the frame-wise or snippet-wise prediction scores to explicitly model action durations or temporal transitions [159, 228, 81].

Inspired by the recent success of region-based detectors in object detection [66, 65], many recent approaches adopt a two-stage, proposal-plus-classification framework [17, 168, 51, 13, 15, 167, 235], i.e. first generating a sparse set of class-agnostic segment proposals from the input video, followed by classifying the action categories for each proposal. A large number of these works focus on improving the segment proposals [17, 51, 15, 13], while others focus on building

more accurate action classifiers [167, 235]. However, most of these methods do not afford end-to-end training on either the proposal or classification stage. Besides, the proposals are typically selected from sliding windows of predefined scales [168], where the boundaries are fixed and may result in imprecise localization if the windows are not dense.

As the latest incarnation of the region-based object detectors, the Faster R-CNN architecture [158] is composed of end-to-end trainable proposal and classification networks, and applies region boundary regression in both stages. A few very recent works have started to apply such architecture to temporal action localization [61, 36, 62, 207], and demonstrated competitive detection accuracy. In particular, the R-C3D network [207] is a classic example that closely follows the original Faster R-CNN in many design details. While being a powerful detection paradigm, we argue that naively applying the Faster R-CNN architecture to temporal action localization might suffer from a few issues. We propose to address these issues in this chapter. We will also clarify our contributions over other Faster R-CNN based methods [61, 36, 62, 207] later when we introduce TAL-Net.

In addition to the works reviewed above, there exist other classes of approaches, such as those based on single-shot detectors [12, 122] or reinforcement learning [224]. Others have also studied temporal action localization in a weakly supervised setting [178, 200], where only video-level action labels are available for training. Also note that besides temporal action localization, there also exists a large body of work on spatio-temporal action localization [67, 93, 173], which is beyond the scope of this chapter.

7.3 Faster R-CNN

We briefly review the Faster R-CNN detection framework in this section. Faster R-CNN is first proposed to address object detection [158], where given an input image, the goal is to output a set of detection bounding boxes, each tagged with an object class label. The full pipeline consists of two stages: *proposal generation* and *classification*. First, the input image is processed by a 2D ConvNet to generate a 2D feature map. Another 2D ConvNet (referred to as the Region Proposal Network) is then used to generate a sparse set of class-agnostic region proposals, by classifying a group of scale varying anchor boxes centered at each pixel location of the feature map. The boundaries of the proposals are also adjusted with respect to the anchor boxes through regression. Second, for each region proposal, features within the region are first pooled into a fixed size feature map (i.e. RoI pooling [65]). Using the pooled feature, a DNN classifier then computes the object class probabilities and simultaneously regresses the detection boundaries for each object class. Fig. 7.1 (left) illustrates the full pipeline. The framework is conventionally trained by alternating between the training of the first and second stage [158].

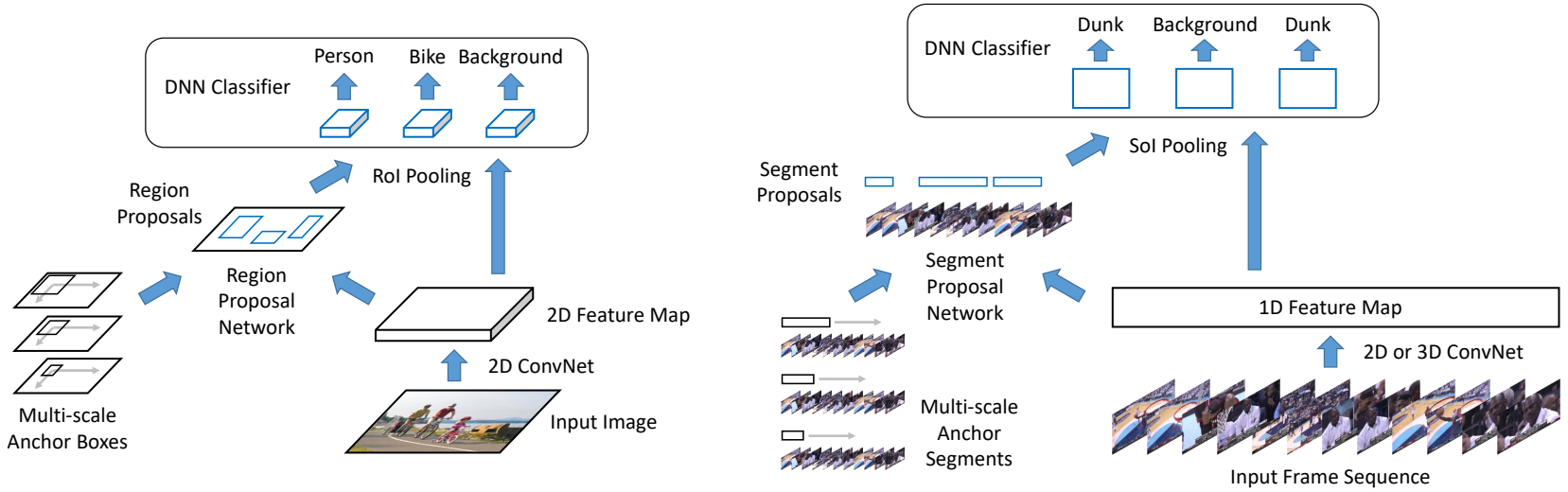


Figure 7.1: Contrasting the Faster R-CNN architecture for object detection in images [158] (left) and temporal action localization in video [61, 36, 62, 207] (right). Temporal action localization can be viewed as the 1D counterpart of the object detection problem.

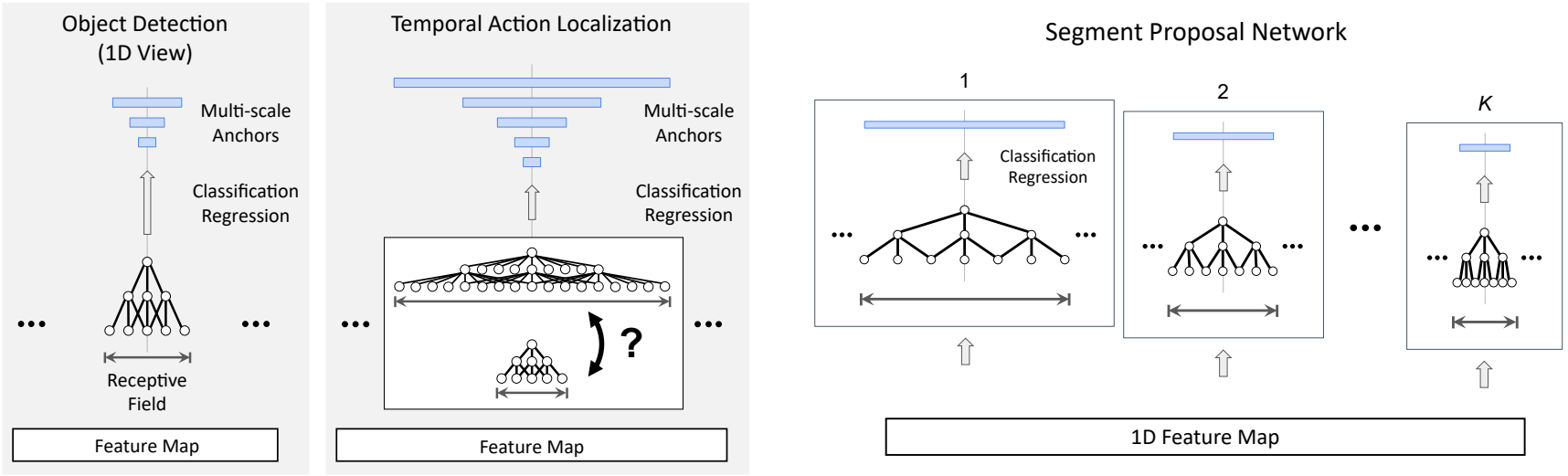


Figure 7.2: Left: The limitation of sharing the receptive field across different anchor scales in temporal action localization. Right: The multi-tower architecture of our Segment Proposal Network. Each anchor scale has an associated network with aligned receptive field.

Faster R-CNN naturally extends to temporal action localization [61, 36, 207]. Recall that object detection aims to detect 2D spatial regions, whereas in temporal action localization, the goal is to detect 1D temporal *segments*, each represented by a *start* and an *end* time. Temporal action localization can thus be viewed as the 1D counterpart of object detection. A typical Faster R-CNN pipeline for temporal action localization is illustrated in Fig. 7.1 (right). Similar to object detection, it consists of two stages. First, given a sequence of frames, we extract a 1D feature map, typically via a 2D or 3D ConvNet. The feature map is then passed to a 1D ConvNet [2] (referred to as the Segment Proposal Network) to classify a group of scale varying anchor segments at each temporal location, and also regress their boundaries. This returns a sparse set of class-agnostic segment proposals. Second, for each segment proposal, we compute the action class probabilities and further regress the segment boundaries, by first applying a 1D RoI pooling (termed "SoI pooling") layer followed by a DNN classifier.

7.4 TAL-Net

TAL-Net follows the Faster R-CNN detection paradigm for temporal action localization (Fig. 7.1 right) but features three novel architectural changes (Sec. 7.4.1 to 7.4.3).

7.4.1 Receptive Field Alignment

Recall that in proposal generation, we generate a sparse set of class-agnostic proposals by classifying a group of scale varying anchors at each location in the feature map. In object detection [158], this is achieved by applying a small ConvNet on top of the feature map, followed by a 1×1 convolutional layers with K filters, where K is the number of scales. Each filter will classify the anchor of a particular scale. This reveals an important *limitation*: the anchor classifiers at each location share the same receptive field. Such design may be reasonable for object detection, but may not generalize well to temporal action localization, because the temporal length of actions can vary more drastically compared to the spatial size of objects, e.g. in THUMOS'14 [90], the action lengths range from less than a second to more than a minute. To ensure a high recall, the applied anchor segments thus need to have a wide range of scales (Fig. 7.2 left). However, if the receptive field is set too small (i.e. temporally short), the extracted feature may not contain sufficient information when classifying large (i.e. temporally long) anchors, while if it is set too large, the extracted feature may be dominated by irrelevant information when classifying small anchors.

To address this issue, we propose to align each anchor's receptive field with its temporal span. This is achieved by two key enablers: a *multi-tower* network and *dilated temporal convolutions*.

[2] "1D convolution" & "temporal convolution" are used interchangeably.

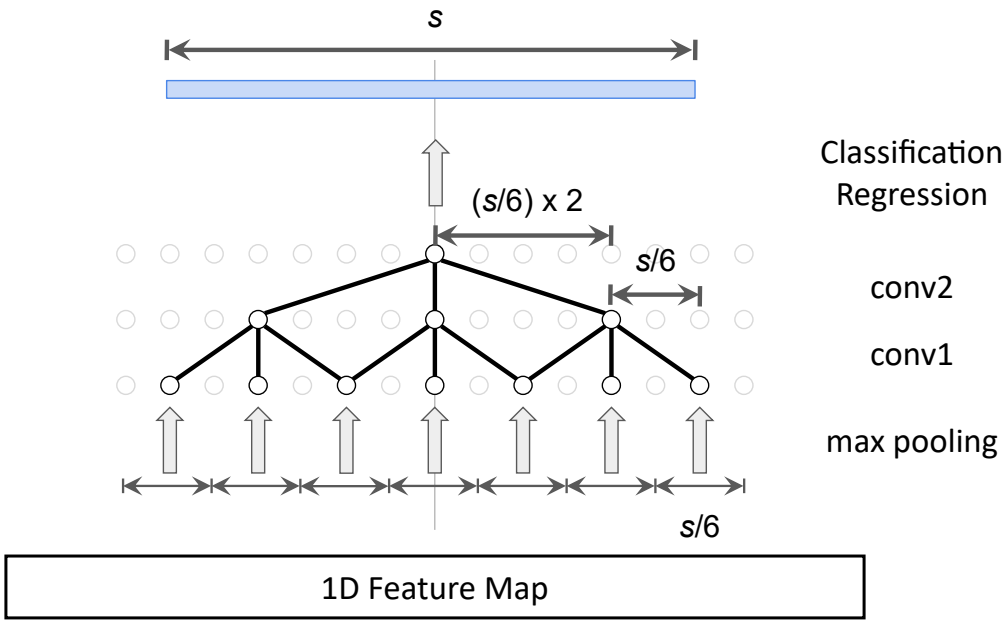


Figure 7.3: Controlling the receptive field size s with dilated temporal convolutions.

Given a 1D feature map, our Segment Proposal Network is composed of a collection of K temporal ConvNets, each responsible for classifying the anchor segments of a particular scale (Fig. 7.2 right). Most importantly, each temporal ConvNet is carefully designed such that its receptive field size coincides with the associated anchor scale. At the end of each ConvNet, we apply two parallel convolutional layers with kernel size 1 for anchor classification and boundary regression, respectively.

The next question is: how do we design temporal ConvNets with a controllable receptive field size s? Suppose we use temporal convolutional filters with kernel size 3 as a building block. One way to increase s is simply stacking the convolutional layers: $s = 2L + 1$ if we stack L layers. However, given a target receptive field size s, the required number of layers L will then grow linearly with s, which can easily increase the number of parameters and make the network prone to overfitting. One solution is to apply pooling layers: if we add a pooling layer with kernel size 2 after each convolutional layer, the receptive field size is then given by $s = 2^{(L+1)} - 1$. While now L grows logarithmically with s, the added pooling layers will exponentially reduce the resolution of the output feature map, which may sacrifice localization precision in detection tasks.

To avoid overgrowing the model while maintaining the resolution, we propose to use dilated temporal convolutions. Dilated convolutions [27, 225] act like regular convolutions, except that one subsamples pixels in the input feature map instead of taking adjacent ones when multiplied with a convolution kernel. This technique has been successfully applied to 2D ConvNets [27, 225] and 1D ConvNets [116] to expand the receptive field without loss of resolution. In our Segment Proposal Network, each temporal ConvNet consists of only two dilated convolutional layers (Fig. 7.3).

To attain a target receptive field size s, we can explicitly compute the required dilation rate (i.e. subsampling rate) r_l for layer l by $r_1 = s/6$ and $r_2 = (s/6) \times 2$. We also smooth the input before subsampling by adding a max pooling layer with kernel size $s/6$ before the first convolutional layer.

7.4.1.1 Contributions beyond [36, 61, 62, 207]

Xu et al. [207] followed the original Faster R-CNN and thus their anchors at each pixel location still shared the receptive field. Both Gao et al. [61, 62] and Dai et al. [36] aligned each anchor's receptive field with its span. However, Gao et al. [61, 62] average pooled the features within the span of each anchor, whereas we use temporal convolutions to extract structure-sensitive features. Our approach is similar in spirit to Dai et al. [36], which sampled a fixed number of features within the span of each anchor; we approach this using dilated convolutions.

7.4.2 Context Feature Extraction

Temporal context information (i.e. what happens immediately before and after an action instance) is a critical signal for temporal action localization for two reasons. First, it enables more accurate localization of the action boundaries. For example, seeing a person standing still on the far end of a diving board is a strong signal that he will soon start a "diving" action. Second, it provides strong semantic cues for identifying the action class within the boundaries. For example, seeing a javelin flying in the air indicates that a person just finished a "javelin throw", not "pole vault". As a result, it is critical to encode the temporal context features in the action localization pipeline. Below we detail our approach to explicitly exploit context features in both the proposal generation and action classification stage.

In proposal generation, we showed the receptive field for classifying an anchor can be matched with the anchor's span (Sec. 7.4.1). However, this only extracts the features within the anchor, and overlooks the contexts before and after it. To ensure the context features are used for anchor classification and boundary regression, the receptive field must cover the context regions. Suppose the anchor is of scale s, we enforce the receptive field to also cover the two segments of length $s/2$ immediately before and after the anchor. This can be achieved by doubling the dilation rate of the convolutional layers, i.e. $r_1 = (s/6) \times 2$ and $r_2 = (s/6) \times 2 \times 2$, as illustrated in Fig. 7.4. Consequently, we also double the kernel size of the initial max pooling layer to $(s/6) \times 2$.

In action classification, we perform SoI pooling (i.e. 1D RoI pooling) to extract a fixed size feature map for each obtained proposal. We illustrate the mechanism of SoI pooling with output size 7 in Fig. 7.5 (top). Note that as in the original design of RoI pooling [65, 158], pooling is applied to the region strictly within the proposal, which includes no temporal contexts. We propose

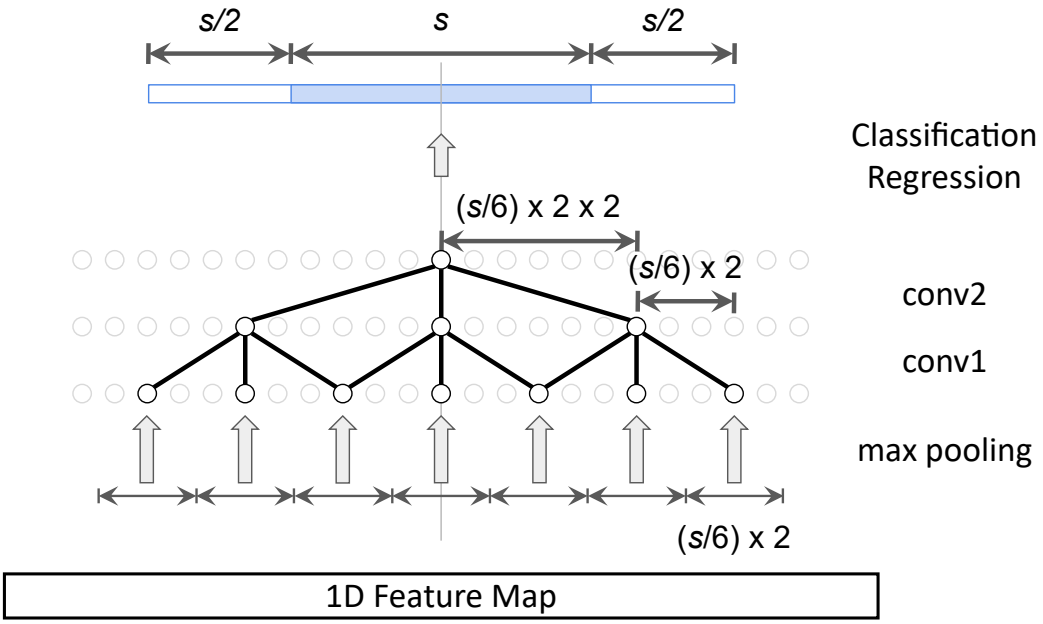


Figure 7.4: Incorporating context features in proposal generation.

to extend the input extent of SoI pooling. As shown in Fig. 7.5 (bottom), for a proposal of size s, the extent of our SoI pooling covers not only the proposal segment, but also the two segments of size $s/2$ immediately before and after the proposal, similar to the classification of anchors. After SoI pooling, we add one fully-connected layer, followed by a final fully-connected layer, which classifies the action and regresses the boundaries.

7.4.2.1 Contributions beyond [36, 61, 62, 207]

Xu et al. [207] did not exploit any context features in either proposal generation or action classification. Dai et al. [36] included context features when generating proposals, but used only the features within the proposal in action classification. Gao et al. exploited context features in either proposal generation only [62] or both stages [61]. However, they average-pooled the features within the context regions, while we use temporal convolutions and SoI pooling to encode the temporal structure of the features.

7.4.3 Late Feature Fusion

In action classification, most of the state-of-the-art methods [170, 144, 201, 19, 55] rely on a two-stream architecture, which parallelly processes two types of input—RGB frames and pre-computed optical flow—and later fuses their features to generate the final classification scores. We hypothesize such two-stream input and feature fusion may also play an important role in temporal action localization. Therefore we propose a *late fusion* scheme for the two-stream Faster R-CNN

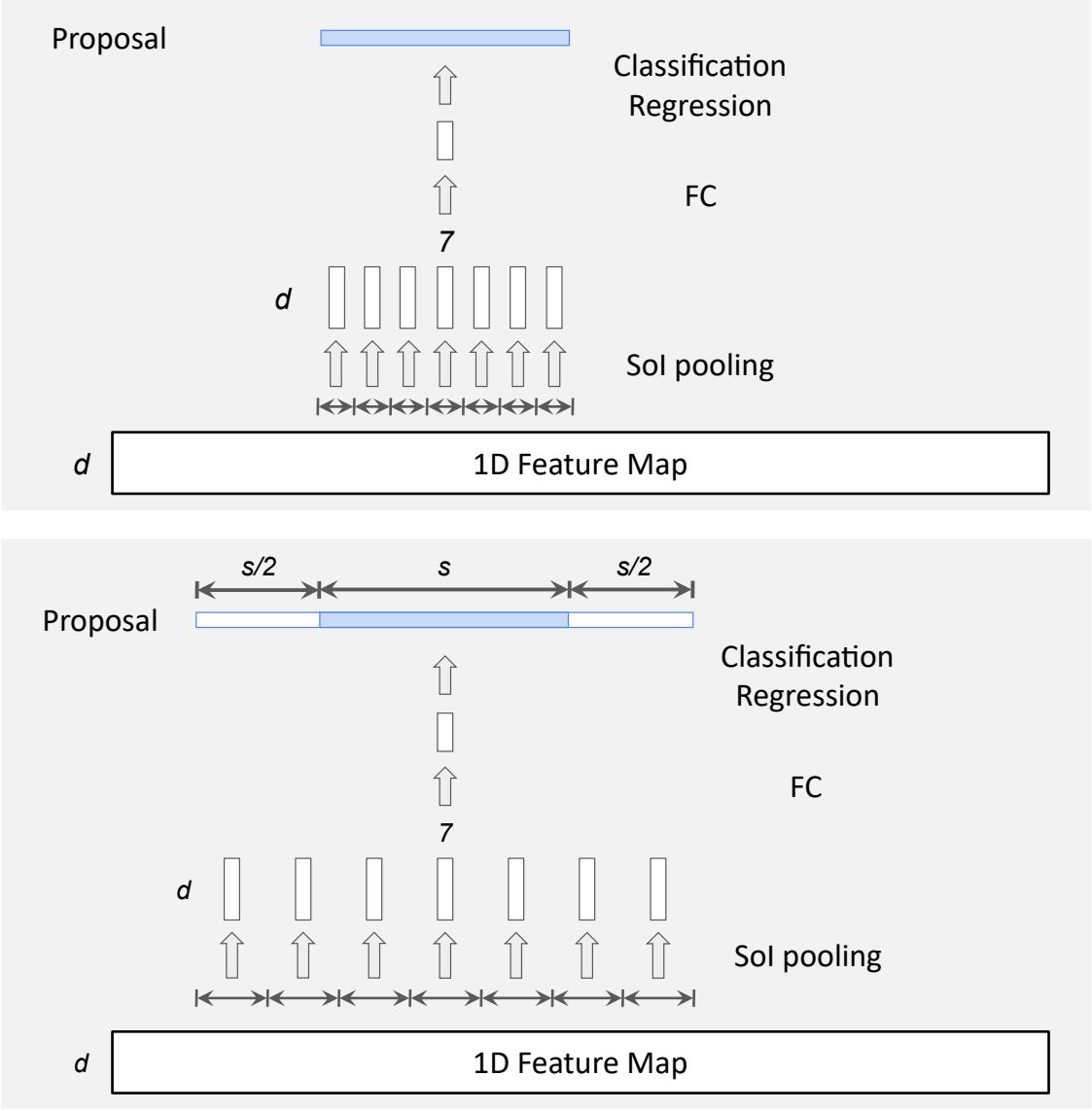


Figure 7.5: Classifying a proposal without (top) [65, 158] and with (bottom) incorporating context features

framework. Conceptually, this is equivalent to performing the conventional late fusion in both the proposal generation and action classification stage (Fig. 7.6). We first extract two 1D feature maps from RGB frames and stacked optical flow, respectively, using two different networks. We process each feature map by a distinct Segment Proposal Network, which parallelly generates the logits for anchor classification and boundary regression. We use the element-wise average of the logits from the two networks as the final logits to generate proposals. For each proposal, we perform SoI pooling parallelly on both feature maps, and apply a distinct DNN classifier on each output. Finally, the logits for action classification and boundary regression from both DNN classifiers are element-wisely averaged to generate the final detection output.

Note that a more straightforward way to fuse two features is through an *early fusion* scheme:

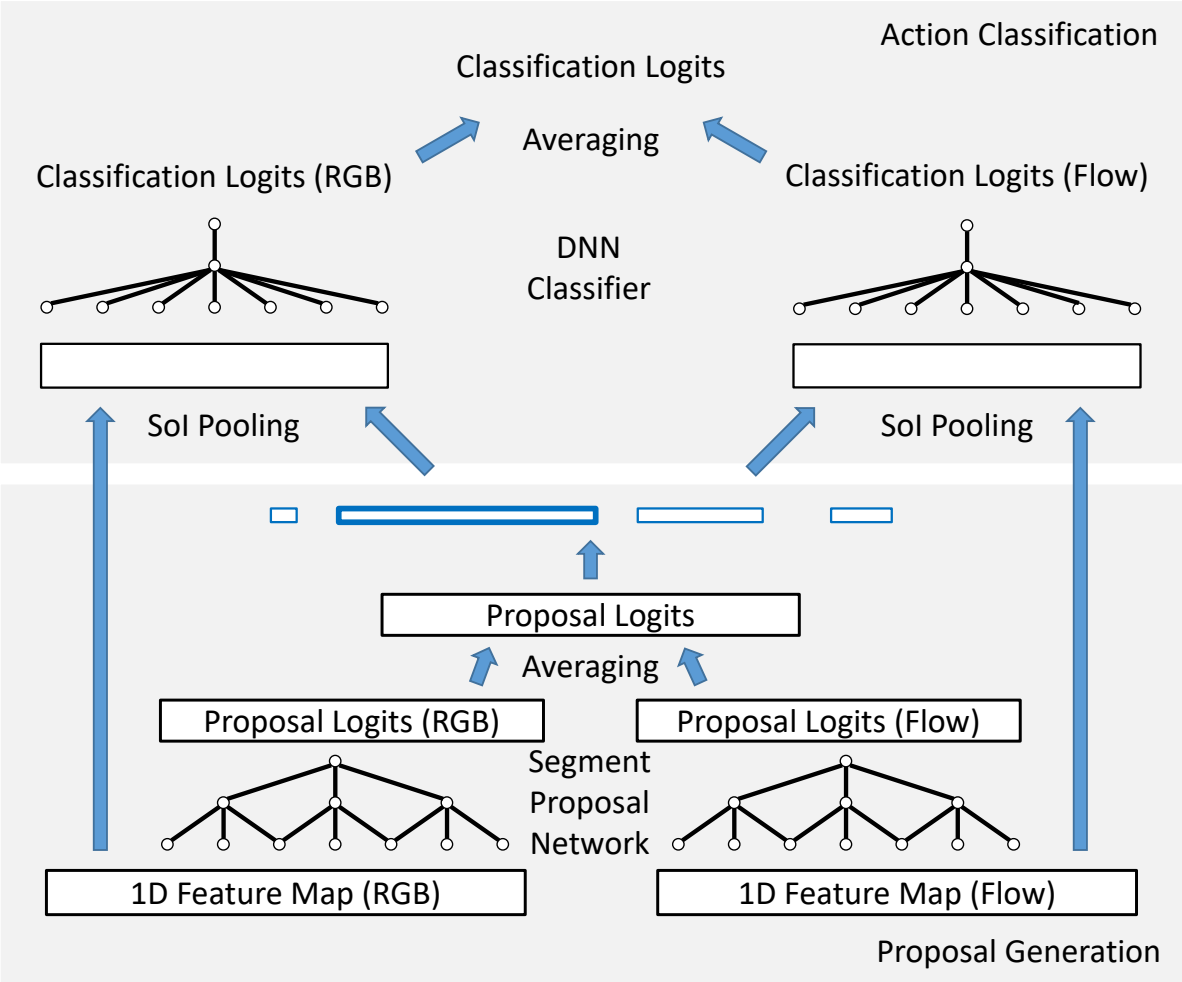


Figure 7.6: The late fusion scheme for the two-stream Faster R-CNN framework.

we concatenate the two 1D feature maps in the feature dimension, and apply the same pipeline as before (Sec. 7.4.1 and 7.4.2). We show by experiments that the aforementioned late fusion scheme outperforms the early fusion scheme.

7.4.3.1 Contributions beyond [36, 61, 62, 207]

Xu et al. [207] only used a single-stream feature (C3D). Both Dai et al. and Gao et al. used two-stream features, but either did not perform fusion [62] or only tried the early fusion scheme [36, 61].

7.5 Experiments

7.5.1 Dataset

We perform ablation studies and state-of-the-art comparisons on the temporal action detection benchmark of THUMOS'14 [90]. The dataset contains videos from 20 sports action classes. Since the training set contains only trimmed videos with no temporal annotations, we use the 200 untrimmed videos (3,007 action instances) in the validation set to train our model. The test set consists of 213 videos (3,358 action instances). Each video is on average more than 3 minutes long, and contains on average more than 15 action instances, making the task particularly challenging. Besides THUMOS'14, we separately report our results on ActivityNet v1.3 [16] at the end of the section.

7.5.2 Evaluation Metrics

We consider two tasks: *action proposal* and *action localization*. For action proposal, we calculate Average Recall (AR) at different Average Number of Proposals per Video (AN) using the public code provided by [51]. AR is defined by the average of all recall values using tIoU thresholds from 0.5 to 1 with a step size of 0.05. For action localization, we report mean Average Precision (mAP) using different tIoU thresholds.

7.5.3 Features

To extract the feature maps, we first train a two-stream "Inflated 3D ConvNet" (I3D) model [19] on the Kinetics action classification dataset [97]. The I3D model builds upon state-of-the-art image classification architectures (i.e. Inception-v1 [179]), but inflates their filters and pooling kernels into 3D, leading to very deep, naturally spatiotemporal classifiers. The model takes as input a stack of 64 RGB/optical flow frames, performs spatio-temporal convolutions, and extracts a 1024-dimensional feature as the output of an average pooling layer. We extract both RGB and optical flow frames at 10 frames per second (fps) as input to the I3D model. To compute optical flow, we use a FlowNet [49] model trained on artificially generated data followed by fine-tuning on the Kinetics dataset using an unsupervised loss [188]. After training on Kinetics we fix the model and extract the 1024-dimensional output of the average pooling layer by stacking every 16 RGB/optical flow frames in the frame sequence. The input to our action localization model is thus two 1024-dimensional feature maps—for RGB and optical flow—sampled at 0.625 fps from the input videos.

7.5.4 Implementation Details

Our implementation is based on the TensorFlow Object Detection API [83]. In proposal generation, we apply anchors of the following scales: $\{1, 2, 3, 4, 5, 6, 8, 11, 16\}$, i.e. $K = 9$. We set the number of filters to 256 for all convolutional and fully-connected layers in the Segment Proposal Network and the DNN classifier. We add a convolutional layer with kernel size 1 to reduce the feature dimension to 256 before the Segment Proposal Network and after the SoI pooling layer. We apply Non-Maximum Suppression (NMS) with tIoU threshold 0.7 on the proposal output and keep the top 300 proposals for action classification. The same NMS is applied to the final detection output for each action class separately.

7.5.5 Training Strategy

The training of TAL-Net largely follows the Faster R-CNN implementation in [83]. Both proposal generation and action classification share a same form of multi-task loss, targeting both classification and regression:

$$\mathcal{L} = \sum_i \mathcal{L}_{cls}(p_i, p_i^*) + \lambda \sum_i [p_i^* \geq 1] \mathcal{L}_{reg}(t_i, t_i^*). \tag{7.1}$$

i is the index of an anchor or proposal in a mini-batch. For classification, p is the predicted probability of the proposal or actions, p^* is the ground-truth label, and $\mathcal{L}_{cls}$ is the cross-entropy loss. Note that $p^* \in \{0, 1\}$ for proposal generation, and $p^* \in \{0, \ldots, C\}$ for action classification, where C is the number of action classes of interest and 0 accounts for the background action class. For regression, t is the predicted offset relative to an anchor or proposal, t^* is the ground-truth offset, and $\mathcal{L}_{reg}$ is the smooth L1 loss defined in [65]. We parameterize the offsets $t = (t_c, t_l)$ and $t^* = (t_c^*, t_l^*)$ by:

$$\begin{aligned} t_c &= 10 \cdot (c - c_a)/l_a, \quad & t_l &= 5 \cdot \log(l/l_a), \\ t_c^* &= 10 \cdot (c^* - c_a)/l_a, \quad & t_l^* &= 5 \cdot \log(l^*/l_a), \end{aligned} \tag{7.2}$$

where c and l denote the segment's center coordinate and its length. c and c^* account for the predicted and ground-truth segments, while c_a accounts for the anchor and proposal segments, for proposal generation and action classification, respectively (similarly for l). The indicator function $[\cdot]$ is used to exclude the background anchors and proposals when the regression loss is computed. In all experiments, we set $\lambda = 1$ for both proposal generation and action classification, and jointly train both stages by weighing both losses equally.

For proposal generation, an anchor is assigned a positive label if it overlaps with a ground-truth segment with temporal Intersection-over-Union (tIoU) higher than 0.7. A negative label is assigned if the tIoU overlap is lower than 0.3 with all ground-truth segments. We also force each ground-

AN	10	20	50	100	200
Single	9.4	15.3	25.3	33.9	41.3
Single + TConv	12.9	20.0	30.3	37.6	44.0
Multi + TConv	13.4	20.6	31.1	38.1	43.7
Multi + Dilated	**14.0**	**21.7**	**31.9**	**38.8**	**44.7**
Single	11.0	18.0	28.9	36.8	43.6
Single + TConv	15.1	23.2	33.7	40.0	44.7
Multi + TConv	15.7	24.0	35.0	41.1	46.2
Multi + Dilated	**16.3**	**25.4**	**35.8**	**42.3**	**47.5**

Table 7.1: Results for receptive field alignment on proposal generation in AR (%). Top: RGB stream. Bottom: Flow stream.

truth segment to have at least one matched positive anchor. For action classification, a proposal is assigned the action label of its most overlapped ground-truth segment, if the ground-truth segment has tIoU overlap over 0.5. Otherwise a background label (i.e. 0) is assigned.

Each mini-batch contains examples sampled from a single video. For proposal generation, we set the mini-batch size to 256 and the fraction of positives to 0.5. For action classification, we set the mini-batch size to 64 and the fraction of foreground actions to 0.25. We use the Adam optimizer with a learning rate of 0.0001.

7.5.6 Receptive Field Alignment

We validate the design for receptive field alignment by comparing four baselines: (1) a single-tower network with no temporal convolutions (Single), where each anchor is classified solely based on the feature at its center location; (2) a single-tower network with non-dilated temporal convolutions (Single+TConv), which represents the default Faster R-CNN architecture; (3) a multi-tower network with non-dilated temporal convolutions (Multi+TConv); (4) a multi-tower network with dilated temporal convolutions (Multi+Dilated, the proposed architecture). All temporal ConvNets have two layers, both with kernel size 3. Here we consider only a single-steam feature (i.e. RGB or flow) and evaluate the generated proposal with AR-AN. The results are reported in Tab. 7.1 (top for RGB and bottom for flow). The trend is consistent on both features: Single performs the worst, since it relies only on the context at the center location; Single+TConv and Multi+TConv both perform better than Single, but still, suffer from irrelevant context due to misaligned receptive fields; Multi-Dilated outperforms the others, as the receptive fields are properly aligned with the span of anchors.

AN	10	20	50	100	200
Multi + Dilated	14.0	21.7	31.9	38.8	44.7
Multi + Dilated + Context	**15.1**	**22.2**	**32.3**	**39.9**	**46.8**
Multi + Dilated	16.3	25.4	35.8	42.3	47.5
Multi + Dilated + Context	**17.4**	**26.5**	**36.5**	**43.3**	**48.6**

Table 7.2: Results for incorporating context features in proposal generation in AR (%). Top: RGB stream. Bottom: Flow stream.

tIoU	0.1	0.3	0.5	0.7	0.9
SoI Pooling	44.9	38.4	28.5	13.0	**0.6**
SoI Pooling + Context	**49.3**	**42.6**	**31.9**	**14.2**	**0.6**
SoI Pooling	49.8	45.7	37.4	**18.8**	0.7
SoI Pooling + Context	**54.3**	**48.8**	**38.2**	18.6	**0.9**

Table 7.3: Results for incorporating context features in action classification in mAP (%). Top: RGB stream. Bottom: Flow stream.

tIoU	0.1	0.3	0.5	0.7	0.9
RGB	49.3	42.6	31.9	14.2	0.6
Flow	54.3	48.8	38.2	18.6	**0.9**
Early Fusion	**60.5**	52.8	40.8	19.3	0.8
Late Fusion	59.8	**53.2**	**42.8**	**20.8**	**0.9**

Table 7.4: Results for late feature fusion in mAP (%).

7.5.7 Context Feature Extraction

We first validate our design for context feature extraction in proposal generation. Tab. 7.2 compares the generated proposals before and after incorporating context features (top for RGB and bottom for flow). We achieve higher AR on both streams after the context features are included. Next, given better proposals, we evaluate context feature extraction in action classification. Tab. 7.3 compares the action localization results before and after incorporating context features (top for RGB and bottom for flow). Similarly, we achieve higher mAP nearly at all AN values on both streams after including the context features.

7.5.8 Late Feature Fusion

Tab. 7.4 reports the action localization results of the two single-stream networks and the early and late fusion schemes. First, the flow based feature outperforms the RGB based feature, which coheres with the common observations in action classification [170, 201, 19, 55]. Second, the fused features outperform the two single-stream features, suggesting the RGB and flow features

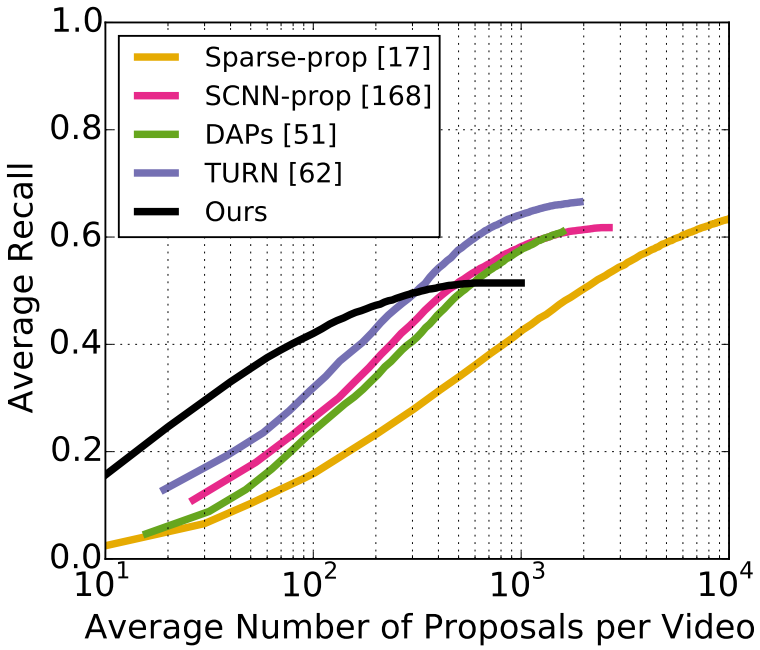


Figure 7.7: Our action proposal result in AR-AN (%) on THUMOS'14 comparing with other state-of-the-art methods.

complement each other. Finally, the late fusion scheme outperforms the early fusion scheme except at tIoU threshold 0.1, validating our proposed design.

7.5.9 State-of-the-Art Comparisons

We compare TAL-Net with state-of-the-art methods on both action proposal and localization. Fig. 7.7 shows the AR-AN curves for action proposal. TAL-Net outperforms all other methods in the low AN region, suggesting our top proposals have higher quality. Although our AR saturates earlier as AN increases, this is because we extract features at a much lower frequency (i.e. 0.625 fps) due to the high computational demand of the I3D models. This reduces the density of anchors and lowers the upper bound of the recall. Tab. 7.5 compares the mAP for action localization. TAL-Net achieves the highest mAP when the tIoU threshold is greater than 0.2, suggesting it can localize the boundaries more accurately. We particularly highlight our result at tIoU threshold 0.5, where TAL-Net outperforms the state-of-the-art by 11.8% mAP (42.8% versus 31.0% from Gao et al. [61]).

7.5.10 Qualitative Results

Fig. 7.8 shows qualitative examples of the top localized actions on THUMOS'14. Each consists of a sequence of frames sampled from a full test video, the ground-truth (blue) and predicted (green) action segments and class labels, and a temporal axis showing the time in seconds. In the

tIoU	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Karaman et al. [94]	4.6	3.4	2.4	1.4	0.9	–	–
Oneata et al. [147]	36.6	33.6	27.0	20.8	14.4	–	–
Wang et al. [199]	18.2	17.0	14.0	11.7	8.3	–	–
Caba Heilbron et al. [17]	–	–	–	–	13.5	–	–
Richard and Gall [159]	39.7	35.7	30.0	23.2	15.2	–	–
Shou et al. [168]	47.7	43.5	36.3	28.7	19.0	10.3	5.3
Yeung et al. [224]	48.9	44.0	36.0	26.4	17.1	–	–
Yuan et al. [227]	51.4	42.6	33.6	26.1	18.8	–	–
Escorcia et al. [51]	–	–	–	–	13.9	–	–
Buch et al. [13]	–	–	37.8	–	23.0	–	–
Shou et al. [167]	–	–	40.1	29.4	23.3	13.1	7.9
Yuan et al. [228]	51.0	45.2	36.5	27.8	17.8	–	–
Buch et al. [12]	–	–	45.7	–	29.2	–	9.6
Gao et al. [61]	60.1	56.7	50.1	41.3	31.0	19.1	9.9
Hou et al. [81]	51.3	–	43.7	–	22.0	–	–
Dai et al. [36]	–	–	–	33.3	25.6	15.9	9.0
Gao et al. [62]	54.0	50.9	44.1	34.9	25.6	–	–
Xu et al. [207]	54.5	51.5	44.8	35.6	28.9	–	–
Zhao et al. [235]	**66.0**	**59.4**	51.9	41.0	29.8	–	–
Ours	59.8	57.1	**53.2**	**48.5**	**42.8**	**33.8**	**20.8**

Table 7.5: Action localization mAP (%) on THUMOS'14.

top example, our method accurately localizes both instances in the video. In the middle example, the action classes are correctly classified, but the start of the leftmost prediction is inaccurate, due to subtle differences between preparation and the start of the action. In the bottom, "ThrowDiscus" is misclassified due to similar context.

Besides Fig. 7.8, we show additional qualitative examples on THUMOS'14 in Fig. 7.9, 7.10, and 7.11. Our approach successfully localizes the actions in most cases. The failure cases include: (1) inaccurate boundaries, e.g. Fig. 7.8 (middle), (2) misclassified actions, e.g. Fig. 7.8 (bottom) and Fig. 7.9 (a), (3) false positives due to indistinguishable body motions, e.g. Fig. 7.9 (b) and Fig. 7.11 (b), and (4) false negatives due to small objects and occlusion, e.g. Fig. 7.10 (a).

7.5.11 Benchmarks using InceptionV3

Besides I3D features, we also evaluate our method with features extracted from an InceptionV3 model pre-trained on ImageNet. This provides an apples-to-apples comparison with the result of Zhao et al. [235] reported in [1]. Tab. 7.6 shows the action localization mAP on THUMOS'14. Our approach outperforms Zhao et al. [235] by 7.7% in mAP, validating the effectiveness of our

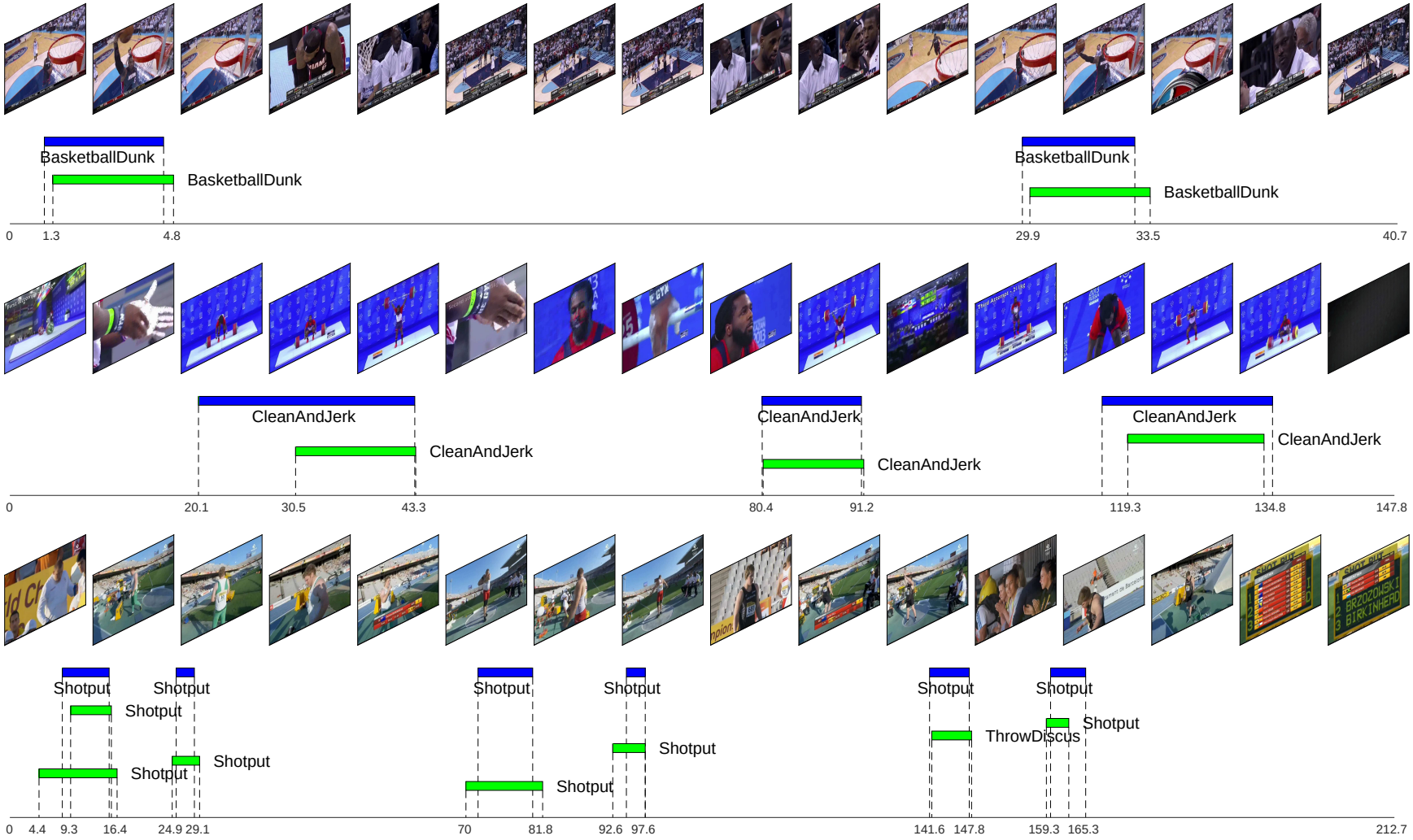


Figure 7.8: Qualitative examples of the top localized actions on THUMOS'14. Each consists of a sequence of frames sampled from a full test video, the ground-truth (blue) and predicted (green) action segments and class labels, and a temporal axis showing the time in seconds.

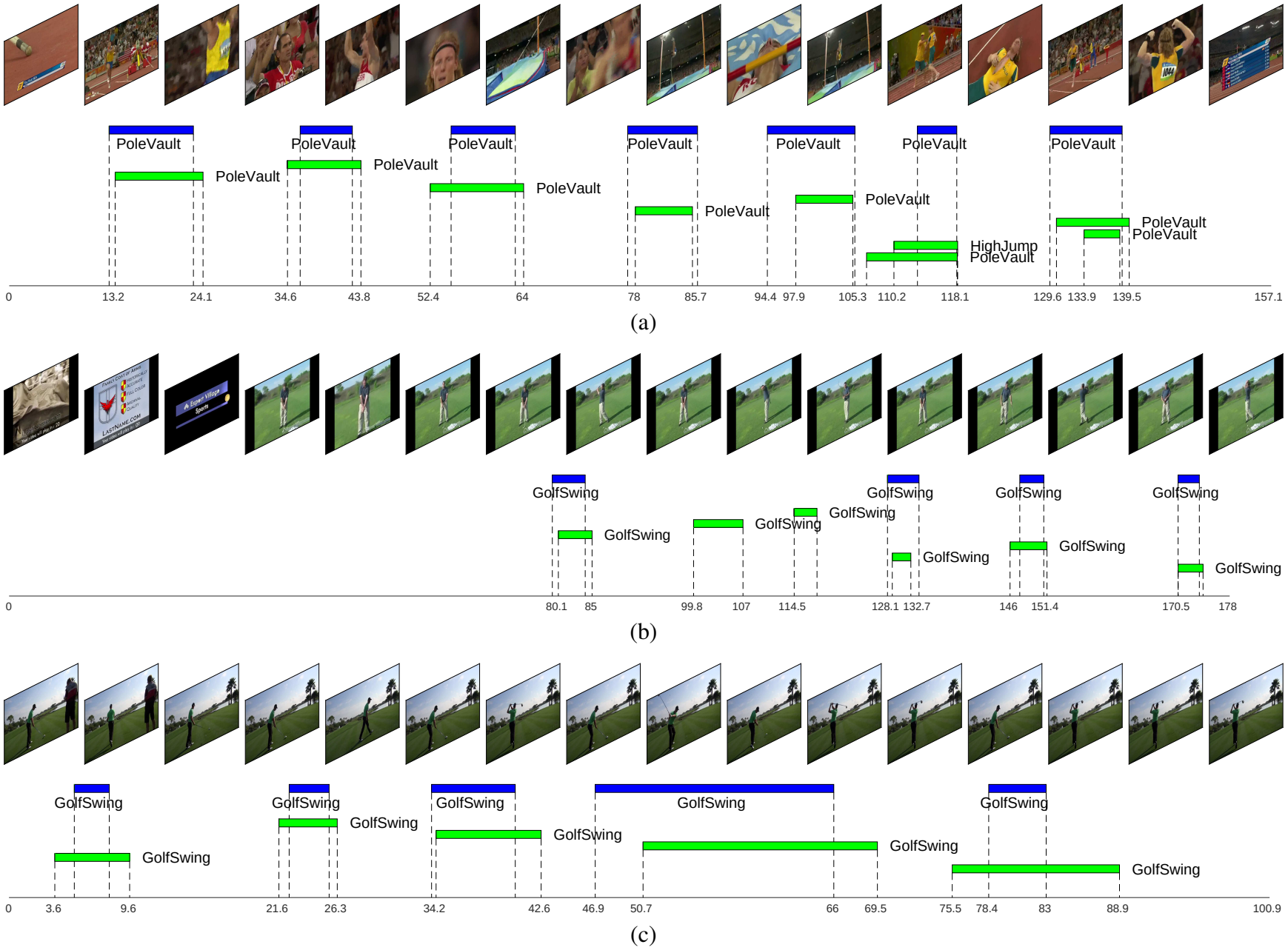
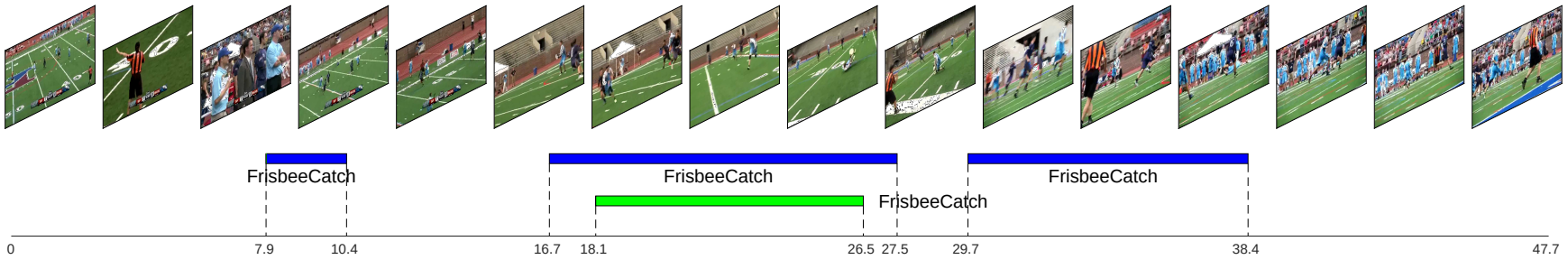
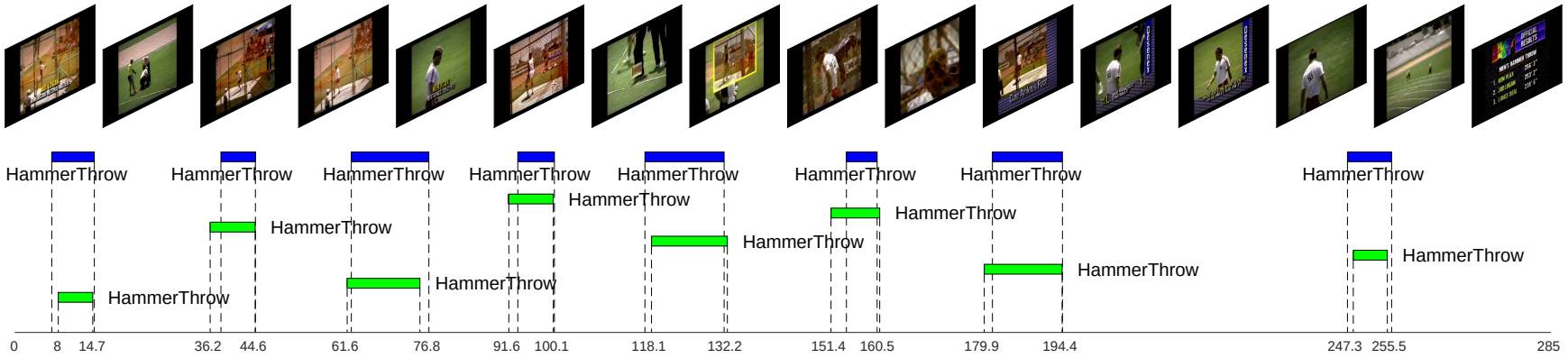


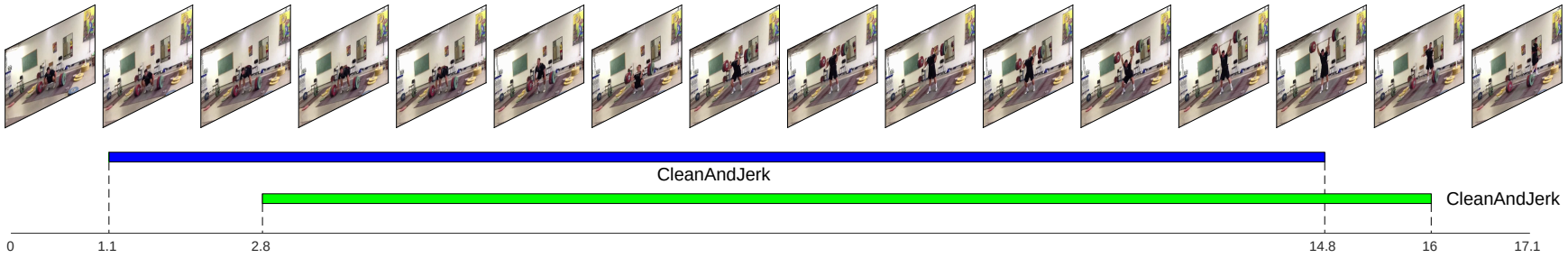
Figure 7.9: Additional qualitative examples of the top localized actions on THUMOS'14. Each consists of a sequence of frames sampled from a full test video, the ground-truth (blue) and predicted (green) action segments and class labels, and a temporal axis showing the time in seconds.



(a)

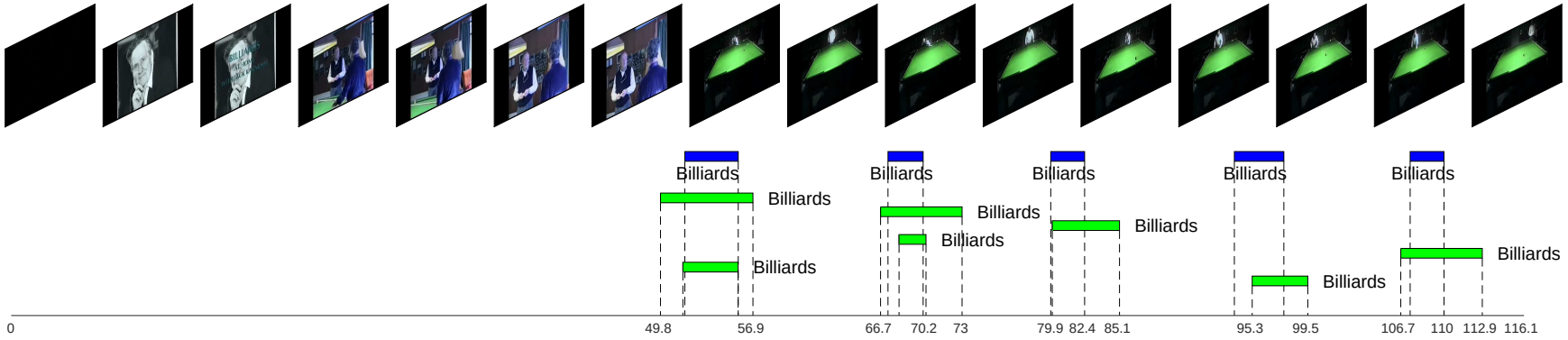


(b)

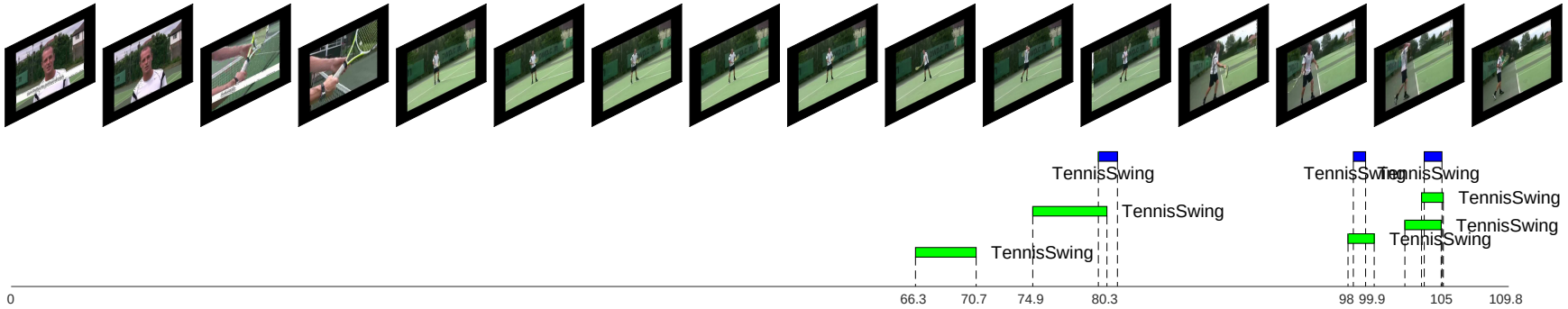


(c)

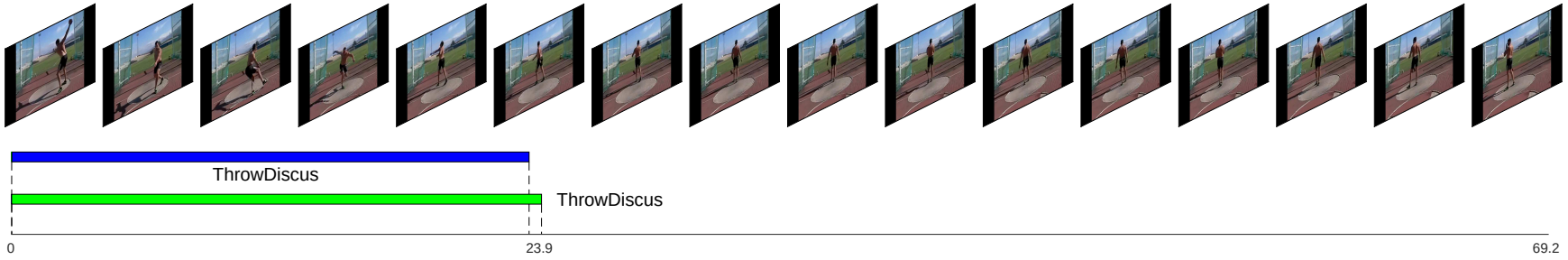
Figure 7.10: Additional qualitative examples of the top localized actions on THUMOS'14. Each consists of a sequence of frames sampled from a full test video, the ground-truth (blue) and predicted (green) action segments and class labels, and a temporal axis showing the time in seconds.



(a)



(b)



(c)

Figure 7.11: Additional qualitative examples of the top localized actions on THUMOS'14. Each consists of a sequence of frames sampled from a full test video, the ground-truth (blue) and predicted (green) action segments and class labels, and a temporal axis showing the time in seconds.

	InceptionV3 RGB (ImageNet pre-trained)
Zhao et al. [235]	18.3
Ours	**26.0**

Table 7.6: Action localization mAP (%) on THUMOS'14 using InceptionV3. The result of [235] is copied from [1].

Step	Running Time (ms)	
Optical Flow	239	per frame
I3D Features	825	per 16 frame input
Proposal + Classification	9	per 3000 frames

Table 7.7: Running time (ms) of each step during test time.

tIoU	0.5	0.75	0.95	Average
Singh and Cuzzolin [172]	34.47	–	–	–
Wang and Tao [202]	43.65	–	–	–
Shou et al. [167]	**45.30**	**26.00**	0.20	**23.80**
Dai et al. [36]	36.44	21.15	**3.90**	–
Xu et al. [207]	26.80	–	–	12.70
Ours	38.23	18.30	1.30	20.22

Table 7.8: Action localization mAP (%) on ActivityNet v1.3 (val).

proposed architecture.

7.5.12 Computational Cost

Tab. 7.7 shows a running time breakdown during the test time for the following three steps: (1) optical flow extraction, (2) I3D feature extraction, and (3) proposal and classification. All these running time experiments are performed on CPUs, so further speedup is possible with GPU devices. The computational bottleneck is on the optical flow extraction (i.e. 239 ms per frame).

7.5.13 Results on ActivityNet

Tab. 7.8 shows our action localization results on the ActivityNet v1.3 validation set along with other recent published results. TAL-Net outperforms other Faster R-CNN based methods at tIoU threshold 0.5 (38.23% vs. 36.44% from Dai et al. [36] and 26.80% from Xu et al. [207]). Note that THUMOS'14 is a better dataset for evaluating action localization than ActivityNet, as the former has more action instances per video and each video contains a larger portion of background activity: on average, the THUMOS'14 training set has 15 instances per video and each video has

71% background, while the ActivityNet training set has only 1.5 instances per video and each video has only 36% background.

7.6 Summary

We introduce TAL-Net, an improved approach to temporal action localization in video that is inspired by the Faster RCNN object detection framework. TAL-Net features three novel architectural changes that address three key shortcomings of existing approaches: (1) receptive field alignment; (2) context feature extraction; and (3) late feature fusion. We achieve state-ofthe-art performance for both action proposal and localization on THUMOS'14 detection benchmark and competitive performance on ActivityNet challenge.

CHAPTER 8

Conclusion, Limitations, and Future Work

8.1 Conclusion

This dissertation has contributed to visual understanding of human-object interactions on the recognition and synthesis tasks. For the recognition task, we have studied the mining of semantic HOI categories (Chapter 2), constructed a large-scale image benchmark (Chapter 3 and 4), and investigated DNN based representations for HOI classification and detection (Chapter 3 and 4). For the synthesis task, we have studied semantic-driven synthesis (Chapter 5), by predicting human dynamics from a static image, and goal-driven synthesis (Chapter 6), by synthesizing human-chair interactions in indoor scenes. We have also studied temporal action localization (Chapter 7) for action understanding in video.

Throughout this dissertation, we have treated recognition and synthesis as two independent tasks and made progress on each side in parallel. In fact, the two tasks are closely connected, and the progress achieved on one side will improve the outcome of the other. On the one hand, *recognition can reinforce synthesis*. As shown in Chapter 5, the ability to recognize the pose and objects in the given image is an important pre-requisite for future pose prediction: seeing a person standing upright and holding a baseball bat helps us predict his next move—swinging the bat. On the other hand, *synthesis can reinforce recognition*. This can be illustrated in process of building socially aware robots. Training robots in the real world can be extremely inefficient and risky. The ability to synthesize realistic human actions is a key enabler for creating virtual environments that simulate the human world, where we can train robots efficiently and at the same time teach them to interact and share space with humans before deploying them in the real world.

8.2 Limitations

As mentioned in Chapter 1, action understanding is a broad and challenging field, and we have carefully set the scope of the problem in this dissertation in order to make progress toward this

goal. The approaches we take herein might be appropriate for the considered scope, but might fall short when dealing with more general scenarios. Below we discuss some potential limitations of our approaches.

8.2.1 Exhaustive Pairwise Classification

Our approach to detecting human-object interactions (Chapter 4) requires running a classification network exhaustively on all possible pairs of the detected humans and objects. The time complexity is thus quadratic in the number of detections, which brings efficiency concerns when we have a large number of detected humans or objects. This will only become more problematic if we try to generalize from pairwise interactions to high-order interactions. For example, the complexity will become cubic if we attempt to detect interactions involving one person and two objects (e.g. "cutting bread with a knife"). Consequently, how to efficiently identify interactive relations among candidate objects and prune out non-interactive ones will be a major challenge in scaling up interaction understanding.

8.2.2 Unimodal Prediction with Supervised Training

Our approach to predicting future poses (Chapter 5) relies on conventional deep neural networks and thus the output is deterministic and unimodal. This is reasonable since the scenarios we consider (i.e. sports actions) often have only a single mode future. However, the future is generally multimodal, and certain outcomes are more likely than the others. Our approach is not directly applicable in such scenarios. Moreover, using a supervised learning framework implies that we can only predict scenarios that have been observed during training. This is unrealistic since the future in general has infinite number of possible outcomes, and some of them are extremely rarely observed. Predicting unseen but plausible scenarios (e.g. a PhD candidate jumping out a window during his final defense) represents a key challenge in action synthesis.

8.3 Future Work

Visual recognition and synthesis of human-object interactions remains a challenging frontier in action understanding. Below we discuss possible future research directions based on the progress made in this dissertation.

8.3.1 Recognizing Interactions in Video

This dissertation has only addressed HOI recognition in the image domain (Chapter 3 and 4). Extending the task to the video domain is obviously important, especially for robotics applications.

As video offers temporal information, this will enable the recognition of motion related categories (e.g. "opening a door" versus "closing a door"), which are indistinguishable from only image input. However, this will also pose additional challenges on both data and algorithm fronts. Data annotation becomes more time-consuming if we keep the same type of annotations (e.g. bounding boxes), since browsing video takes a longer time and data annotation has to be performed on a per-frame basis. Besides, the extra dimension in the input increases the challenge of training and inference from a computational perspective. The research question thus involves how to approach the task with weaker supervision and how to improve computational efficiency.

8.3.2 3D Representation for Interactions

Chapter 4 has demonstrated the importance of spatial relations between the human and object for interaction recognition. However, our approach only extracts feature for 2D spatial relations (i.e. the spatial configuration of two bounding boxes). Therefore the extracted feature may suffer from the ambiguity induced by camera projection. For example, consider the case of a person bounding box sitting on top of a bicycle bounding box. Rather than the common case of "riding a bike", it may well be the case that the person is in a far distance behind the bike and thus is not interacting with the bike at all. Since human actions take place in the 3D space, a better representation would be directly encoding the 3D spatial relations. An immediate question is what type of 3D representation should be used for the scene and how to obtain it. One example would be using a skeleton based 3D representation for humans and 3D point clouds for the surrounding environment. The following research question is thus how to learn a compact feature representation for 3D spatial relations from such representation.

8.3.3 Long-Term Prediction

Chapter 5 has addressed pose prediction only in a short time horizon—up to a few seconds ahead. One challenging but important direction is to extend the prediction to longer terms. One possible application is household robot assistants, where the robot needs to predict the next actions of a person in order to help preemptively. For example, a person taking out bottle milk from a fridge is likely to fetch a cup or bowl next, and thus the robot can help fetch them before the person doing so. Predicting long-term future poses many challenges. First, the number of possible futures increases exponentially as the time horizon increases. One thus needs to handle such uncertainty with probabilistic predictions, and more importantly, to allow predictions with multiple modes as mentioned in the last section. Second, longer-term prediction involves action transitions and thus requires temporal modeling of action semantics, e.g. a person washing an apple is likely to peel it next, followed by cutting it with a knife, and finally eating it. How to acquire such knowledge is

an important research question.

8.3.4 Motion Synthesis in Realistic Environments

As the first step in interaction synthesis, Chapter 6 used a simplified simulated environment with only a human and a chair model. The synthesized motion will thus not be realistic for indoor scenes filled with other furniture and household objects. For example, if there is a table in front of the chair, the body parts of the human might penetrate through the table in the current result, since the presence of the table is not simulated. A promising direction is thus to simulate a more realistic environment for learning. Apparently a key research question is how to automatically collect 3D indoor scene models with diverse and realistic layouts. Besides human-object interactions, another interesting direction is to synthesize human-human interactions. This requires simultaneously simulating two humanoid models in the environment. Finally, the ultimate goal is to simulate realistic environments for interactions involving multiple people and multiple objects.

BIBLIOGRAPHY

[1] https://github.com/yjxiong/action-detection.

[2] Bullet Physics SDK. https://github.com/bulletphysics/bullet3.

[3] CMU Graphics Lab Motion Capture Database. http://mocap.cs.cmu.edu.

[4] OpenAI Roboschool. https://blog.openai.com/roboschool/.

[5] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 43(3):16:1–16:43, Apr 2011.

[6] S. Agrawal and M. van de Panne. Task-based locomotion. In *SIGGRAPH*, 2016.

[7] I. Akhter and M. J. Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *CVPR*, 2015.

[8] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014.

[9] A. F. Bobick. Movement, activity and action: the role of knowledge in the perception of motion. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 352(1358):1257–1265, 1997.

[10] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *ECCV*, 2016.

[11] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *SIGMOD*, 2008.

[12] S. Buch, V. Escorcia, B. Ghanem, L. Fei-Fei, and J. C. Niebles. End-to-end, single-stream temporal action detection in untrimmed videos. In *BMVC*, 2017.

[13] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles. SST: Single-stream temporal action proposals. In *CVPR*, 2017.

[14] J. Bütepage, M. J. Black, D. Kragic, and H. Kjellström. Deep representation learning for human motion prediction and classification. In *CVPR*, 2017.

[15] F. Caba Heilbron, W. Barrios, V. Escorcia, and B. Ghanem. SCC: Semantic context cascade for efficient action detection. In *CVPR*, 2017.

[16] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[17] F. Caba Heilbron, J. C. Niebles, and B. Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *CVPR*, 2016.

[18] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, 2010.

[19] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the Kinetics dataset. In *CVPR*, 2017.

[20] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[21] Y.-W. Chao, Y. Liu, X. Liu, H. Zeng, and J. Deng. Learning to detect human-object interactions. In *WACV*, 2018.

[22] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar. Rethinking the faster R-CNN architecture for temporal action localization. In *CVPR*, 2018.

[23] Y.-W. Chao, Z. Wang, Y. He, J. Wang, and J. Deng. HICO: A benchmark for recognizing human-object interactions in images. In *ICCV*, 2015.

[24] Y.-W. Chao, Z. Wang, R. Mihalcea, and J. Deng. Mining semantic affordances of visual object categories. In *CVPR*, 2015.

[25] Y.-W. Chao, J. Yang, W. Chen, and J. Deng. Learning to sit: Synthesizing human-chair interactions via hierarchical control. Manuscript submitted for publication, 2018.

[26] Y.-W. Chao, J. Yang, B. Price, S. Cohen, and J. Deng. Forecasting human dynamics from static images. In *CVPR*, 2017.

[27] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*, 2015.

[28] W. Chen, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen. Synthesizing training images for boosting human 3D pose estimation. In *3DV*, 2016.

[29] W. Chen, C. Xiong, R. Xu, and J. J. Corso. Actionness ranking with lattice conditional ordinal random fields. In *CVPR*, 2014.

[30] X. Chen, A. Shrivastava, and A. Gupta. NEIL: Extracting visual knowledge from web data. In *ICCV*, 2013.

[31] X. Chen and A. L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*. 2014.

[32] A. Clegg, W. Yu, J. Tan, C. K. Liu, and G. Turk. Learning to dress: Synthesizing human dressing motion via deep reinforcement learning. In *SIGGRAPH Asia*, 2018.

[33] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011.

[34] B. Dai, Y. Zhang, and D. Lin. Detecting visual relationships with deep relational networks. In *CVPR*, 2017.

[35] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object detection via region-based fully convolutional networks. In *NIPS*. 2016.

[36] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Q. Chen. Temporal context network for activity localization in videos. In *ICCV*, 2017.

[37] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[38] A. Dave, O. Russakovsky, and D. Ramanan. Predictive-corrective networks for action detection. In *CVPR*, 2017.

[39] D. Davidson. Actions, reasons, and causes. *Journal of Philosophy*, 60(23):685–700, 1963.

[40] V. Delaitre, D. F. Fouhey, I. Laptev, J. Sivic, A. Gupta, and A. A. Efros. Scene semantics from long-term observation of people. In *ECCV*, 2012.

[41] V. Delaitre, I. Laptev, and J. Sivic. Recognizing human actions in still images: a study of bag-of-features and part-based representations. In *BMVC*, 2010.

[42] V. Delaitre, J. Sivic, and I. Laptev. Learning person-object interactions for action recognition in still images. In *NIPS*. 2011.

[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[44] C. Desai and D. Ramanan. Detecting actions, poses, and objects with relational phraselets. In *ECCV*, 2012.

[45] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for static human-object interactions. In *CVPR Workshop on Structured Models in Computer Vision*, 2010.

[46] P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, Y. Wu, and P. Zhokhov. OpenAI Baselines. https://github.com/openai/baselines, 2017.

[47] S. Divvala, A. Farhadi, and C. Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*, 2014.

[48] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[49] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. FlowNet: Learning optical flow with convolutional networks. In *ICCV*, 2015.

[50] Y. Du, Y. Wong, Y. Liu, F. Han, Y. Gui, Z. Wang, M. Kankanhalli, and W. Geng. Marker-less 3D human motion capture with monocular image sequence and height-maps. In *ECCV*, 2016.

[51] V. Escorcia, F. Caba Heilbron, J. C. Niebles, and B. Ghanem. DAPs: Deep action proposals for action understanding. In *ECCV*, 2016.

[52] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan 2015.

[53] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, Jun 2008.

[54] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[55] C. Feichtenhofer, A. Pinz, and R. P. Wildes. Spatiotemporal multiplier networks for video action recognition. In *CVPR*, 2017.

[56] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, Sept 2010.

[57] P. F. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[58] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *NIPS*. 2016.

[59] D. F. Fouhey and C. L. Zitnick. Predicting object dynamics in scenes. In *CVPR*, 2014.

[60] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *ICCV*, 2015.

[61] J. Gao, Z. Yang, and R. Nevatia. Cascaded boundary regression for temporal action detection. In *BMVC*, 2017.

[62] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia. TURN TAP: Temporal unit regression network for temporal action proposals. In *ICCV*, 2017.

[63] P. Ghosh, J. Song, E. Aksan, and O. Hilliges. Learning human motion models for long-term predictions. In *3DV*, 2017.

[64] J. J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.

[65] R. Girshick. Fast R-CNN. In *ICCV*, 2015.

[66] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[67] G. Gkioxari and J. Malik. Finding action tubes. In *CVPR*, 2015.

[68] G. Gkioxari, A. Toshev, and N. Jaitly. Chained predictions using convolutional neural networks. In *ECCV*, 2016.

[69] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. YouTube2Text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, 2013.

[70] L.-Y. Gui, Y.-X. Wang, X. Liang, and J. M. F. Moura. Adversarial geometry-aware human motion prediction. In *ECCV*, 2018.

[71] L.-Y. Gui, Y.-X. Wang, D. Ramanan, and J. M. F. Moura. Few-shot human motion prediction via meta-learning. In *ECCV*, 2018.

[72] G. Guo and A. Lai. A survey on still image based human action recognition. *Pattern Recognition*, 47(10):3343 – 3361, 2014.

[73] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, Oct 2009.

[74] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. From 3D scene geometry to human workspace. In *CVPR*, 2011.

[75] S. Gupta and J. Malik. Visual semantic role labeling. *arXiv preprint arXiv:1505.04474*, 2015.

[76] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[77] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. M. A. Eslami, M. Riedmiller, and D. Silver. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

[78] N. Heess, G. Wayne, Y. Tassa, T. Lillicrap, M. Riedmiller, and D. Silver. Learning and transfer of modulated locomotor controllers. *arXiv preprint arXiv:1610.05182*, 2016.

[79] D. Holden, T. Komura, and J. Saito. Phase-functioned neural networks for character control. In *SIGGRAPH*, 2017.

[80] D. Holden, J. Saito, and T. Komura. A deep learning framework for character motion synthesis and editing. In *SIGGRAPH*, 2016.

[81] R. Hou, R. Sukthankar, and M. Shah. Real-time temporal action localization in untrimmed videos by sub-action discovery. In *BMVC*, 2017.

[82] J.-F. Hu, W.-S. Zheng, J. Lai, S. Gong, and T. Xiang. Recognising human-object interaction via exemplar based modelling. In *ICCV*, 2013.

[83] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *CVPR*, 2017.

[84] S. Huang, S. Qi, Y. Zhu, Y. Xiao, Y. Xu, and S.-C. Zhu. Holistic 3D scene parsing and reconstruction from a single RGB image. In *ECCV*, 2018.

[85] N. Ikizler, R. G. Cinbis, S. Pehlivan, and P. Duygulu. Recognizing actions from still images. In *ICPR*, 2008.

[86] N. Ikizler-Cinbis, R. G. Cinbis, and S. Sclaroff. Learning actions from the web. In *ICCV*, 2009.

[87] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, July 2014.

[88] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-RNN: Deep learning on spatio-temporal graphs. In *CVPR*, 2016.

[89] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[90] Y.-G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/, 2014.

[91] J. Johnson, A. Karpathy, and L. Fei-Fei. DenseCap: Fully convolutional localization networks for dense captioning. In *CVPR*, 2016.

[92] R. Jozefowicz, W. Zaremba, and I. Sutskever. An empirical exploration of recurrent network architectures. In *ICML*, 2015.

[93] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid. Action tubelet detector for spatio-temporal action localization. In *ICCV*, 2017.

[94] S. Karaman, L. Seidenari, and A. D. Bimbo. Fast saliency based pooling of fisher encoded dense trajectories. http://crcv.ucf.edu/THUMOS14/, 2014.

[95] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[96] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classication with convolutional neural networks. In *CVPR*, 2014.

[97] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The Kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[98] F. S. Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, A. M. Lopez, and M. Felsberg. Coloring action recognition in still images. *International Journal of Computer Vision*, 105(3):205–221, Dec 2013.

[99] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. Activity forecasting. In *ECCV*, 2012.

[100] H. Kjellström, J. Romero, and D. Kragić. Visual object-action recognition: Inferring object affordances from human demonstration. *Computer Vision and Image Understanding*, 115(1):81 – 90, 2011.

[101] P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*, 2005.

[102] C. Kong, D. Lin, M. Bansal, R. Urtasun, and S. Fidler. What are you talking about? text-to-image coreference. In *CVPR*, 2014.

[103] H. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. In *RSS*, 2013.

[104] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from RGB-D videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013.

[105] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In *SIGGRAPH*, 2002.

[106] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, May 2017.

[107] N. Krishnamoorthy, G. Malkarnenkar, R. Mooney, K. Saenko, and S. Guadarrama. Generating natural-language video descriptions using text-mined knowledge. In *AAAI*, 2013.

[108] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*. 2012.

[109] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *ICCV*, 2011.

[110] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *NIPS*. 2016.

[111] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[112] T. Lan, T.-C. Chen, and S. Savarese. A hierarchical representation for future action prediction. In *ECCV*, 2014.

[113] T. K. Landauer and S. T. Dutnais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

[114] D.-T. Le, R. Bernardi, and J. Uijlings. Exploiting language models to recognize unseen actions. In *ICMR*, 2013.

[115] D.-T. Le, J. Uijlings, and R. Bernardi. TUHOI: Trento universal human object interaction dataset. In *COLING Workshop on Vision and Language*, 2014.

[116] C. Lea, M. Flynn, R. Vidal, A. Reiter, and G. Hager. Temporal convolutional networks for action segmentation and detection. In *CVPR*, 2017.

[117] C. Li, Z. Zhang, W. S. Lee, and G. H. Lee. Convolutional sequence to sequence model for human dynamics. In *CVPR*, 2018.

[118] L.-J. Li and L. Fei-Fei. What, where and who? classifying events by scene and object recognition. In *CVPR*, 2007.

[119] S. Li and A. B. Chan. 3D human pose estimation from monocular images with deep convolutional neural network. In *ACCV*, 2014.

[120] Y. Li, W. Ouyang, X. Wang, and X. Tang. ViP-CNN: Visual phrase guided convolutional neural network. In *CVPR*, 2017.

[121] X. Liang, L. Lee, and E. P. Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *CVPR*, 2017.

[122] T. Lin, X. Zhao, and Z. Shou. Single shot temporal action detection. In *ACM Multimedia*, 2017.

[123] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[124] Y. Lin, J.-B. Michel, E. L. Aiden, J. Orwant, W. Brockman, and S. Petrov. Syntactic annotations for the google books ngram corpus. In *ACL*, 2012.

[125] L. Liu and J. Hodgins. Learning to schedule control fragments for physics-based characters using deep Q-learning. *ACM Transactions on Graphics*, 36(3), Jun 2017.

[126] L. Liu and J. Hodgins. Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. In *SIGGRAPH*, 2018.

[127] L. Liu, M. van de Panne, and K. Yin. Guided learning of control graphs for physics-based characters. *ACM Transactions on Graphics*, 35(3):29:1–29:14, May 2016.

[128] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.

[129] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *ECCV*, 2016.

[130] S. Ma, L. Sigal, and S. Sclaroff. Learning activity progression in LSTMs for activity detection and early detection. In *CVPR*, 2016.

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

[132] A. Mallya and S. Lazebnik. Learning models for actions and person-object interactions with transfer to question answering. In *ECCV*, 2016.

[133] J. Martinez, M. J. Black, , and J. Romero. On human motion prediction using recurrent neural networks. In *CVPR*, 2017.

[134] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi scale video prediction beyond mean square error. In *ICLR*, 2016.

[135] T. Mensink, E. Gavves, and C. G. M. Snoek. COSTA: Co-occurrence statistics for zero-shot classification. In *CVPR*, 2014.

[136] J. Merel, Y. Tassa, D. TB, S. Srinivasan, J. Lemmon, Z. Wang, G. Wayne, and N. Heess. Learning human behaviors from motion capture by adversarial imitation. *arXiv preprint arXiv:1707.02201*, 2017.

[137] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*. 2013.

[138] G. A. Miller. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, Nov 1995.

[139] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, 2006.

[140] R. Mottaghi, H. Bagherinezhad, M. Rastegari, and A. Farhadi. Newtonian image understanding: Unfolding the dynamics of objects in static image. In *CVPR*, 2016.

[141] T. S. Motwani and R. J. Mooney. Improving video activity recognition using object recognition and text mining. In *ECAI*, 2012.

[142] H.-H. Nagel. From image sequences towards conceptual descriptions. *Image and Vision Computing*, 6(2):59–74, 1988.

[143] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[144] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015.

[145] B. Ni, X. Yang, and S. Gao. Progressively parsing interactional objects for fine grained action detection. In *CVPR*, 2016.

[146] B. X. Nie, C. Xiong, and S.-C. Zhu. Joint action recognition and pose estimation from video. In *CVPR*, 2015.

[147] D. Oneata, J. Verbeek, , and C. Schmid. The LEAR submission at thumos 2014. http://crcv.ucf.edu/THUMOS14/, 2014.

[148] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011.

[149] T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet::Similarity: Measuring the relatedness of concepts. In *HLT-NAACL*, 2004.

[150] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne. DeepMimic: Example-guided deep reinforcement learning of physics-based character skills. In *SIGGRAPH*, 2018.

[151] X. B. Peng, G. Berseth, K. Yin, and M. van de Panne. DeepLoco: Developing locomotion skills using hierarchical deep reinforcement learning. In *SIGGRAPH*, 2017.

[152] X. B. Peng, A. Kanazawa, J. Malik, P. Abbeel, and S. Levine. SFV: Reinforcement learning of physical skills from videos. In *SIGGRAPH Asia*, 2018.

[153] S. L. Pintea, J. C. van Gemert, and A. W. M. Smeulders. Déjà vu: Motion prediction in static images. In *ECCV*, 2014.

[154] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015.

[155] A. Prest, C. Schmid, and V. Ferrari. Weakly supervised learning of interactions between humans and objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):601–614, March 2012.

[156] V. Ramanathan, C. Li, J. Deng, W. Han, Z. Li, K. Gu, Y. Song, S. Bengio, C. Rossenberg, and L. Fei-Fei. Learning semantic relationships for better action retrieval in images. In *CVPR*, 2015.

[157] M. Rastegari, A. Diba, D. Parikh, and A. Farhadi. Multi-attribute queries: To merge or not to merge? In *CVPR*, 2013.

[158] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*. 2015.

[159] A. Richard and J. Gall. Temporal action detection using a statistical language model. In *CVPR*, 2016.

[160] G. Rogez and C. Schmid. MoCap-guided data augmentation for 3D pose estimation in the wild. In *NIPS*. 2016.

[161] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec 2015.

[162] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012.

[163] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011.

[164] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the Fisher Vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245, Dec 2013.

[165] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[166] G. Sharma, F. Jurie, and C. Schmid. Expanded parts model for human attribute and action recognition in still images. In *CVPR*, 2013.

[167] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, 2017.

[168] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage CNNs. In *CVPR*, 2016.

[169] A. Shrivastava, S. Singh, and A. Gupta. Constrained semi-supervised learning using attributes and comparative attributes. In *ECCV*, 2012.

[170] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*. 2014.

[171] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *CVPR*, 2016.

[172] G. Singh and F. Cuzzolin. Untrimmed video classification for activity detection: submission to ActivityNet challenge. In *ActivityNet Large Scale Activity Recognition Challenge*, 2016.

[173] G. Singh, S. Saha, M. Sapienza, P. Torr, and F. Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *ICCV*, 2017.

[174] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng. Parsing with compositional vector grammars. In *ACL*, 2013.

[175] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[176] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *ICML*, 2015.

[177] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009, Jan 2009.

[178] C. Sun, S. Shetty, R. Sukthankar, and R. Nevatia. Temporal localization of fine-grained actions in videos by domain transfer from web images. In *ACM Multimedia*, 2015.

[179] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[180] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua. Structured prediction of 3D human pose with deep neural networks. In *BMVC*, 2016.

[181] C. Tessler, S. Givony, T. Zahavy, D. J. Mankowitz, and S. Mannor. A deep hierarchical approach to lifelong learning in Minecraft. In *AAAI*, 2017.

[182] J. Thomason, S. Venugopalan, S. Guadarrama, K. Saenko, and R. Mooney. Integrating language and vision to generate natural language descriptions of videos in the wild. In *COLING*, 2014.

[183] C. Thurau and V. Hlaváč. Pose primitive based human action recognition in videos or still images. In *CVPR*, 2008.

[184] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*. 2014.

[185] A. Toshev and C. Szegedy. DeepPose: Human pose estimation via deep neural networks. In *CVPR*, 2014.

[186] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015.

[187] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, Nov 2008.

[188] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. SfM-Net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017.

[189] R. Villegas, J. Yang, D. Ceylan, and H. Lee. Neural kinematic networks for unsupervised motion retargetting. In *CVPR*, 2018.

[190] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee. Learning to generate long-term future via hierarchical prediction. In *ICML*, 2017.

[191] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating visual representations from unlabeled video. In *CVPR*, 2016.

[192] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NIPS*. 2016.

[193] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016.

[194] J. Walker, A. Gupta, and M. Hebert. Patch to the future: Unsupervised visual prediction. In *CVPR*, 2014.

[195] J. Walker, A. Gupta, and M. Hebert. Dense optical flow prediction from a static image. In *ICCV*, 2015.

[196] J. Walker, K. Marino, A. Gupta, and M. Hebert. The pose knows: Video forecasting by generating pose futures. In *ICCV*, 2017.

[197] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *ICCV*, 2011.

[198] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[199] L. Wang, Y. Qiao, and X. Tang. Action recognition and detection by combining motion and appearance features. http://crcv.ucf.edu/THUMOS14/, 2014.

[200] L. Wang, Y. Xiong, D. Lin, and L. Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, 2017.

[201] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[202] R. Wang and D. Tao. UTS at ActivityNet 2016. In *ActivityNet Large Scale Activity Recognition Challenge*, 2016.

[203] X. Wang, R. Girdhar, and A. Gupta. Binge watching: Scaling affordance learning from sitcoms. In *CVPR*, 2017.

[204] Y. Wang, H. Jiang, M. S. Drew, Z.-N. Li, and G. Mori. Unsupervised discovery of action classes. In *CVPR*, 2006.

[205] J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman. Single image 3D interpreter network. In *ECCV*, 2016.

[206] C. Xu, S.-H. Hsieh, C. Xiong, and J. J. Corso. Can humans fly? action understanding with multiple classes of actors. In *CVPR*, 2015.

[207] H. Xu, A. Das, and K. Saenko. R-C3D: Region convolutional 3D network for temporal activity detection. In *ICCV*, 2017.

[208] T. Xue, J. Wu, K. L. Bouman, and W. T. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NIPS*. 2016.

[209] K. Yamane, J. J. Kuffner, and J. K. Hodgins. Synthesizing animations of human manipulation tasks. In *SIGGRAPH*, 2004.

[210] X. Yan, A. Rastogi, R. Villegas, K. Sunkavalli, E. Shechtman, S. Hadap, E. Yumer, and H. Lee. MT-VAE: Learning motion transformations to generate multimodal human dynamics. In *ECCV*, 2018.

[211] J. Yang, S. E. Reed, M.-H. Yang, and H. Lee. Weakly-supervised disentangling with recurrent transformations for 3D view synthesis. In *NIPS*. 2015.

[212] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *CVPR*, 2010.

[213] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011.

[214] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2878–2890, Dec 2013.

[215] Y. Yang, C. L. Teo, H. Daumé III, and Y. Aloimonos. Corpus-guided sentence generation of natural images. In *EMNLP*, 2011.

[216] B. Yao and L. Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *CVPR*, 2010.

[217] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010.

[218] B. Yao and L. Fei-Fei. Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1691–1703, Sept 2012.

[219] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *ICCV*, 2011.

[220] B. Yao, A. Khosla, and L. Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *CVPR*, 2011.

[221] B. Yao, J. Ma, and L. Fei-Fei. Discovering object functionality. In *ICCV*, 2013.

[222] T. Yao, M. Wang, B. Ni, H. Wei, and X. Yang. Multiple granularity group interaction prediction. In *CVPR*, 2018.

[223] H. Yasin, U. Iqbal, B. Krüger, A. Weber, , and J. Gall. A dual-source approach for 3D pose estimation from a single image. In *CVPR*, 2016.

[224] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016.

[225] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[226] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S.-F. Chang. Designing category-level attributes for discriminative visual recognition. In *CVPR*, 2013.

[227] J. Yuan, B. Ni, X. Yang, and A. A. Kassim. Temporal action localization with pyramid of score distribution features. In *CVPR*, 2016.

[228] Z. Yuan, J. C. Stroud, T. Lu, and J. Deng. Temporal action localization by structured maximal sums. In *CVPR*, 2017.

[229] J. Yuen and A. Torralba. A data-driven approach for event prediction. In *ECCV*, 2010.

[230] W. Zaremba and I. Sutskever. Learning to execute. *arXiv preprint arXiv:1410.4615*, 2014.

[231] M. Y. L. Zettlemoyer and A. Farhadi. Situation recognition: Visual semantic role labeling for image understanding. In *CVPR*, 2016.

[232] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua. Visual translation embedding network for visual relation detection. In *CVPR*, 2017.

[233] J. Zhang, M. Elhoseiny, S. Cohen, W. Chang, and A. Elgammal. Relationship proposal networks. In *CVPR*, 2017.

[234] W. Zhang, M. Zhu, and K. G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*, 2013.

[235] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, D. Lin, and X. Tang. Temporal action detection with structured segment networks. In *ICCV*, 2017.

[236] Y. Zheng, Y.-J. Zhang, X. Li, and B.-D. Liu. Action recognition in still images using a combination of human pose and context information. In *ICIP*, 2012.

[237] T. Zhou, H. Shan, A. Banerjee, and G. Sapiro. Kernelized probabilistic matrix factorization: Exploiting graphs and side information. In *SDM*, 2012.

[238] X. Zhou, S. Leonardos, X. Hu, and K. Daniilidis. 3D shape estimation from 2D landmarks: A convex relaxation approach. In *CVPR*, 2015.

[239] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis. Sparseness meets deepness: 3D human pose estimation from monocular video. In *CVPR*, 2016.

[240] Y. Zhou, Z. Li, S. Xiao, C. He, Z. Huang, and H. Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. In *ICLR*, 2018.

[241] Y. Zhu, A. Fathi, and L. Fei-Fei. Reasoning about object affordances in a knowledge base representation. In *ECCV*, 2014.

[242] Y. Zhu, Y. Zhao, and S.-C. Zhu. Understanding tools: Task-oriented object modeling, learning and recognition. In *CVPR*, 2015.

[243] C. L. Zitnick and D. Parikh. Bringing semantics into focus using visual abstraction. In *CVPR*, 2013.