CHAPTER

11

Learning for action-based scene understanding

Cornelia Fermüller[a] *and Michael Maynord*[b]

[a]University of Maryland, Institute for Advanced Computer Studies, Iribe Center for Computer Science and Engineering, College Park, MD, United States [b]University of Maryland, Computer Science Department, Iribe Center for Computer Science and Engineering, College Park, MD, United States

CHAPTER POINTS

- An action-centric framework for scene and activity interpretation.
- Studies on object affordances and functionalities and their use in the context of action recognition and robot learning.
- Studies on activity recognition as an interplay between cognition and perception.
- The merging of vision and language through embedding spaces.
- Discussion on the future of action and activity understanding through the lens of the action-centric framework.

11.1 Introduction

The purpose of Computer Vision (CV) is to produce interpretations of images and video which are of use to humans. Action is *important* to model because it is a primary means through which others and ourselves interact with our environment, and it is largely through interaction that the environment becomes meaningful. Because of the centrality of action in what humans find meaningful in their environment, humans structure their environments around action. So to fully understand human environments requires an understanding of their relation to actual and possible action. Most contemporary CV methods are not action

Advanced Methods and Deep Learning in Computer Vision
https://doi.org/10.1016/B978-0-12-822109-9.00020-5

based in their approach – in this chapter we present methods and frameworks which in modeling the observed scene employ an *action based* or *functional* interpretation.

The centering of action in perception aligns with embodied cognition theories (Varela et al., 1993; Barsalou, 2008), which argue that many aspects of cognition take their origin in motor behavior and action. In a computational approach we can leverage action based representation at multiple time scales for a hierarchical approach to scene understanding. At the early hierarchical levels are static components, the objects, humans, and simple movements of the limbs. These are then combined into increasingly more complex notions that involve interactions between scene components. Temporally *actions* chain together in structured ways to constitute *activities*.

The use of action based representations in computational perception approaches is *challenging*. The classic approach to CV is to recognize scene constituents based purely on their appearance. However, the aspects of the scene related to action are often semantic and relational rather than appearance based in nature. To better model interactions, more complicated architectures are required that not only model visual appearance but which leverage a more cognitive understanding of the intermediate semantic and relational structure of action in the input.

Classic end-to-end visual learning becomes intractable with larger input state spaces, as are found with video and action of increasing duration. This is because the variability in visual appearance increases, presenting challenges both in data and in modeling. In order to scale, a more cognitive approach which models not the appearance but the action structure of the activity is necessary.

A primary advantage of the action based approach to scene interpretation is generalization, i.e., the ability to recognize scene quantities beyond those visually observed in the training set. For example, if we can recognize what makes an object usable for cutting, this will allow us to recognize new kinds of cutting tools, such as an Alaskan ulu, although this object has not been in our training set. Similarly, if we can interpret an observed human activity by understanding the interaction of constituents and by understanding the underlying goal, we can be more robust. Individual constituents may be difficult to recognize because of occlusions, size, unfavorable viewing angles or variability in visual appearance and movements, but reasoning about the cognitive plausibility of the activity can allow the recovering from classification errors. Furthermore, action modeling provides the potential to predict far into the future.

This chapter presents CV learning-based approaches and concepts centered on action. We now outline in brief the contents of the rest of this chapter.

Section 11.2 covers *affordances* – a variety of action-based object description. Affordances have been of great interest to Robot Vision. But also classic, nonembodied CV can benefit from the use of affordances. They reflect how the different objects in a scene can be used, and they are an essential component for action understanding. They carry information on the possible cooccurrences of observed objects, humans, and other scene constituents. Section 11.2 covers the best known CV works on the topic, which include early studies that reason about affordances via geometric measurements, studies that learn affordance maps using algorithms for object detection and semantic segmentation on depth and geometric feature maps, and studies that combine affordances with other constituents for action recognition.

Section 11.3 is devoted primarily to our own work on understanding manipulation activities. We argue that activity interpretation should be implemented as a continuous interplay between reasoning and perception processes. Activities are modeled hierarchically. At the lower level are modules for objects, actions, spatial relations, etc, which are merged at the higher level via a grammatical formulation. The grammar and selected modules supporting an action-driven understanding are described.

Section 11.4 focuses on methods that can achieve a tighter integration of appearance and semantic and relational constraints. We consider the integration within the context of the task of Zero-Shot Learning. We cover first simple methods involving engineered attributes, and proceed through more sophisticated approaches involving merging language and vision through shared embedding spaces, capturing semantic and relational information.

This is followed by a discussion on how these concepts could be applied to action and activity understanding in Section 11.5, and Conclusions in Section 11.6.

11.2 Affordances of objects

Psychologist James Gibson coined the term "affordance" (Gibson, 1977), referring to the action possibilities that an object presents based on humans' (or animals') physical capabilities. For instance, a knife affords "cutting," "stabbing," "poking," "slicing," "throwing," etc. (to a human). The notion of affordance has recently received great interest in the cognitive science and neuroscience literature, strengthened by brain imaging evidence that showed that observing tools activate motor areas of the brain (for a review see Martin, 2007). The concept has been studied in different areas, including developmental psychology, industrial design, sport science, and human computer interaction, and there have been many interpretations and discussions on its meaning. Most distinguish between "affordance" and "function," with the former meaning properties of objects and the latter referring to the role that an object plays in satisfying some purpose. For example the handle of a cup affords "grasping," and its interior "containing," while an electricity plug supports the function to "powering kitchen appliances," or "charging devices," and a water faucet supports the function of "getting drinking water." However, a formal definition does not exist.

In this section, we first motivate the use of affordances in CV (Sec. 11.2.1). Then different works from the literature are discussed: Sec. 11.2.2 is about the earlier approaches, which selected geometric features computed from 3D data to classify affordances of chairs or everyday objects. Sec. 11.2.3 describes works on learning affordances of objects and their parts using CV recognition algorithms applied to depth data or geometric feature maps. Sec. 11.2.4 describes approaches using affordances together with other detectors for scene and action recognition, and approaches that learn affordances for embodied agents. Sec. 11.2.5 concludes with suggestions for future work.

11.2.1 Why would computer vision be interested in affordances?

Looking at objects and scene surfaces from the viewpoint of affordances provides information for visual scene interpretation that is complimentary to the classic cues and aids in

robustness and generalizability of learned representations. This information is about the "actionability" that the scene presents at multiple spatial and temporal scales relating to objects, groups of objects, and the complete spatio-temporal scene. Therefore affordances provide information and constraints for scene understanding both in the present and in projecting into the future – thus aiding recognition in addition to prediction, as detailed next.

Models of affordances learned over some objects are transferable to novel object categories. I.e., if our recognition modules can recognize an affordance, they can detect it in objects never seen before, even in a stone that has the right properties. This is because how an object is used depends on physical properties such as its shape, size, material, and weight (Hermans et al., 2011), and we can design processes that pick up these physical properties from images, depth maps, and other modalities, independent of previously encountered object categories. In contrast, classifying objects in images in a conventional end-to-end fashion does not give insight into how visual features such as affordances relate to the object.

Affordances provide valuable information to visual object understanding, such as in understanding the "valid functionality" of objects (Hassanin et al., 2018) – e.g., an inverted cup cannot be used to pour into, or similarly a broken chair cannot be used for sitting (Grabner et al., 2011). Another example is the subcategorization of the classical visual object categories, such as differentiating between chairs for different uses (Stark and Bowyer, 1991).

Since affordances represent the possible actions that can be performed with an object, they carry valuable information for predicting future actions (Koppula and Saxena, 2015; Qi et al., 2018) – because actions relate to each other over time. For example, a bread knife as a whole presents affordances ("graspable", "cut with") allowing the action of "slicing bread," and slicing bread is part of the activity "preparing the bread basket" – an activity consisting of multiple actions extended through time with temporal dependencies. Knowledge of the possibility of "slicing bread" informs possible subsequent actions such as putting the basket on the table. To summarize, affordances and functionalities at the object level also contain information about possible object interactions, spatio-temporal relations, and activities at longer temporal scales. Modeling these relations to get explicit or implicit relations at multiple time scales and semantic levels of abstraction has value for the task of activity understanding.

The concept of affordances has been central to Robot Vision and to research along the Active Vision Paradigm (Bajcsy, 1988). The latter advocates that the vision of systems should not be considered a passive process. Biological systems "move their eyes to select what they see" in an active process. Similarly, artificial embodied systems should be able to change the viewpoint of their cameras in order to select what information to gather from their environment, as different viewpoints present different information. Going further, the paradigm also suggests that embodied systems should avoid employing heavy general-purpose vision processes for all purposes, and only process the information necessary to solve the task at hand (Fermüller and Aloimonos, 1995). Therefore, when a robot or artificial system interacts with objects, often it is more effective to compute what an object can be used for – i.e., compute its affordance and how it can be used – rather than to classify the object according to our language representations. Thus, while the advantages of affordances discussed in this section apply to the classic passive CV formulation, where there is no agent interacting with the environment, a great portion of the research on affordances focuses on Robot Vision.

11.2.2 Early affordance work

Affordances relate to actions. As a consequence they are also grounded in action related physical quantities. For example, an object to sit on or an object to pour into have certain physical quantities, e.g., a certain shape, size, or certain material, etc. All of the earlier approaches utilized such explicit physically meaningful representations in affordance recognition modules.

The first studies used shape and geometry. Stark and Bowyer (1991) proposed the first affordance-based approach to object recognition using 3D CAD models as input. A knowledge-graph, similar to a decision tree, was created to classify chairs and subcategories of chairs (e.g., conventional chair, balance chair, high chair, lounge chair), where the leaves of this graph were procedures for classification of geometric features. These features included relative orientation between surfaces, object dimension, stability, and proximity of surfaces.

Grabner et al. (2011) detected surfaces that afford "sitting," by checking the geometry of a 3D human skeleton model in a sitting pose against the object's geometry. Their features include distance and the intersection of the human's mesh with the object's mesh. The detector was evaluated on Google Warehouse models as well as real 3D data collected with a time of flight camera. For best performance the method was combined with an image based classifier. Similarly, Gupta et al. (2011) modeled affordances in 3D indoor scenes by detecting the regions of the space which allow a human to use it for one of three functions: "laying down," "sitting upright," and "sitting reclined." Like Gupta et al. (2011), they also used constraints based on the occupied 3D space and the contact with a human skeleton. However, their method can take as input images, from which it first derives 3D geometry via learning-based regression methods such as Hedau et al. (2009); Lee et al. (2010).

Hermans et al. (2011) learned the affordances of everyday objects via intermediate representations that encode visual and physical characteristics. Visual characteristics included color, discrete shape, and texture, and physical characteristics included weight and size. Standard classifiers were used in the pipeline, and the approach was demonstrated on seven affordance classes in the robotics domain.

11.2.3 Affordance detection, classification, and segmentation

The problem of recognizing affordances associated with objects and scene surfaces is conceptually similar to the problem of object recognition. A number of recent approaches have used tools from object detection, classification, segmentation, and semantic labeling for affordance localization and recognition. However, these techniques usually were not applied to images, but instead either to RGBD data or to feature maps computed from depth data. This section discusses a few such approaches.

11.2.3.1 Affordance detection from geometric features

This section describes the work of Myers et al. (2015), the first approach applying modern machine learning tools on geometric features. The section details the approach to affordance detection and discusses computational implications.

The focus of the study were tools used in everyday workspaces, and specifically the detection of tool parts associated with different affordances. A dataset (the RGB-D Part Affordance

Dataset) of 105 kitchen, workshop and garden tools was collected. Objects were put on a revolving turntable and recorded with a Kinect camera from a full 360° field of view, about 300 frames for each object, out of which 10,000 RGB-D images were annotated at the pixel level. Fig. 11.1 shows example objects for five of the seven affordances, along with the annotation for one of the objects. It should be noted that affordance is associated with surfaces, for example the inner surface of a cup is "contain" while the outer surface is "wrap-grasp."

FIGURE 11.1 Sample objects from the RGB-D Part Affordance Dataset, and an example of a full frame image with hand-labeled ground truth (at the lower right). The ground truth labels include rankings for multiple affordances (from Myers et al., 2015).

From the raw depth data, shape features were computed patchwise, specifically, the surface normal, principal curvature, shape index, and the HoG-Depth descriptor (histogram of depth gradients). Using these features as input, two classification approaches were proposed: first, a Structured Random Forest (SRF), which creates point-wise classification; and second the S-HMP (Superpixel Hierarchical Matching Pursuit) algorithm (Bo et al., 2013). The latter works by first oversegmenting the RGB-D image into superpixels. Then, using a dictionary learning technique, the shape features are sparsely encoded at multiple scales per superpixel. Finally, the features are max-pooled over the superpixels and classified via an SVM. Example results are shown in Fig. 11.2 for both the S-HMP and the SRF method, where the gray value encodes the probability for the affordance assignment.

There are two computational aspects to the approach discussed above, that deserve special attention. First, sometimes overlooked, the assignment of affordances to object surfaces in general may not be unique. The same object part may be used for multiple purposes. Assigning affordances is thus a multiclass labeling problem. In Myers et al. (2015) this issue was addressed by having multiple annotators rank how close other affordances were with respect to the essential affordance, from which an ordinal scale for affordance assignment at testing was derived.

Second, a main advantage of the approach is its good generalization to new objects and surfaces. Referring to Fig. 11.2 (Bottom), one can see that the bottom of a cup is classified with the affordance "Pounding" and the edge of a spatula with the affordance "Cutting." This is because the shape of these objects indicates these properties. However, shape by itself would not be sufficient for classification in a practical system. One would have to add additional properties, the most obvious is material. This would allow to decide that a paper cup cannot be used for pounding, or an object with a soft edge cannot be used for cutting.

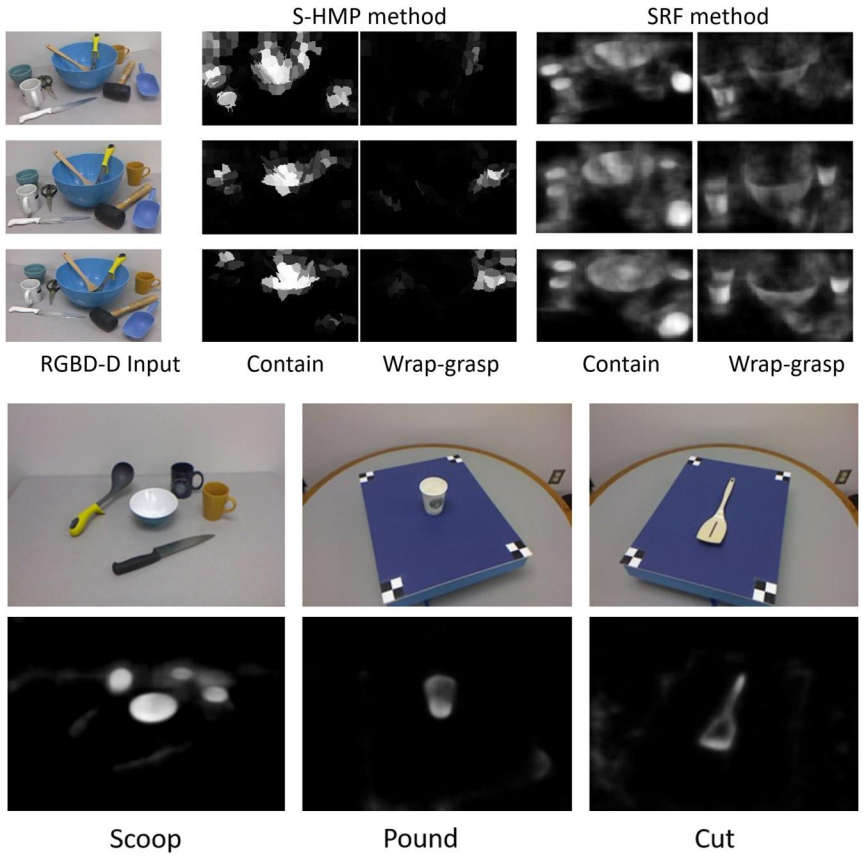

FIGURE 11.2 (Top) Results of affordance detection across three input RGB-D frames using the S-HMP and the SRF method over a cluttered sequence for the target affordances "contain" and "wrap-grasp". Brighter means higher probability of the target affordance. (from Myers et al., 2015) (Bottom) Demonstration of the generalization of the method for new objects for the SRF-method: the bottom of the cup was detected with high probability for "Pounding" and the edge of the spatula with high probability for "Cutting."

11.2.3.2 Semantic segmentation, and classification from images

Many of the works that followed Myers et al. (2015) employed neural network approaches using geometric feature maps as input. Affordance detection at pixel level, thus became a semantic labeling problem. However, different from Myers et al. (2015), these approaches often used 2D images as input. In a preprocessing step, depth maps or feature maps were regressed via neural networks. Furthermore, some considered natural images with multiple objects, and employed object detection algorithms to localize the objects, before assigning affordances.

For example, Nguyen et al. (2017) created a dataset with ten object categories and nine affordance categories from the household and workshop domain. It consists of both RGB-D scans and natural images (a subset of ImageNet (Russakovsky et al., 2015)) – for the latter

depth maps were created using the CNN approach of Liu et al. (2015). The images were annotated with bounding boxes and affordances at the pixel level. The paper's method first applied an object detector, then within each region computed affordances using a modified VGG-16 network trained for semantic labeling, and finally the affordance values were post-processed with a CRF.

Srikantha and Gall (2016) used the dataset of Koppula and Saxena (2014), which features rich contextual information in terms of human-object interactions, and curated it with pixel-level affordance annotations. The work explored different levels of supervision for semantic segmentation, using a deep convolutional neural network within an expectation maximization framework to take advantage of weakly labeled data like image level annotations or keypoint annotations, as well as human pose as context.

Roy and Todorovic (2016) worked with the indoor scenes from the NYU dataset (Silberman et al., 2012). Their approach first infers the depth map, surface normals, and coarse-level semantic segmentation using a multiscale CNN as mid-level cues, which are then jointly fed as inputs to another multiscale CNN for prediction of the affordance maps.

Ye et al. (2017) designed a method for localizing and recognizing functional areas in indoor scenes. An ontology, as shown in Fig. 11.3 (Left), was defined to categorize image regions according to their affordance or functionality. Categories include: "open with spherical grasp" (such as a door knob), "open with wrap grasp or drag to open" (such as an oven door), "turn on electricity" (such as light switch), etc., as shown in the second last column of the figure. The dataset has 500 images featuring kitchens from the SUN dataset (Xiao et al., 2010), which were curated. The method first runs a CCN-based detector trained to detect the region and then a classifier based on a VGG architecture. Fig. 11.3 (Right) shows example results.

	Functional Area	Main Function	End Category	Symbol	Color Code
Functional area ontology	Small part on a furniture/ appliance/ wall	Open	Spherical grasp to open		
			Wrap grasp to open		
		Turn on/off	Turn on/off electricity		
			Turn on/off water		
			Turn on/off fire		
	objects (vessels and tools)	Move	Two hands raise and move		
			Cylindrical grasp to move		
			Hook grasp to move		
			Pinch grasp to move		
		Manipulate	Manipulate elongated tools		
	Furniture	Use of furniture	To sit, to place and etc.		

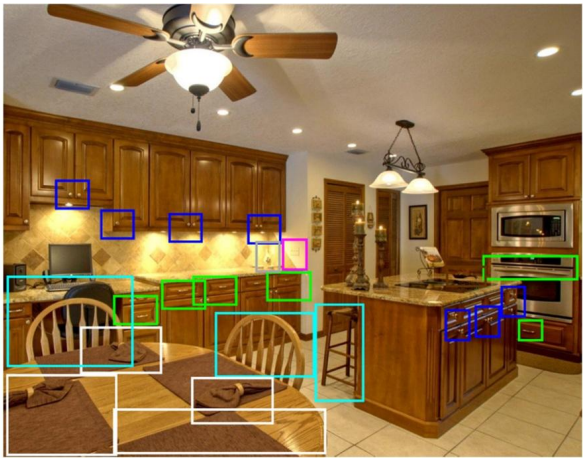

FIGURE 11.3 (Left) Functionality Ontology (Right) Sample detection results (from Ye et al., 2017).

11.2.4 Affordance in the context of action recognition and robot learning

In this section we highlight a few approaches that used affordances in conjunction with other quantities for scene and action understanding. We then discuss approaches that addressed the learning of affordances by robots.

11.2.4.1 Action recognition

Affordances encode features of the possible interactions a human can have with the environment. Thus, naturally they provide a glue between different quantities in the scene in space-time, such as between different objects, or between objects and actions. A number of studies have built on this idea, and used affordance relationships as a context for activity and action recognition and prediction. These methods employed various models to encode relations between the different quantities involved, including CRFs, MRFs, And-Or-Graphs, and probabilistic state automata.

Kjellström et al. (2011) investigated the problem of learning action-object interactions from demonstration, which they define as affordances. Hand actions were classified in the context of the manipulated objects using a CRF that gets as input object and hand features. Objects were modeled using hand-crafted features, and actions were modeled by the hand's global velocity, orientation, and joint angles, which were computed from the output of a 3D hand reconstruction and tracking method.

Koppula et al. (2013) considered the problem of learning sequences of subactivties performed by humans and their interactions with objects. They jointly modeled the human activities and object affordances in a Markov Random Field where the nodes represent objects and subactivities, and the edges represent the relationships between object affordances, their relations with subactivities, and their evolution over time. Affordance-subactivity relations were computed from relative geometric features between the object and the human's skeletal joints, and affordance relations between objects from spatial relations. The description was demonstrated for a PR2 robot in performing assistive tasks. In Koppula and Saxena (2015), Koppula and Saxena added to the Markov Model also possible future states in order to predict the next action.

Qi et al. (2017) used a Spatial-Temporal And-Or Graph (ST-AOG) to represent the structure of activities and predict future actions in RGBD video input. Their model is hierarchical: subactivities are modeled by the human action, the objects, and their affordances in spatial graphs, and a stochastic grammar defined over the subactivities encodes the activity. Dutta and Zielinska (2017) also considered the problem of predicting the next action based on object affordances and human interaction. They employed spatio-temporal based probabilistic state automata to model the interactions. In addition to the action class, they also computed the possible action trajectory. Depending on where an object is relative to the human it has different affordances, and depending on the affordance, its orientation and distance to the possible action trajectories were encoded as heatmaps.

11.2.4.2 Affordance learning in robot vision

Affordance has been a central concept in the field of neurorobotics, which aims to understand the cognition of a system whose body is embedded in the environment. In this research, robots acquire increasingly more complex skills using perception and interaction with the environment. Through interaction robots learn affordances, and build upon these hierarchically

an understanding of actions, activities and the environment. This research in developmental robotics was enabled by the development of robotic platforms, best known among them, the humanoid robot, iCub (Metta et al., 2008).

In Fitzpatrick et al. (2003) the authors discussed three broad stages in the development of a robot: first learning a body image, second learning the interactions with external objects, and third learning to interpret object-object interactions. Affordances are central to the latter two stages. The humanoid robot through pushing and pulling actions in different directions learned to interact, and by observing affected objects' movements learned affordances, such as whether a spherical object is rollable and a cuboid is slide-able. Finally, the robot also learned to mimick an observed action.

Similarly, the authors of Montesano et al. (2008) defined the three main stages in the architecture of a developing humanoid robot as sensory–motor coordination, world interaction, and imitation. Affordances play a central role for world interactions. In this approach, the system started with basic vision and motor skills from which more complex vision and motor skill were acquired using clustering algorithms. Then, during interaction, effects were observed using perception, such as the changes in object position, velocity, and tactile sensing. A Bayesian network was used to learn affordances, which in this case were encoded as probabilistic relations between actions and percepts (object features and effects). The system was demonstrated to imitate the actions of humans by performing movements with similar effect.

Ugur et al. (2011) also demonstrated a robot learning object affordances through interaction and self-observation. In a first step the robot discovered commonalities in its action-effect experience by discovering effect categories. Building upon these, in a second step, affordance predictors for different behaviors were obtained by learning the mapping from the object features to the effect categories. Ugur and Piater (2016) went a step further and studied mechanisms that produce hierarchical structuring of affordance learning tasks. Guided by intrinsic motivation, the robot started with easy tasks, and building on its knowledge of interactions progressively learned more complex tasks by selecting to explore the object and action most different from previously explored ones. For the experiments the robot could compute the visual features of object dimension, surface patch shape, and surface normals, and its actions were poking from three different directions and stacking. In earlier stages it explored the poking actions to observe their effects on single objects. Building upon these, it then explored in a second stage the stacking of two objects and resulting effects.

11.2.5 Discussion on affordance learning

This section discussed approaches to affordance learning, many of which fall into the domain of Robot Vision and have been conducted with few examples and limited amounts of data. So far, deep learning approaches have not been much used for affordance understanding. The major reason is the lack of large annotated datasets in this domain, necessary for deep learning.

However, we expect that as research shifts away from supervised to unsupervised and self-supervised approaches, we will see learning approaches building on the concept of affordances and observed interactions between humans and objects. This will be facilitated by

datasets, such as the EPIC Kitchens dataset, which features a variety of manipulation actions in natural scenes (Damen et al., 2018).

Affordances and functionalities at the object level also encode information about possible object interactions, spatio-temporal relations and possible activities at longer temporal scales. We have discussed in Section 11.2.4 approaches using affordances for action modeling. However, in future work, we could model these relations to get explicit or implicit relations at longer time scales for the problem of activity understanding, the topic of Section 11.3.

Finally, when creating mappings from perception to action for robot learning, we may ground them in affordances. Humans can learn manipulation actions using their perception only. When we see somebody performing actions with a tool unfamiliar to us we can understand the tool's affordance and perform the same action. Similarly, we could approach robot motor learning using perception and action in a tight loop, grounding them in affordances, something that has not yet been done. The robot would learn the task by observing the action and affordances and issuing commands (based on its existing skill set constrained by affordances) to generate the action approximating the observed one, and then adapt gradually to improve performance. The suggested research tasks then amount to developing self-supervised learning and reinforcement-learning approaches grounded in affordance-based representations.

11.3 Functional parsing of manipulation actions

This section describes work – largely from our group – on the interpretation of manipulation activities. Inspired by the embodied cognition paradigm (Varela et al., 1993), this work considers the understanding of human activities a process that involves perception, cognition, and the motor system. The major components are a formalism to combine the different modalities, and CV modules for obtaining semantically meaningful descriptors of action.

11.3.1 The active interplay between cognition and perception

Understanding human actions and activities is the most challenging task currently studied in CV. It is not a task of vision only. Humans can understand what others are doing, because they have models of actions and activities. They understand the goals of actions, and this allows them to interpret their observations despite the large variations in which actions can be executed and variations in visual conditions. Knowledge of some form comes into the interpretation process quite early.

We observe that human behavior is active and exploratory. We continuously shift our gaze to different locations in the scene. We recognize objects and actions and this in turn leads us to fixate at new locations. In this process, perception continuously interacts with cognition at different levels of abstraction: to guide attention, to make predictions, to constrain the search space for recognition, and to reason over what is being perceived. We call this interaction between perception and higher level processes the *Cognitive Dialogue* (Aloimonos and Fermüller, 2015), as it amounts to an iteration of questions and answers, with the cognitive or linguistic processes asking questions about the what and where of quantities in the scene,

and the visual processes performing localization, detection, recognition, and reconstruction. A possible simple way to selecting the next question would be by using information-theoretic criteria (Yu et al., 2011).

The reasoning can be implemented through knowledge-based engineering (Aditya et al., 2018) or the use of language. There has been much interest in CV to introduce additional higher-level knowledge about image relationships into the interpretation process. While many studies get this additional information from captions or accompanying text, others (as discussed in Section 11.4) use advanced language processing to obtain additional high level information. In current research, most commonly, the Word2vec space (Mikolov et al., 2013) (see Sec. 11.4.3) is used as language representation, which encodes similarity about linguistic concepts. Alternatively, one could use older, hand-crafted resources encoding lexical semantics, for example the Word-Net database (Miller et al., 1990), which relates words through synonymy (words having the same meaning, like "argue" and "contend") and hypernymy ("is–a" relationships, as between "car" and "vehicle"), among many others. Verbnet (Schuler, 2005), which organizes verb classes, is particularly interesting for action understanding.

11.3.2 Grammars of action

Various mechanisms have been used to encode relationships between the different semantic concepts, that is between actors, objects, verbs, spatio-temporal relations, and attributes. Section 11.2, discussed the use of and-or-graphs and Markov models. Others include Markov Logic Networks (Tran and Davis, 2008), and planning tools (Guha et al., 2013). In this section we describe work on grammars, which can capture the composition of observed activities as sequences of scene constituents, and their recursive structure.

The main motivation for the use of grammars originates from the idea that actions observed in a video have syntactic structure. By considering the goal of actions, the video can be broken into meaningful segments, and these segments together can be organized in the form of a simple grammar. Thus, interpreting the action that is taking place in a video is like understanding a sentence that we read or hear. To parse the video into the primitive actions that constitute complex tasks, the segments of the video are mapped to particular symbols involving objects, tools, movement, and spatial relations. Importantly, the action grammar temporally segments a video at contact, that is, when the hand touches or releases an object, or objects merge or separate. At these points in time a new subaction starts. In applying the grammar for the analysis of a video, a parse tree is produced, which we call the *activity tree*. Fig. 11.4 illustrates the concept. From a video recording of a person performing the activity "cutting a plank," a graph is created, in which nodes of hands, objects, and tools merge into a common node, whenever they touch, or nodes split when the objects and hands separate. Referring to the figure, to compute the quantities involved, different processes (shown in the four subareas of the video frames) extract the human's body, the hands, objects, and geometric relations.

We now discuss several grammar approaches in Section 11.3.2.1, and then discuss in Section 11.3.2.2 whether such grammar representations are sufficiently expressive to capture action and activity structure, and sufficiently parsimonious to be preferred over other representations.

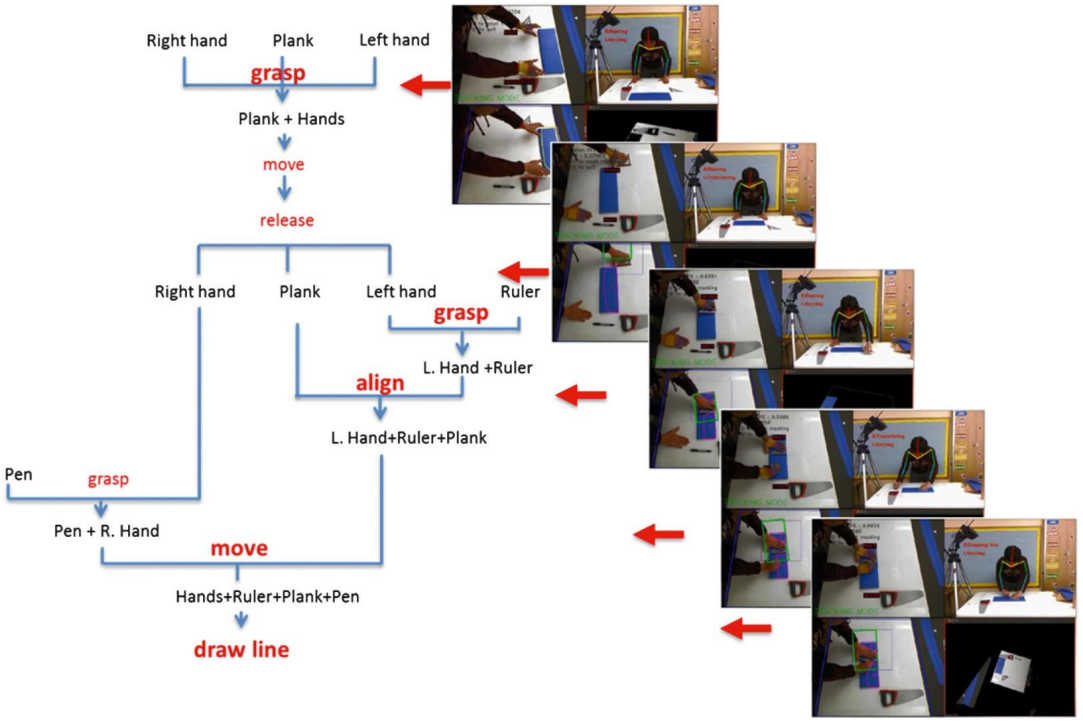

FIGURE 11.4 Illustration of activity description. The camera monitored a person cutting a plank. Four parallel processes computed essential components: (left up) detection of the hand and classification of the grasp type, (right up) gross motion via skeleton fitting, (left down) segmentation of the object, (right down) 3D shape description of the scene.

11.3.2.1 Different implementations of the grammar

The descriptions are based on context-free grammars, originally introduced in Pastra and Aloimonos (2012). Summers-Stay et al. (2012) implemented the idea for parsing assembly actions from RGBD video using only one symbol for all actions, and Yang et al. (2014) enhanced the description introducing grasp into the description and differentiating hand-to-object contact and object-to-object contact. The grammar describes actions at a level of abstraction that is useful for both video interpretation and robot execution. In Yang et al. (2015b) this was demonstrated with some examples. By automatically parsing videos that feature cooking instructions from the Youcook dataset (Das et al., 2013), actions were parsed, which then were performed by a Baxter robot equipped with the necessary motion capabilities.

Abstract descriptions of actions/verbs are necessary to achieve generalization, and perform what is called in the current terminology, few-shot learning or zero-shot learning (see Section 11.4). The basic action grammar reduces the description of actions to only the sequence of "touching relations," that is when the hand touches an object, two objects touch, the hand releases an object, or two objects or pieces of a single object separate (Dessalene et al., 2021).

Wörgötter et al. explore this concept to formulate an ontology (Wörgötter et al., 2013) considering one-handed actions. At the first level, actions are categorized according to the sequence of relations that the hand and one or two objects can have, into six classes, which are: rearrange, destroy, break, take-down, hide, construct. From there they iterate possible actions, and they come up with about 30 fundamental manipulations. Yang et al. (2013) proposed a related concept. They metaclassified actions according to the consequence an action has on an object, that is, what happens geometrically or topologically to an object. They proposed six categories – divide an object, merge two parts, transfer an object, deform an object, object appears, object disappears from the scene – and they also provided algorithms that combine tracking with segmentation to detect topological changes to detect the essential events in video.

11.3.2.2 Are grammars expressive and parsimonious descriptions?

An important question is whether the grammatical representations actually are sufficiently rich to allow for classification of many activities. The authors of Wörgötter et al. (2020) performed psychophysical and computational experiments to answer this question. They described, as above, actions by the sequence of contacts, using five quantities: the hand, the ground, and three objects. In addition they considered ten spatial relations, i.e., above, below, between, etc., to differentiate between altogether 35 different configurations or actions. A subset of ten of these actions (put, shake, stir, take, uncover, chop, cut, hide, lay, and push) was performed in a virtual environment, but instead of the actual objects, cubes were used. Experiments found that humans can recognize these actions, and so can their algorithms. Even more, the description was found to be very powerful for prediction; subjects on average only required 56% of the action duration to recognize the action. Thus, it appears that a description relying only on contact and spatial relations is very powerful for visual recognition.

11.3.3 Modules for action understanding

Individual vision processes are required to recognize discrete components, which then can be combined in higher level reasoning processes – such as the grammars from Section 11.3.2 – to achieve activity recognition and prediction. The descriptors which we discuss in this section differ from those heavily covered in the literature (for an evaluation of successful concepts in current approaches see Sigurdsson et al. (2017)). Specifically, in Section 11.3.3.1 we discuss representations of grasp, and in Section 11.3.3.2 we discuss explicit representation of geometry.

11.3.3.1 Grasping: an essential feature for action understanding

The grasp type provides crucial information about actions. As a motivational example, consider the two scenes in Fig. 11.5 from the VOC challenge (Everingham et al., 2010). Standard CV systems have object and human detectors to recognize the bicycle and the cyclist and pose detectors to confirm that these two cyclists are riding a bike. But humans can tell that the cyclist on the left side is not racing (since his hands are in a "Rest or Extension" grasp), whereas the one on the right is intent on racing (since the hands firmly hold the handlebar in a "Power Cylindrical" grasp).

FIGURE 11.5 (Left) Rest or Extension on the handlebar vs. (Right) Firm power cylindrical grasp of the handlebar (from Yang et al., 2015a).

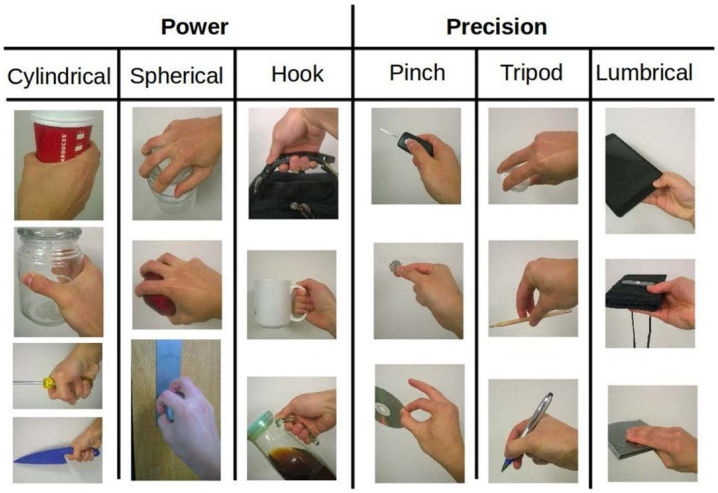

FIGURE 11.6 Basic classification (Cutkosky, 1989) of active grasps with examples. At the highest level, grasps are categorized into power and precision grips. Power grips are used when an object is held with force, and can be classified as cylindrical, spherical, and hook. Precision grips provide fine movement and accuracy, and are subdivided into pinch, tripod, and lumbrical.

We cover here two papers, the first employs a basic ontology of grasps types in action understanding tasks, the second studies subtle changes in grasp for differentiating between similar manipulation actions, and develops learning approaches for online action prediction and regression of associated finger forces.

The recognition of grasp type provides essential information for a more detailed analysis of action. (See Fig. 11.6.) Researchers in several areas, including robotics, developmental medicine, and biomechanics, have developed grasping taxonomies that represent a hierarchy

of the most common hand postures used for object grasping, with each taxonomy based on the needs of the tasks in the field. In Yang et al. (2015a) a basic classification of the main functional grasps (Cutkosky, 1989) in manipulation tasks was used and then demonstrated as a useful feature in two tasks: for segmentation of activities involving fine motor actions, and for characterizing action intention, i.e., whether the task is casual, or requires skills or forces.

Cognitive studies showed that an actor's intention shapes his/her movement kinematics during movement execution (Ansuini et al., 2015). For example, when subjects grasped a bottle for pouring, the middle and the ring fingers were more extended than when they grasped the bottle with the intent of displacing, throwing, or passing it. Inspired by these findings, in Fermüller et al. (2018), the authors developed a recurrent neural network architecture that monitors hands for predicting actions. Specifically, they considered sets of actions with the same object, such as "squeezing", "flipping", "washing", "wiping" and "scratching" with a sponge (see Fig. 11.7). They analyzed the system, which predicted in real-time the ongoing action to determine at what point in time the classification became accurate, and they also performed a psychophysical experiment, evaluating human performance on the same task. At 10 frames after the contact of the hand with the object, the system and the humans started understanding the action (75% classification accuracy for the sponge actions), and at 25 frames the judgment was very good (95% accuracy for the sponge actions). The visual architecture was an RNN using as input tracked image patches around the hand from which VGG-16 features (Simonyan and Zisserman, 2014) were computed.

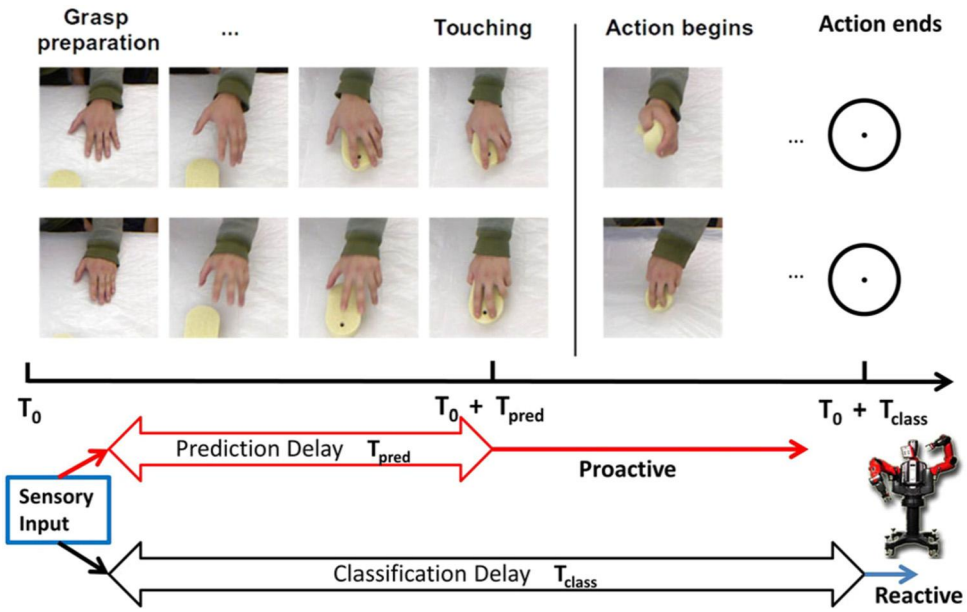

FIGURE 11.7 Examples demonstrating that early movements are strong indicators of the intended manipulation actions. Early prediction of action significantly reduces the delay in real-time interaction, which is fundamentally important for a proactive system.

In addition the paper also demonstrated association of vision with forces. Data was recorded from subjects that performed the same action with both hands. Sensors on the fingers of one hand recorded the forces, and the other hand was recorded visually. A recurrent neural network was trained to regress from vision to forces. It was then shown that using as input video only, when the visual classifier was combined with regressed forces, improved performance could be achieved. The concept appears promising. As shown, learning the mapping from vision to forces creates a bi-modal space that can aid visual recognition. Furthermore, there are immediate implications for robotics. Currently robots rely on haptic devices or force and torque sensors to learn tasks. If we can predict the forces exerted by the human demonstrator visually, it would allow us to teach robots much more efficiently.

11.3.3.2 Geometry to robustify

The use of geometry is important because it provides robust information for scene description and additional information for recognition. Geometry is computed using reconstruction processes, which are low-level (requiring only the image features and knowledge of camera positions), but no machine learning or training data is required. With the advent of cheap RGB-D sensors more than a decade ago, reconstructing scene geometry has become much easier and more accurate, and thus these sensors have become the standard vision sensors in robotics. Their use facilitates computing accurately and fast the distances to control the robot's movement, as well as computing the geometry and shape of objects to aid scene interpretation. This section discusses three geometric methods: accurate tracking of nonrigid object transformations and detection of topological changes; computation of pairwise spatial relations of objects over time; and computation of object symmetry and its use for better foreground-background segmentation.

Building on an efficient point cloud library (Zampogiannis et al., 2018), Zampogiannis et al. (2019) developed a technique for accurately tracking nonrigid object transformations and detecting the topological changes, that is, contacts and separations of body parts and objects needed for the grammatical description (see Section 11.3.2). The gist of the method lies in a warp field estimation that considers forward and backward warps between consecutive frames to detect regions of the deformed geometry that undergo topological changes.

Activity descriptions can also profit from descriptors of spatial relations between objects. In Zampogiannis et al. (2015), the authors introduced a representation for manipulation actions based on the evolution of the spatial relations between objects in the scene. The method was implemented by tracking objects in RGBD video, and reasoning over the spatial relations of observed object pairs. The resulting descriptor amounts to a sequence of spatial relation predicates (e.g., in, left, right, front, behind, below, above, touch), and this descriptor was shown to be sufficiently expressive for distinguishing between four different actions.

Another concept is to exploit general knowledge of object shape properties. For example, symmetry detection can help with segmentation both in 2D (Teo et al., 2015) and 3D (Ecins et al., 2016). Imagine looking at a cluttered scene. Since most objects we work with are symmetric, either bilateral or rotational, we can "fill in the back of the object" that is not visible, and this aids the segmentation and recognition.

11.3.4 Discussion on activity understanding

Activity Understanding is a very challenging problem. End-to-end solutions do not scale well because of the large variations in appearance at high levels of abstraction and temporal extension.

We discussed hierarchical approaches, and we detailed one higher level description – action grammars. Action grammars can segment activities at times of contact and capture the recursive structure of action sequences, similar to that found in language. We described experiments demonstrating the expressive power of action grammars. We also described processes at the lower level, which have not received much attention in CV, but which are essential for supporting an action-based approach. These include affordances, grasp-type and the use of geometry to aid temporal and spatial segmentation and descriptions of spatial relations.

In this section we emphasized the necessity to utilize meaningful action-based representations at different levels of the hierarchy. To elaborate further, it is very important that these representations need to be robust, because of the many challenges involved in activity understanding. We can achieve robustness in part through the use of geometry – geometry does not require memorization, and can be estimated from low-level measurements. Thus, we should introduce geometry into the pipeline whenever possible before starting with recognition. Beyond geometry, any concept that provides universally true information is meaningful for activity understanding. We may model physical laws. We may include model ontologies to aid generalization, for example by grouping verbs according to the effect they have on objects (Yang et al., 2013). We also may include processes that model causality; actions constrain each other causally, some combinations are not possible physically. These representations capture more knowledge and better constraints for activity interpretation.

The integration of vision and cognition or language is hard. This is because of the semantic gap, i.e., the disparity between the symbolic or linguistic representations and the visual representations based on signals. We want an integration of the two which is not brittle. Thus, we need to avoid setting thresholds or converting to purely symbolic representation too early within the pipeline. This is because if vision fails to return the right quantities, then imprecisions compound through further abstraction, resulting in failed reasoning. The next research challenge is to study learning approaches that relate perception with higher-level reasoning for a deeper integration. Section 11.4 covers deep learning approaches useful for such integration. Currently such approaches are primarily constrained to object recognition, and to an extent to action recognition. Activity understanding would benefit from these methods.

11.4 Functional scene understanding through deep learning with language and vision

Here we consider the merging of vision and language and associated representations – the merging of "signal" and "symbol". The merging of multiple representations is important due to the suitability of different representations for capturing different characteristics of the world. Here we seek to allow information from lower level representations of appearance

and information from higher level representations of relations and semantics to complement each other.

Systems involving symbolic and continuous representations often have hard boundaries, below which the system is continuous, and above which the system is symbolic. Precisely where this boundary is set varies, but generally does not fall below the level of abstraction reflected in human language as language is a primary source of symbolically represented world knowledge.

Symbolic representation is more important for action and activity understanding than for other CV tasks such as object detection as the nature of this task is more abstract and less appearance based. Action has temporal structure at multiple scales, and is structured around satisfaction of conditions – the defining characteristics of action are semantic and relational.

Many computer vision tasks can benefit from the integration of vision and language. However, one task which is ideal for the study of the integration of the two is Zero Shot Learning (ZSL) – this is because unlike other tasks it cannot be solved without the introduction of nonvisual knowledge such as is reflected in language.

ZSL is a task with two sets: a training set, and a test set. The categories of these sets break into "seen" and "unseen" categories. The training set consists of only "seen" categories, while the test set contains "unseen" categories as well as, optionally, "seen" categories. To illustrate, there could be a ZSL task which includes "run" and "stand" in seen categories, and "walk" in unseen categories. The task would then be to learn to visually recognize and properly categorize walking, when walking has never previously been visually encountered, but running and standing have been visually encountered.

There are multiple approaches to ZSL. Early work on ZSL focuses on *attributes* – visually recognizable characteristics with differential class (e.g., object classes or action classes) associations. Attributes are generalizable in that attribute detectors trained only on the seen set are able to detect attributes in samples from both the seen and unseen set.

More recent work on ZSL tends to focus on *semantic embedding spaces*. These are Euclidean vector spaces where semantic categories, such as reflected in language, are associated with vectors – or points in space. These vector representations of words are of significantly lower dimension than naive 1-hot encodings (vectors where each dimension corresponds to a class, and all but 1 dimension have 0 values), and have the quality that words which are similar in semantics are associated with similar vectors nearby in the embedding space.

With semantic embedding spaces, symbolic representations are vectorized in such a way that the semantic relations of the symbol categories are preserved. This allows for integration of symbolic semantics into deep architectures, whose internal representations consist of vectors. This integration amounts to properly aligning the visual vector representations with the vectorized symbolic representations.

Different ZSL methods use different shared embeddings – some embed visual features into the semantic space, some embed the semantic space into the visual feature space, and some embed both into a third shared space. Once both visual and semantic representations lie in the same space, categorizing visual input is a matter of finding the nearest semantic label in this space.

State-of-the-art contemporary ZSL methods often rely on CNNs for visual features and produce shared embedding spaces with pretrained semantic embedding spaces such as word2vec (Mandal et al., 2019; Xian et al., 2018). Some shared embedding based ZSL meth-

ods structure and train their models in an end-to-end fashion, which is more challenging, but provides dividends in performance (Zhang et al., 2017).

The remainder of this section is structured as follows: In Section 11.4.1 we detail a simple use of attributes for ZSL; in Section 11.4.1 we detail a more nuanced *relative attribute* formulation; in Section 11.4.2 we cover the use of shared semantic spaces in ZSL; in Section 11.4.3 we cover basic approaches to semantic vector space construction; in Section 11.4.4 we cover the incorporation of knowledge in the form of graphs for ZSL action classification.

11.4.1 Attributes in zero-shot learning

The use of attributes – including action centric attributes such as affordances covered in Section 11.2 – in recognition allows the construction of classifiers which are humanly interpretable and specifiable. Attributes mitigate the issue of opacity through use of an explicit predefined mid-level representation below the level of class categories.

Use of attributes allows an easy mechanism through which to learn visual representations from the available training data and to transfer those representations onto the classes for which no training data is available. Attributes are general in that they have presence across multiple class categories, and through taking multiple attributes with different distributions across classes allow representation of select classes.

The use of attributes in ZSL is as follows: Attribute detectors are trained over the seen set and associated attribute labels. These detectors are generalizable across both the seen and unseen set. Due to the different class coverage (e.g., object or action class) of different attribute categories, different combinations of attributes can represent different classes – class detectors can then be instantiated on top of attribute detectors. This instantiation follows a specification of which attributes are associated with which classes. When given a specification for unseen categories, detectors can be built even though no visual samples of the unseen categories have been encountered.

Conventional use of attributes in computer vision is binary: an attribute is represented as either present, or absent. This limits the representational power of attribute representations. However, binary representations can be generalized to scalar representations where each attribute is associated with a scalar degree rather than a binary category. This is both more flexible representationally, and allows the inclusion of attributes which do not so cleanly fall into a binary categorization. For example, while an attribute of "indoors / outdoors" often is clearly binary, an attribute of "moving fast / moving slow" has a more even distribution over gradations in visual input.

Parikh and Grauman (2011) presents one approach to generalizing binary attributes to scalar attributes. A challenge in generalizing from binary to scalar attributes is inconsistency in annotations, as different annotators may have different understandings of what different attributes' degrees correspond to in the scalar representation. They resolve this challenge by requesting that annotators not assign scalar values to attributes, but rank images in terms of attribute degree. After images are ranked, scalar attribute values can be derived from their annotated relative attribute degrees.

For binary attributes, attribute detectors can be trained using conventional classifiers, but producing scalar attributes requires other methods. Parikh and Grauman (2011) train a rank-

ing function over images for attributes over the seen set, and use that ranking function in the production of a scalar value.

Relative attribute representations allow for greater flexibility in class specifications. With consideration to the attribute of "moving fast / moving slow" one can specify that an unseen category "running" is faster than the seen category of "walking", or that the unseen category of "standing" is slower than the seen category of "walking". This is done without a need to define either a binary specification or intuit a scalar value with which to describe the unseen categories.

11.4.2 Shared embedding spaces

The generalization of binary attributes to scalar attributes increases their representational power. However, the increased representational power came at the cost of increased difficulty in annotating attributes and specifying classes. Relative Attributes (Parikh and Grauman, 2011) introduced one solution to these challenges.

Attributes can also be abstracted away. It need not be the case that visual representations move through humanly understandable attributes. Abstracting attributes away has a couple advantages:

- Avoid the imprecision introduced by passing through engineered, rather than learned, representations.
- Avoid the overhead and imprecision of annotating attributes.

The question is then how to construct a system such that training for classification of seen classes results in a classifier which works not only for seen classes, but for unseen classes as well, without leveraging an engineered mid-level representation which allows the specification of unseen classes in terms of that mid-level representation.

One approach that can be used is to define unseen classes in terms of their similarity relations to seen classes, where these relations are learned from text corpora. Extensive work exists in Natural Language Processing (NLP) producing *semantic vector spaces* which represent similarity relations among words – word2vec (Mikolov et al., 2013) is one popular example. The intuition is that terms in these spaces are located in proximity to other terms with which they share semantic similarity. Trained semantic spaces are publicly available – these can be taken and used without a need to produce them from scratch. See Section 11.4.3 for details on construction of such spaces.

Terms with semantic similarity often share similarity in the visual space as well – e.g., "jogging" is both semantically and visually part way between "running" and "walking". And so, it is often the case that if the visual similarity to known categories can be determined, then the semantic similarity relations can be established as well. Then, from these semantic relations we can infer semantic categories of visual inputs.

To illustrate: Consider a set of seen classes including "running" and "walking", and a set of unseen classes including "jogging", a semantic vector space capturing semantic proximity between these categories, and a computer vision architecture, such as a CNN, which produces visual feature representations of input. Then, consider an input sample of unknown class. Say that the visual representation of this input as produced by the CNN is part way between the visual representation for "running" and for "walking". We then go to the semantic language

space and see that the label that is located part way between "running" and "walking" is jogging and assign this label to the input.

Most often, comparing similarities between samples of unknown class and known classes is not done in the visual space. It is more common to map the input into the semantic space and perform comparisons to known classes there. This requires embedding one space into another.

The mechanism of embedding one space into another can be as simple as a linear transformation applied over one space, which is then trained over a similarity loss between two spaces. DeViSe (Frome et al., 2013) is a good example of an architecture using this method. This involves two pretrained representations – visual features taken from, for instance, a CNN trained over classification, and word vectors in an embedding space, which may be produced through means discussed in Section 11.4.3.

One method of embedding visual features into the semantic space of the word vectors is then illustrated in Fig. 11.8. A layer of nodes is appended to the top of the pretrained CNN features, and then trained. The loss that this layer is trained over is the similarity (e.g., cosine similarity) between the output of this layer, and the vectors in the semantic embedding space corresponding to the labels of the visual input. This learns a simple linear mapping from visual feature vectors onto text derived semantic features.

More sophisticated mappings than linear translations are often used. Using multiple layers of neurons, in conjunction with nonlinear activations, produce nonlinear mappings (e.g., Kato et al., 2018 use such a method). Kodirov et al. (2017) use an autoencoder with semantic constraints to produce the embedding, and find that the constraint of visual reconstruction leads to better generalization to unseen classes in ZSL.

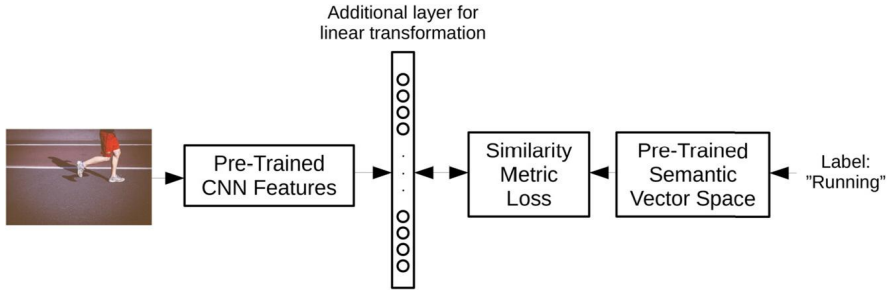

FIGURE 11.8 A simple approach to embedding visual representations into a semantic space, as is used by methods such as DeViSe (Frome et al., 2013). Two pretrained models are used: 1) pretrained visual features, such as produced by a CNN trained over a visual task, and 2) a semantic vector space as produced through methods discussed in Section 11.4.3. A simple linear transformation is produced through appending a layer of nodes on top of the visual features, and training them w.r.t. their similarity to the semantic word vector of the labels corresponding to visual input. Once this linear transformation is learned, it constitutes a mapping from the space of visual features into the semantic vector space.

Once both visual input and semantic representations are placed in the same space, then determining the class of novel visual input is as simple as representing visual input in that space and then finding the nearest semantic label in that same space.

11.4.3 Construction of semantic vector spaces

11.4.3.1 word2vec

Here we wish to construct a vector space into which words are embedded in such a way that their semantics are captured spatially. The end results are a space where, for example, vectors for "jog", "run", and "walk" are located in proximity, with "jog" being placed in the middle of the three.

How can a word's semantics be defined? One answer is through the interactions that word has with other words in text corpora – semantics of words can be defined in relation to other words. How do we model words' relations to other words? One simple approach is cooccurrence – if two words occur in proximity, they are taken to be related. The more frequently that they cooccur together, the more strongly they are related. This is the principle on which methods embedding words into vector spaces such as word2vec are based.

We start from the simplest vector representation for words – a 1-hot encoding, where each vector position corresponds to one word in a vocabulary V. This is a high-dimensional, inefficient representation, where there are no meaningful spatial relations among words. We can embed words into a lower dimensional space with said desirable properties through the solving of one of two related tasks:

1. Predicting a target word based on context
2. Predicting context based on a target word

Here "context" C is defined as the set of terms "nearby" the target term. These are defined as the other terms present in an n-gram associated with the target term, without consideration for word order.

Each of these tasks can be solved using simple architectures, and the solving of these tasks produce in the process lower dimensional representations which can then be taken and used for other tasks.

A method – *Continuous Bag Of Words* (CBOW) – for solving Task 1 (Mikolov et al., 2013) is shown in Fig. 11.9(a). Input consists of multiple words of context, each represented as a 1-hot vector of dimension $|V|$. These vectors are summed to produce a vector of size $|V|$. This is fed through a single layer of N neurons, where N is the dimension of the embedding space. On top of this we have one more layer, of size $|V|$, whose job it is to predict, in 1-hot representation, the word associated with the context consisting of the terms fed as input to the first layer.

A method – *Skip Gram* – for solving Task 2 (Mikolov et al., 2013) is shown in Fig. 11.9(b). Input consists of a single term, represented as a 1-hot vector of dimension $|V|$. This is fed into a single layer of dimension N, where N is the dimension of the embedding space. On top of this layer we have an output layer, consisting of $|C|$ sets of $|V|$ nodes, each associated with one term of context C in the n-gram associated with the input term.

Both architectures are trained through serially feeding in n-grams extracted from large text corpora, and applying a soft-max loss to the final layer to enforce alignment with expected terms.

After these architectures are trained, the hidden layer then constitutes a mapping of V from a 1-hot representation of size $|V|$ to an embedded representation of size N where terms are spatially located in proximity to terms with similar semantics.

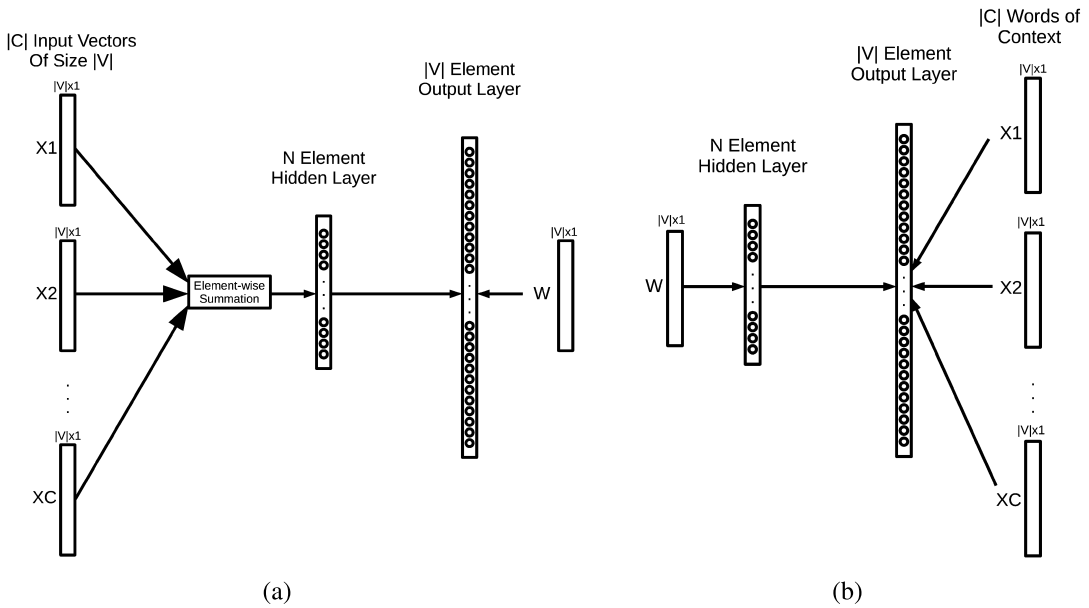

FIGURE 11.9 (a) The Continuous Bag Of Words method to solving the task of predicting a target word based on that words context (Mikolov et al., 2013). $|C|$ input vectors of context are fed through two layers, the first a hidden layer of size N, the second an output layer of size V. Loss is computed w.r.t. the word W. After training the output layer can be discarded, and the hidden layer used as a translation from 1-hot word encodings into an embedded space of size N. (b) Skip Gram method to solving the task of predicting the context of a term (Mikolov et al., 2013). Word W is fed through two network layers, the first a hidden layer of size N, the second an output layer of size V. Loss is computed w.r.t. $|C|$ words of context. After training, the output layer can be discarded, and the hidden layer used as a translation from 1-hot word encodings into an embedded space of size N.

11.4.4 Shared embedding spaces and graphical models

ZSL of action can benefit from additional structure beyond what simple mapping into a shared embedding space can provide. Additional structure can be represented in the form of graphs. Multiple works (Ghosh et al., 2020a,b; Kato et al., 2018; Yan et al., 2018) employ graphs for the purpose of recognition of actions, processing these through use of a Graph Convolutional Network (GCN) to produce vector representations of action categories suitable for ZSL.

Ghosh et al. (2020a) evaluate 3 different graphical representations, the last of which is applicable to Few-Shot Learning rather than ZSL due to its use of visual features from a small number of samples. Kato et al. (2018) construct a graph based on Subject Verb Object triplets derived from knowledge corpora. All of these are processed through a GCN.

Throughout (Ghosh et al., 2020a) sentence2vec (Pagliardini et al., 2017) is used rather than word2vec, as authors find that action categories are better represented through phrases than through single words which can be confounded by multiple meanings. The first graph is composed of nodes taking values of sentence2vec embeddings of action category phrases. These nodes are linked together based on cosine similarity between vector representations – the top N closest neighbors per node are given edges. The second graph associates verbs and

nouns – derived from Part-Of-Speech tagging of phrases describing actions – with each action class, incorporating nouns as a strong connection between seen and unseen categories. The third graph incorporates visual representations derived from a small number of samples of the unseen categories – this graph is thus applicable to Few-Shot Learning, rather than ZSL. The reason for the incorporation of visual representations is that categories which are similar in the semantic space may nevertheless have distinct visual appearances – authors give the example of "pommel horse" and "horse walking", which have similar word embedding representations but dissimilar visual appearances.

Kato et al. (2018) construct a graph consisting of three types of nodes: noun nodes, object nodes, and action nodes. Action nodes are linked to the verbs and nouns which the associated action involves.

Values of graph nodes are generally initialized with values derived from semantic vector spaces – this is important as it establishes the initial relations which the GCN iterates over. This iteration incorporates relations defined by the edges in the graph, and allows information to transfer from node-to-node along edges. E.g., in Kato et al. (2018) action nodes, which are initially set to zero vectors, acquire a representation determined by the vectors of neighboring noun and verb nodes, which have been initialized with semantic vectors.

Like previously detailed methods, these methods learn a mapping from visual features (taken from a CNN pretrained on a separate task) into a shared embedding space – though here that space is shared with representations produced by the GCN. In Kato et al. (2018) that mapping is produced through two layers of neurons with sigmoid activations, resulting in a nonlinear mapping from the visual features into the shared semantic space. Similar to previous work, these layers are trained through applying a loss measuring the similarity between the visual features after embedding, and the vectors associated with the labels of the input as produced by the GCN.

The mapping from pretrained visual features onto the GCN produced action vectors can then be trained over the seen training set. To obtain action predictions when applying the network to novel actions during testing, nearest neighbor can be performed between the visual features and the action vectors.

11.5 Future directions

Here we discuss implications and future work implied by the action based framework outlined in the rest of this chapter. Action has implications for *tasks and datasets* – it enables conceptual modeling conductive to generalization and longer term temporal prediction. Action benefits from modeling of *concepts* beyond those from conventional CV. As CV progresses the scalability of fully supervised methods becomes an increasing issue, and action based methods help mitigate this issue while benefiting from *semi- and unsupervised* paradigms. Finally action helps enable an integration of *cognition* and symbolic modeling into *perception*, including across multiple perceptual modalities.

Tasks and datasets: Activities span long time spans. Thus when seeking solutions to visual activity understanding, we face problems much more challenging than those we encounter in current action recognition tasks. Objects, actions, affordances, and other scene constituents relate to each other semantically over multiple time scales, and we need to find ways to

model these relations. We think that this capability is not well demonstrated with the task of recognizing actions. Instead, we should pick tasks that demonstrate generalization and a conceptual understanding of action (as opposed to a purely appearance based understanding). Such tasks include zero-shot learning and the prediction of future actions, and translation from one viewpoint to another (e.g., from first person to third person). Today's CV research is largely driven by the collection of new datasets and definition of new challenges – existing datasets do not sufficiently cover long term and conceptual modeling of activities. The datasets ideally should have recordings from multiple views, because this opens possibilities for interesting research, for example, to solve the challenge of transferring knowledge between the first person and the third person view. Lastly, most datasets feature indoor scenes. It will be interesting to collect outdoor scenes and analyze them, as discussed above, by looking at the relations between affordances, interactions and long-term relations. We could also attempt an action-based analysis on data from the autonomous driving domain.

Concepts for long-term activity understanding: Activity understanding requires action-based concepts at multiple time scales. We discussed in this chapter such quantities at the single image and short-term time scales, including affordances, hand grasps, geometric relations, and we emphasized the use of geometric reconstruction processes because of their robustness (see discussion in Section 11.3.4). The next step will be to add further constraints and include robust constraints for modeling temporal-relations at longer scales. We could make use of ontologies in categorizing objects and actions. On verbs we can impose classifications based on action effects (Yang et al., 2013), ergonomic principles, or force and location related constraints. Longer-term relations include causal relations, and constraints on possible and impossible action sequences. We can also model physics constraints, and use physics engines – but in order to integrate these into deep architectures we need to include these constraints into vector spaces that relate perception to cognition (Section 11.4).

Reducing supervision: Early approaches on integrating language with vision (see Sections 11.4.1 and 11.4.4) have relied heavily on supervision. For example, visual attribute recognition or object recognition has been implemented via supervised learning. Graph-based models using shared embeddings for ZSL need to incorporate in advance the categories of the "unseen" set which they may encounter during test time. Naturally, evolving approaches will find their way into the action-based framework, including unsupervised and self-supervised learning, transfer learning, metalearning, and eventually never-ending learning (Mitchell et al., 2015). For example, in constructing visual ontologies, we don't want to rely fully on supervised visual learning of metaverb classes. One solution to such modeling is dictionary learning (Zheng et al., 2016), but so far these approaches have been limited to simple actions. We will need methods that scale to more complex human manipulation actions. Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) have been shown successful in modeling fine-grained action images and videos. We could, for example, use VAEs to learn the underlying distribution of data into a discretized latent space that encodes the metacategories.

Involving the whole brain: Humans' understanding of the world is grounded in our motoric cognition and all our senses. Similarly, our models should include other sensory modalities like auditory, tactile, or proprioception signals in addition to modeling of vision and cognition representations. Different modalities provide complementary information, and because of this allow for different ways of organizing concepts. In addition to questions of

learning with different modalities, we also need methods for accessing stored concepts when given perception from any modality. Studying the integration of diverse modalities will also benefit from studying memory. A framework known as Vector Symbolic Architectures (VSA) (Plate, 1995; Eliasmith, 2013), which includes Hyperdimensional (HD) Computing (methods making use of very high dimensional vector spaces) (Pentti, 2009), has been proposed as a theoretical model for artificial intelligence. HD Computing combines advantages of neural-based AI approaches with systematic compositionality and rule-like behavior from classical symbolic AI (Levy and Gayler, 2008). In this framework concepts are encoded into vector spaces, and algebraic operations are defined on these vector spaces. These operations include the addition of related concepts and the binding of vectors of different origin – for example this could be sound and vision, or vision and motor (Mitrokhin et al., 2019). These operations maintain the separability of one modality from the other. We may build on this framework and integrate it with neural network approaches. The goal will be to retain an explicit memory encoding different perception modalities, while maintaining the capability of recalling information from any modality.

11.6 Conclusions

The purpose of Computer Vision is to produce interpretations which are of use to humans. Action is central to our understanding of the world, yet is underutilized in contemporary CV. This chapter covered scene and activity understanding that has the concept of action and interaction at its center. We covered action based approaches to scene understanding involving modeling at multiple temporal scales, starting from object interpretation in terms of affordances at the instantaneous level, up to basic actions, then up to full activities at the longer temporal scale. We described the well developed area of affordance learning, and described works on activity understanding combining cognitive and linguistic approaches with humanly interpretable modules essential in characterizing activities and segmenting video temporally. We discussed methods for the integration of visual representations with knowledge, both engineered and derived from text corpora. We covered the integration of vision, centered on action, with graphical constraints, and discussed future directions including creating new challenges and datasets, adding concepts for encoding long-term relations, adapting semi- and unsupervised learning approaches, and incorporating memory as a central component to the action-based framework.

Acknowledgment

The support of the National Science Foundation under grants BCS 1824198 and OISE 2020624 is gratefully acknowledged.

References

Aditya, Somak, Yang, Yezhou, Baral, Chitta, Aloimonos, Yiannis, Fermüller, Cornelia, 2018. Image understanding using vision and reasoning through scene description graph. Computer Vision and Image Understanding 173, 33–45.

Aloimonos, Yiannis, Fermüller, Cornelia, 2015. The cognitive dialogue: a new model for vision implementing common sense reasoning. Image and Vision Computing 34, 42–44.

Ansuini, Caterina, Cavallo, Andrea, Bertone, Cesare, Becchio, Cristina, 2015. Intentions in the brain: the unveiling of mister Hyde. The Neuroscientist 21 (2), 126–135.

Bajcsy, Ruzena, 1988. Active perception. Proceedings of the IEEE 76 (8), 966–1005.

Barsalou, Lawrence W., 2008. Grounded cognition. Annual Review of Psychology 59, 617–645.

Bo, Liefeng, Ren, Xiaofeng, Fox, Dieter, 2013. Unsupervised feature learning for rgb-d based object recognition. In: Experimental Robotics. Springer, pp. 387–402.

Cutkosky, Mark R., 1989. On grasp choice, grasp models, and the design of hands for manufacturing tasks. IEEE Transactions on Robotics and Automation 5 (3), 269–279.

Damen, Dima, Doughty, Hazel, Maria Farinella, Giovanni, Fidler, Sanja, Furnari, Antonino, Kazakos, Evangelos, Moltisanti, Davide, Munro, Jonathan, Perrett, Toby, Price, Will, et al., 2018. Scaling egocentric vision: the EPIC-kitchens dataset. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 720–736.

Das, Pradipto, Xu, Chenliang, Doell, Richard F., Corso, Jason J., 2013. A thousand frames in just a few words: lingual description of videos through latent topics and sparse object stitching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2634–2641.

Dessalene, Eadom, Devaraj, Chinmaya, Maynord, Michael, Fermüller, Cornelia, Aloimonos, Yiannis, 2021. Forecasting action through contact representations from first person video. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Dutta, V., Zielinska, T., 2017. Action prediction based on physically grounded object affordances in human-object interactions. In: Proceedings of the 11th International Workshop on Robot Motion and Control.

Ecins, Aleksandrs, Fermüller, Cornelia, Aloimonos, Yiannis, 2016. Cluttered scene segmentation using the symmetry constraint. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2271–2278.

Eliasmith, Chris, 2013. How to Build a Brain: A Neural Architecture for Biological Cognition. Oxford University Press.

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2010. The Pascal visual object classes (VOC) challenge. International Journal of Computer Vision 88 (2), 303–338.

Fermüller, Cornelia, Aloimonos, Yiannis, 1995. Vision and action. Image and Vision Computing 13 (10), 725–744.

Fermüller, Cornelia, Wang, Fang, Yang, Yezhou, Zampogiannis, Konstantinos, Zhang, Yi, Barranco, Francisco, Pfeiffer, Michael, 2018. Prediction of manipulation actions. International Journal of Computer Vision 126 (2), 358–374.

Fitzpatrick, Paul, Metta, Giorgio, Natale, Lorenzo, Rao, Sajit, Sandini, Giulio, 2003. Learning about objects through action-initial steps towards artificial cognition. In: IEEE International Conference on Robotics and Automation, vol. 3, pp. 3140–3145.

Frome, Andrea, Corrado, Greg, Shlens, Jonathon, Bengio, Samy, Dean, Jeffrey, Ranzato, Marc'Aurelio, Devise, Tomas Mikolov, 2013. A deep visual-semantic embedding model. Advances in Neural Information Processing Systems 26.

Ghosh, Pallabi, Saini, Nirat, Davis, Larry S., Shrivastava, Abhinav, 2020a. All about knowledge graphs for actions. arXiv preprint. arXiv:2008.12432.

Ghosh, Pallabi, Yao, Yi, Davis, Larry, Divakaran, Ajay, 2020b. Stacked spatio-temporal graph convolutional networks for action segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 576–585.

Gibson, James J., 1977. The theory of affordances. In: Bransford, John, Shaw, Robert E. (Eds.), Perceiving, Acting, and Knowing: Toward an Ecological Psychology. Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 67–82.

Grabner, Helmut, Gall, Jürgen, Van Gool, Luc, 2011. What makes a chair a chair? In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1529–1536.

Guha, Anupam, Yang, Yezhou, Fermüller, Cornelia, Aloimonos, Yiannis, 2013. Minimalist plans for interpreting manipulation actions. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5908–5914.

Gupta, Abhinav, Satkin, Scott, Efros, Alexei A., Hebert, Martial, 2011. From 3d scene geometry to human workspace. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1961–1968.

Hassanin, Mohammed, Khan, Salman, Tahtali, Murat, 2018. Visual affordance and function understanding: a survey. arXiv preprint. arXiv:1807.06775.

Hedau, Varsha, Hoiem, Derek, Forsyth, David, 2009. Recovering the spatial layout of cluttered rooms. In: IEEE International Conference on Computer Vision, pp. 1849–1856.

Hermans, Tucker, Rehg, James M., Bobick, Aaron, 2011. Affordance prediction via learned object attributes. In: IEEE International Conference on Robotics and Automation (ICRA): Workshop on Semantic Perception, Mapping, and Exploration.

Pentti, Kanerva, 2009. Hyperdimensional computing: an introduction to computing in distributed representation with high-dimensional random vectors. Cognitive Computation 1, 139–159.

Kato, Keizo, Li, Yin, Gupta, Abhinav, 2018. Compositional learning for human object interaction. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 234–251.

Kjellström, Hedvig, Romero, Javier, Kragić, Danica, 2011. Visual object-action recognition: inferring object affordances from human demonstration. Computer Vision and Image Understanding 115 (1), 81–90.

Kodirov, Elyor, Xiang, Tao, Gong, Shaogang, 2017. Semantic autoencoder for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3174–3183.

Koppula, Hema S., Saxena, Ashutosh, 2014. Physically grounded spatio-temporal object affordances. In: European Conference on Computer Vision. Springer, pp. 831–847.

Koppula, Hema S., Saxena, Ashutosh, 2015. Anticipating human activities using object affordances for reactive robotic response. IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (1), 14–29.

Koppula, Hema Swetha, Gupta, Rudhir, Saxena, Ashutosh, 2013. Learning human activities and object affordances from rgb-d videos. The International Journal of Robotics Research 32 (8), 951–970.

Lee, David C., Gupta, Abhinav, Hebert, Martial, Kanade, Takeo, 2010. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In: Advances in Neural Information Processing Systems, pp. 1288–1296.

Levy, S.D., Gayler, R., 2008. Vector symbolic architectures: a new building material for artificial general intelligence. In: Proceedings of the First Conference on Artificial General Intelligence (AGI-08). IOS Press.

Liu, Fayao, Shen, Chunhua, Lin, Guosheng, 2015. Deep convolutional neural fields for depth estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5162–5170.

Mandal, Devraj, Narayan, Sanath, Kumar Dwivedi, Sai, Gupta, Vikram, Ahmed, Shuaib, Shahbaz Khan, Fahad, Shao, Ling, 2019. Out-of-distribution detection for generalized zero-shot action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9985–9993.

Martin, A., 2007. The representation of object concepts in the brain. Annual Review of Psychology 58, 25–45.

Metta, Giorgio, Sandini, Giulio, Vernon, David, Natale, Lorenzo, Nori, Francesco, 2008. The iCub humanoid robot: an open platform for research in embodied cognition. In: Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems, pp. 50–56.

Mikolov, Tomas, Chen, Kai, Corrado, Greg, Dean, Jeffrey, 2013. Efficient estimation of word representations in vector space. arXiv preprint. arXiv:1301.3781.

Miller, George A., Beckwith, Richard, Fellbaum, Christiane, Gross, Derek, Miller, Katherine J., 1990. Introduction to wordnet: an on-line lexical database. International Journal of Lexicography 3 (4), 235–244.

Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohamed, T., Nakashole, N., Platanios, E., Ritter, A., Samadi, M., Settles, B., Wang, R., Wijaya, D., Gupta, A., Chen, X., Saparov, A., Greaves, M., Welling, J., 2015. Never-ending learning. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15).

Mitrokhin, A., Sutor, P., Fermüller, C., Aloimonos, Y., 2019. Learning sensorimotor control with neuromorphic sensors: toward hyperdimensional active perception. Science Robotics 4 (30), eaaw6736.

Montesano, Luis, Lopes, Manuel, Bernardino, Alexandre, Santos-Victor, José, 2008. Learning object affordances: from sensory–motor coordination to imitation. IEEE Transactions on Robotics 24 (1), 15–26.

Myers, Austin, Teo, Ching L., Fermüller, Cornelia, Aloimonos, Yiannis, 2015. Affordance detection of tool parts from geometric features. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 1374–1381.

Nguyen, Anh, Kanoulas, Dimitrios, Caldwell, Darwin G., Tsagarakis, Nikos G., 2017. Object-based affordances detection with convolutional neural networks and dense conditional random fields. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5908–5915.

Pagliardini, Matteo, Gupta, Prakhar, Jaggi, Martin, 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. arXiv preprint. arXiv:1703.02507.

Parikh, Devi, Grauman, Kristen, 2011. Relative attributes. In: International Conference on Computer Vision, pp. 503–510.

Pastra, Katerina, Aloimonos, Yiannis, 2012. The minimalist grammar of action. Philosophical Transactions of the Royal Society B: Biological Sciences 367 (1585), 103–117.

Plate, Tony A., 1995. Holographic reduced representations. IEEE Transactions on Neural Networks 6 (3), 623–641.
Qi, Siyuan, Huang, Siyuan, Wei, Ping, Zhu, Song-Chun, 2017. Predicting human activities using stochastic grammar. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1164–1172.
Qi, Siyuan, Wang, Wenguan, Jia, Baoxiong, Shen, Jianbing, Zhu, Song-Chun, 2018. Learning human-object interactions by graph parsing neural networks. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 401–417.
Roy, Anirban, Todorovic, Sinisa, 2016. A multi-scale cnn for affordance segmentation in rgb images. In: European Conference on Computer Vision. Springer, pp. 186–201.
Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, et al., 2015. Imagenet large scale visual recognition challenge. International Journal of Computer Vision 115 (3), 211–252.
Schuler, Karin Kipper, 2005. VerbNet: a broad-coverage, comprehensive verb lexicon. PhD thesis. Computer and Information Science Department, University of Pennsylvania, Philadelphia, PA.
Sigurdsson, Gunnar A., Russakovsky, Olga, Gupta, Abhinav, 2017. What actions are needed for understanding human actions in videos? In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2137–2146.
Silberman, Nathan, Hoiem, Derek, Kohli, Pushmeet, Fergus, Rob, 2012. Indoor segmentation and support inference from rgbd images. In: European Conference on Computer Vision. Springer, pp. 746–760.
Simonyan, Karen, Zisserman, Andrew, 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint. arXiv:1409.1556.
Srikantha, Abhilash, Gall, Jürgen, 2016. Weakly supervised learning of affordances. arXiv preprint. arXiv:1605.02964.
Stark, Louise, Bowyer, Kevin, 1991. Achieving generalized object recognition through reasoning about association of function to structure. IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (10), 1097–1104.
Summers-Stay, Douglas, Teo, Ching L., Yang, Yezhou, Fermüller, Cornelia, Aloimonos, Yiannis, 2012. Using a minimal action grammar for activity understanding in the real world. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4104–4111.
Teo, Ching Lik, Fermüller, Cornelia, Aloimonos, Yiannis, 2015. Detection and segmentation of 2d curved reflection symmetric structures. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1644–1652.
Tran, Son D., Davis, Larry S., 2008. Event modeling and recognition using Markov logic networks. In: European Conference on Computer Vision. Springer, pp. 610–623.
Ugur, Emre, Erhan, Oztop, Erol, Sahin, 2011. Goal emulation and planning in perceptual space using learned affordances. Robotics and Autonomous Systems 59 (7–8), 580–595.
Ugur, Emre, Piater, Justus, 2016. Emergent structuring of interdependent affordance learning tasks using intrinsic motivation and empirical feature selection. IEEE Transactions on Cognitive and Developmental Systems 9 (4), 328–340.
Varela, Francisco J., Rosch, Eleanor, Thompson, Evan, 1993. The Embodied Mind: Cognitive Science and Human Experience. MIT Press.
Wörgötter, Florentin, Erdal Aksoy, Eren, Krüger, Norbert, Piater, Justus, Ude, Ales, Tamosiunaite, Minija, 2013. A simple ontology of manipulation actions based on hand-object relations. IEEE Transactions on Autonomous Mental Development 5 (2), 117–134.
Wörgötter, Florentin, Ziaeetabar, F., Pfeiffer, S., Kaya, O., Kulvicius, T., Tamosiunaite, M., 2020. Humans predict action using grammar-like structures. Scientific Reports 10 (1), 1–11.
Xian, Yongqin, Lorenz, Tobias, Schiele, Bernt, Akata, Zeynep, 2018. Feature generating networks for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5542–5551.
Xiao, Jianxiong, Hays, James, Ehinger, Krista A., Oliva, A., Torralba, A., 2010. Sun database: large-scale scene recognition from abbey to zoo. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 3485–3492.
Yan, Sijie, Xiong, Yuanjun, Lin, Dahua, 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32.
Yang, Yezhou, Fermüller, Cornelia, Aloimonos, Yiannis, 2013. Detection of manipulation action consequences (MAC). In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2563–2570.
Yang, Yezhou, Guha, Anupam, Fermüller, Cornelia, Aloimonos, Yiannis, 2014. A cognitive system for understanding human manipulation actions. Advances in Cognitive Systems 3, 67–86.
Yang, Yezhou, Fermüller, Cornelia, Li, Yi, Aloimonos, Yiannis, 2015a. Grasp type revisited: a modern perspective on a classical feature for vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 400–408.

Yang, Yezhou, Li, Yi, Fermüller, Cornelia, Aloimonos, Yiannis, 2015b. Robot learning manipulation action plans by "watching" unconstrained videos from the world wide web. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29.

Ye, Chengxi, Yang, Yezhou, Mao, Ren, Fermüller, Cornelia, Aloimonos, Yiannis, 2017. What can I do around here? Deep functional scene understanding for cognitive robots. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 4604–4611.

Yu, Xiaodong, Fermüller, Cornelia, Teo, Ching Lik, Yang, Yezhou, Aloimonos, Yiannis, 2011. Active scene recognition with vision and language. In: 2011 International Conference on Computer Vision, pp. 810–817.

Zampogiannis, Konstantinos, Fermüller, Cornelia, Cilantro, Yiannis Aloimonos, 2018. A lean, versatile, and efficient library for point cloud data processing. In: Proceedings of the 26th ACM International Conference on Multimedia, pp. 1364–1367.

Zampogiannis, Konstantinos, Fermüller, Cornelia, Aloimonos, Yiannis, 2019. Topology-aware non-rigid point cloud registration. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Zampogiannis, Konstantinos, Yang, Yezhou, Fermüller, Cornelia, Aloimonos, Yiannis, 2015. Learning the spatial semantics of manipulation actions through preposition grounding. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1389–1396.

Zhang, Li, Xiang, Tao, Gong, Shaogang, 2017. Learning a deep embedding model for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2021–2030.

Zheng, J., Jiang, Z., Chellappa, R., 2016. Cross-view action recognition via transferable dictionary learning. IEEE Transactions on Image Processing 25 (6), 2542–2556.

Biographies

Cornelia Fermüller is a Research Scientist at the University of Maryland Institute for Advanced Computer Studies. She holds a Ph.D. from the Vienna University of Technology, Austria and an M.S. from the Graz University of Technology, both in Applied Mathematics. Her research interest has been to understand principles of active vision systems and develop biological-inspired methods, especially in the area of motion. Her recent work has focused on human action interpretation and the development of 3D motion algorithms for extreme conditions using event-based sensors.

Michael Maynord is a PhD candidate in the department of Computer Science at the University of Maryland College Park, advised by Yiannis Aloimonos and Cornelia Fermüller. His background encompasses symbolic Artificial Intelligence, including cognitive architectures, Computer Vision, including action understanding, and methods integrating AI and CV.