Fine-grained Action Plausibility Rating

Timo Lüddecke and Florentin Wörgötter

University of Goettingen
III. Institute of Physics
Friedrich-Hund-Platz 1, 37073 Göttingen
Bernstein Center for Computational Neuroscience Göttingen

Corresponding author: timo.lueddecke@uni-goettingen.de

DOI: https://doi.org/10.1016/j.robot.2020.103511

Abstract

An essential capability of humans is the effortless identification of useful tasks based on visual cues in everyday situations. Objects and their surroundings are integrated and processed to differentiate plausible from implausible actions. In this work, we study how to teach this ability to robots. In contrast to many tasks in computer vision where the goal is an accurate description (object labels, caption, scene class) of the present situation here the challenge is to make reasonable guesses about which forms of plausible and implausible actions can be conducted. To this end, we collect a dataset that associates images with probabilities over a set of actions. A convolutional neural network is trained to match these ground truth plausibility scores using this dataset. We compare the performance of state-of-the-art encoder architectures and specifically analyze the role of contextual cues quantitatively. While the object recognition capabilities of the encoder have a strong impact on performance, using context did not lead to substantial improvements. We show qualitatively the utility of such a system for robotic action selection in a household setting.

1 Introduction

In a given situation humans often have plenty of action possibilities, but commonly only a tiny fraction is appropriate. Making such action decisions in everyday life feels effortless and often happens subconsciously. The sense of appropriateness that guides the decision is probably not innate but learned, while growing up. Robotic systems, however, naturally lack this skill and therefore, can exhibit a behavior that is surprising and unexpected for humans due to the robot's misinterpretation of a situation. Thus, transferring this particular aspect of common-sense knowledge to machines would have a great impact on their usability, in particular for situations where interaction with humans is required.

The problem of representing common-sense knowledge itself is not new and has been addressed for decades. Most of these past approaches to represent common-sense facts are sym-

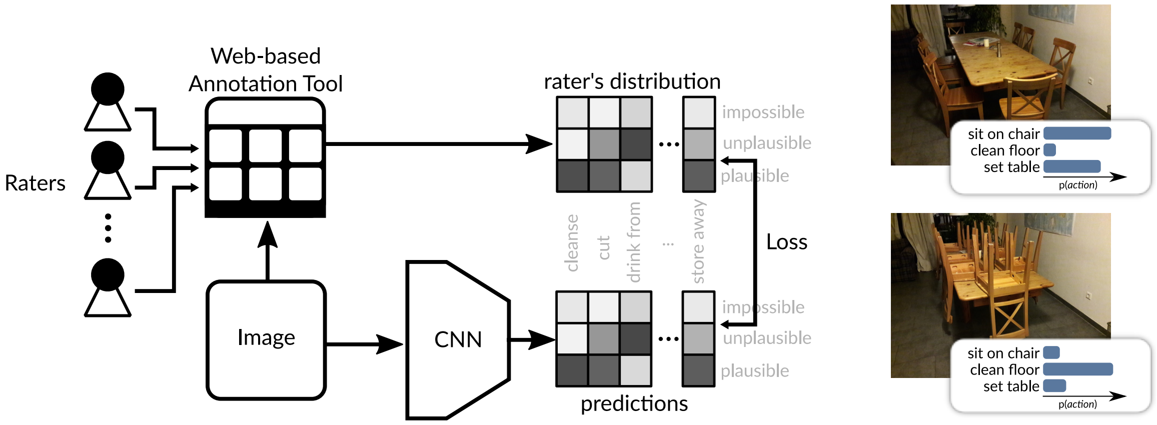


Figure 1: The idea presented in this paper: Based on human ratings, a CNN is trained to learn the plausibility of actions in different contexts. The sketch on the right side highlights that small changes in the image can have strong impact on action plausibility.

bolic, i.e. they assume that knowledge can be expressed in terms of a finite set of discrete symbols and their relations. While this comes with the advantage of interpretability, it is unlikely that all knowledge can be easily expressed in this form, and this is especially problematic for inherently continuous facts such as *the likelihood of eating from a dish decreases with its level of dirtiness*. Examples of this kind of knowledge representation can be found in the psychological literature where action possibilities (affordances) are modeled as probabilistic functions instead of binary attributes [12].

Symbolic approaches fail if they only take class labels into account as contextual and if appearance details are crucial for a task. *Simple* symbolic mappings (like object labels mapped to actions) are doomed, because common object classes are very broad (high within-class variance). For example, the class ″cup″ consists of full and empty cups as well as cups with and without handle. However, actions often depend on the state of the object, which cannot be inferred from such simple labels. Often, the state can be much more informative concerning an action than the object label, since an action can be compatible with a broad range of object classes in a certain state. Of course, with quite some effort these aspects can be incorporated into symbolic systems, too. However, for this all relevant details need to be known and specified a-priori. If the fill-level of a cup is critical for drinkability, a logical variable “fill-level” needs to be added to the system along with an image recognition component that can detect it. The same is true for each and every such object and situation-dependent aspect leading to a massive effort in pre-defining all of this in the right way. Different from that, the advantage of our approach is that there is no need to explicitly model the space of all variables influencing an action as the system learns these relationships end-to-end.

In this article we address the novel problem of rating how plausible certain actions are. An illustration of this idea is shown in Figure 1. For this purpose, we develop a hybrid system that represents common-sense knowledge in a distributed, implicit way but also relies on a hard coded action compatibility table that defines if actions can in-principle be conducted on different object classes. We take images from the OpenImages dataset [23] and then ask humans to rate how plausible they consider certain actions. Having obtained a such-labeled dataset, we use it to train a neural network to predict action plausibilities. This way rules like *if dirty dishes are close to each other stack them* or *if room is empty use remote control to turn off TV* can be learned from the data. Importantly, after training, the system is able to directly map from pixels to action plausibility probabilities, evading a symbolic scene representation.

Contributions The main contribution of this work is a dataset of action plausibilities (PlausiblAct) and an associated training procedure based on the mapping from ratings to action plausibilities. Together with a suitable loss function, this allows using conventional CNNs to predict action plausibilities from images. We are confident that a CNN trained with the proposed method can be an important component in robotic systems toward endowing them with more situation-dependent autonomy. This specifically concerns the domain of service robotics where robots need to identify appropriate maintenance tasks (such as cleaning up) autonomously. Here a robot could use our method to infer appropriate actions from visual input only without receiving explicit instructions from the human. Similarly, context dependent action selection is also important for machines that need to collaborate with humans and thus have to anticipate human actions. Here the predictions of our method provide guesses of what the human's next action might be. The robot can consider that when conducting own actions, e.g. in order to avoid cooperation conflicts. Our system is trained to operate on real-world data, which is more challenging than set-up lab scenes. As a consequence, the method is not only relevant in robotics but also for the computer vision community where natural images are ubiquitous.

In the following we will first discuss the state of the art, followed by an introduction to the new data set and the here-used algorithms. Results and conclusions concerning this approach including its limitations will end the paper.

2 Related Work

We are not aware of any approach that explicitly deals with the problem of rating actions with respect to their plausibility from observed scenes. However, several related tasks have been addressed before. Particularly, we discuss anticipation tasks, which try to identify what will happen next. These share the goal of predicting future actions with our approach but does not differentiate actions based on their *individual* likelihood. The latter represents the focus of our work. Subsequently, we present an overview, organized by the input data the respective methods use.

Video-based Methods A large body of work in anticipation operates on videos, which seems natural since movies involve a temporal dimension to base predictions on. In the work of Lan et al. [25], the next action in a TV show is predicted based on previous frames and object bounding boxes. For this a hierarchical video representation called *movemes* is proposed. The anticipation of human activities that is addressed in Koppula and Saxena [21] can be considered a closely related task. They model human pose, object affordances, object locations, and sub-activities in a graph that changes over time through a temporal conditional random field. By sampling from this model, prospective activities can be predicted. These possible futures could also involve actions we are interested in. While their dataset only comprises 120 scenes, we prefer a larger number of scenes to allow for more detail within scenes. Vondrick et al. [43] model the development of visual feature representations (obtained from a CNN) over time in a self-supervised setting. Some video recognition approaches have been evaluated in an early recognition setting [49, 51]. Given only a certain fraction (e.g. 20%) of the first frames of an action, the goal is to determine the action, which can also be seen as a weak form of anticipation.

Our task differs from the tasks addressed in these papers in using only a single RGB image as input. This implies that models cannot rely on patterns that occur in sequences of actions but have to identify cues in the provided static image. The set of actions we consider is vastly different from other methods' sets of actions. Therefore, evaluating on their data is not possible. Furthermore, some approaches do not frame the problem as a classification task but use a different output space (e.g. the trajectory evaluation in [21]).

Still Image-based Methods However, anticipations can be made from static images, too. For example, Walker et al. [45] predict pixel-wise trajectories. For each pixel a prediction of how it will evolve in the future is conducted using an autoencoder. A similar idea is pursued by Chao et al. [4]. Instead of dense pixel trajectories, they specialize entirely on anticipating pose dynamics. Similar to us, Vu et al. [44] predict distributions over plausible actions from images for which they collected the SUN Action dataset. While they predict general actions for whole scenes, we focus on more specific actions considering only individual objects. Fouhey and Zitnick [11] follow a single image setting, too, but they use abstract scene representations to learn what might happen next. Instead of predicting specific actions they consider the dynamics of objects.

In the work of Qi et al. [33], interactions between humans and objects are studied in images as well as in videos. Scenes are parsed into a graph that indicates relations between objects. In one experiment, this graph is used to anticipate future activities on the CAD-120 dataset [22]. Similarly, to the video-based methods a direct comparison with these methods is not feasible due to different output labels (e.g. [44] uses a large set of generic actions such as "talk" or "drive"). However, we conduct a comparison with [44] by training our CNN on their data in Section 5.3.

Psychology and the Concept of Affordances Action plausibility scoring is related to the concept of affordances coined by Gibson [13] and later refined by Gibson [14, Chapter 8]. While affordances indicate which interactions with the environment are possible for an agent, they do not come with any notion of preference. This means, No differentiation about what action is more likely to happen takes place. Hence, affordances can be considered to be less-abstract than the plausibilities we propose in this paper. Affordances have been studied in various forms: for whole images [50], as poses [15], bounding boxes [9, 48], densely for every pixel [28, 29, 31, 34, 37] or from video [22, 46]. However, existing research is not limited to discovering action possibilities: Mechanisms that drive the selection of actions have been investigated in neuroscience [1] including the creation of computational models [6, 39].

Note that the concept of affordances centers strongly on objects, essentially asking: which actions are suggested by different objects? Agents, humans or robots, however many times are rather plan-driven and they ask this question the other way round: which object can I use for a planned action? To better accommodate both types of queries, recently the concept of Object-Action Complexes (OACs) had been introduced [24, 47] that assumes that objects and (planned) actions are inseparably intertwined. Our current study takes this one step further stating that objects *with certain properties* and actions are intertwined. For example, *full* cups are for drinking, *dirty* cups for cleaning, etc.

Several works from experimental psychology address the selection of actions. Riddoch et al. [35] focus on the relation between object pairs in patients with parietal lesions. They found subjects to perform better in identifying objects when the objects were arranged correctly for an action to be directly performed compared to other relative positions of the objects. A similar effect was found by Roberts and Humphreys [36] in healthy participants. Handy et al. [16] showed that graspable objects draw attention if they are perceived in the lower right visual field, although they are irrelevant for the given task. A comprehensive and recent review on the complexity of human action selection and its relation to (context-dependent) affordances can be found in [3]. In general, aspects of scene context are not in the core of the older psychological literature but during the last decade Schubotz and coworkers have more thoroughly addressed this using fMRI [10].

While these works explore a related domain, experimental psychology generally seeks to answer how biological or cognitive mechanisms work. This involves understanding human cognition [27, 36] or representations of action in the brain [7] rather than enhancing robotic systems. Objective measures such as reaction time are tracked to allow for a statistical analysis instead of subjective judgments of participants. Furthermore, psychology often tries to identify linear relationships that allow for interpretation. In contrast to work in this field, we gather the content of the responses (rather than their timings and accuracies) for training complex non-linear models.

3 The PlausiblAct Dataset

In this section, we introduce the PlausiblAct dataset, which associates images with a probability distribution over a set of ten actions. We explain the design of the dataset from the selection of actions via collecting data to generating probability distributions from the gathered annotations. The images of PlausiblAct come from the OpenImages dataset [23], which contains scenes (images) showing multiple objects with corresponding bounding boxes. For our dataset, we extract individual objects and denote them as instances.

3.1 Choice of Actions and Ratings

In contrast to labeling object names, it is more challenging to assign actions. Actions are to some degree subjective, depend on a state (e.g. hungry, tired) or on past actions. Therefore, a key challenge in this work is to constrain the setting in such a way that actions become less subjective. To this end, we focus on actions that tend to be *unconditional*. This involves actions whose utility immediately pops up when a scene is perceived, without depending on the state of the observer. We say "tend to" because even under these considerations the here-chosen actions remain *somewhat* conditioned on the state but to a smaller extent than many others. Specifically, actions, which are either plan-driven (e.g. to hammer a nail to fix something) or mood-driven (e.g. watch TV, read a book) are excluded. In such cases we would not expect the actions to be reliably rate-able as raters might assume different states leading to inconsistent ratings.

We identify a set of ten actions $\mathcal{A}$ that match these principles. They are presented in Figure 2. In addition to actions, we need to define possible ratings for an action instance. In order to reduce the cognitive load for the raters, we follow a simple approach and use only three possible ratings $\mathcal{R} = \{\text{impossible}, \text{implausible}, \text{plausible}\}$. While impossible refers to the physical layout of a scene, plausibility decisions often depend on the context within a scene.

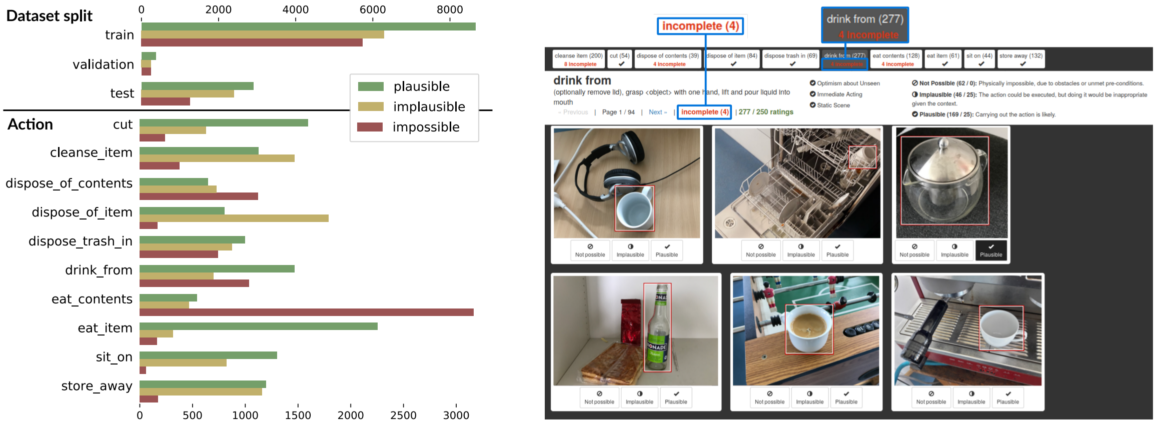
For each of these ten actions, we manually enumerate the complete subset of compatible object classes from all 600 object classes in OpenImages [23] (see appendix). Compatible means that, based on the object class name, it is hypothetically possible to conduct the action on an object of this class. E.g. a glass is hypothetically compatible with the action drinking, but not always, as it can be empty. Incompatible object-action pairs (as specified by the table in the appendix) are implicitly rated as impossible. For instance, let us assume there were only the actions *eat* and *sit on* and the object *cake*. Then defining the set of eat-able objects to be $\{cake\}$ implies that the cake is never sit-able.

3.2 Scene and Instance Selection

Having defined compatibility between actions and objects, the next step is to select *good* scenes from the set of remaining scenes. Note that people do not take photos randomly. They rather focus on beautiful and tasty things. E.g. food is most often photographed before and not during eating. This leads to image databases not really being representative illustrations of reality but collections of cherry-picked moments. However, to generate reasonable action plausibilities we need a good coverage of all situations. In the following, we introduce mechanisms that counteract these biases.

As a first step, scenes are excluded when one of these criteria is met:

- Small coverage (less than 2% of all pixels), as the crop would not be recognizable.



drink from

(optionally remove lid), grasp <object> with one hand, lift and pour liquid into mouth

Optimism about Unseen
Immediate Acting
Static Scene

Not Possible (62 / 0): Physically impossible, due to obstacles or unmet pre-conditions.

Implausible (46 / 25): The action could be executed, but doing it would be inappropriate given the context.

Plausible (169 / 25): Carrying out the action is likely.

Figure 2: Left: Frequencies of the ratings for each action. The top three rows refer to the subset, the other rows to the action. Right: Screen-shot of the annotation tool (in this screenshot, Open-Images images have been replaced by own images for copyright issues.). In blue we highlight the indicators that point to required samples. By clicking "incomplete" only required samples will be shown.

- Large coverage (more than 70% of all pixels), as there would be little room for context.
- Contains a human, as this would often require the rater (and later-on also the system) to infer intention, which we consider beyond the scope of this paper.

Furthermore, we maintain only one bounding box if two bounding boxes overlap with an intersection over union of over 0.5. Then we use the one for the less frequent class. Lastly, we manually remove scenes showing humans that were not considered by the labels and hence slipped through our previous filtering mechanism. Additionally, product photos and images having poor quality are manually removed. Lastly, we put an upper limit on the number of occurrences of each object class. To prefer larger objects we sort all instances descending by size and then select the first 1000 instances of each object class which increases the variety of the included object classes.

3.3 Collection of Annotations

Annotations are gathered using a web-based interface. After receiving instructions and being shown example ratings, raters could explore a large number of instances for each action. The order of instances is shuffled individually for each rater. Which instances are available for which action depends on the manually defined compatibility between object classes and actions, e.g. drinking from a chair is incompatible and therefore not presented. This way, we reduce the workload of the raters as they do not have to rate many impossible actions. Each rater sees the compatible instances in a different, randomized order and chooses freely which instances to rate and which to ignore (not rate).

Rater instructions All raters received explicit instructions. Pilot experiments suggest that these are critical for obtaining a reasonable inter-rater reliability as the annotation of actions can be highly ambiguous. Following our observations from the pilot experiments, we instructed raters to follow three principles. These are the original instructions presented to the raters:

- **Optimism about the Unseen**: If you are uncertain about some unseen aspects of the scene, please assume the most favorable situation for the given action.
- **Immediate Acting**: Consider the plausibility of conducting the action without delay. Do not assume that the action execution could wait.
- **Static Scene**: Do not assume changes to the scene that make the action possible that go beyond the definition of the action. Only consider the presented situation and pay attention to the action definition.

In addition, we showed eleven examples of how these principles are supposed to be interpreted to the raters.

Consider rating a scene involving an opaque bottle on a table regarding the action *drink*. The principles above mean that the action should be rated by assuming that the bottle contains drinkable liquid (optimism), the table layout cannot be changed (static scene) and we cannot conduct other actions before drinking (like filling the bottle first).

While we first experimented with a sequential design, where only one instance at a time is presented to the rater, we finally decided to employ a multi-scene paradigm. For a given action, multiple scenes are presented, and the user can freely select, which instances to annotate. This allows for faster and more reliable annotations as hard or unclear samples can be skipped. Furthermore, this paradigm allows us to ask the raters to provide a minimal number of ratings for the categories implausible and plausible, which results in a more balanced dataset. The web-based tool is shown in Figure 2 (right). We discuss inter-rater reliability in Section 5.2.4, after the explanation of the metrics used in this work. We use the split in training, validation and test data defined by OpenImages [23]. For the training data, we allow choosing annotations freely as described above. As a consequence, the training procedure has to deal with incompletely annotated instances. For creating the ground truth of the test data, we requested the raters to label instances completely (i.e. all compatible actions must be rated), which enables computing meaningful metrics on the test set. For this, indicators of missing instances were shown in the web-based interface to prompt the rater to complete a rating procedure.

Statistics In Figure 2 (left) and Figure 3 we present distributions of the user-provided ratings for all actions and selected objects. In total, eight raters provided 28,046 ratings on 18,837 instances. Impossible was chosen 7,219, implausible 8,922 and possible 11,905 times. The eight raters have three different nationalities and four of them are male and four are female. Five of them are PhD students from our department but unaware of the purpose of this work. The other three are recruited students from University of Göttingen and received a compensation of 28 Euros, each. The raters received an electronic introduction to the task and provided annotations for about three hours. They were allowed to take breaks when necessary by their own admission. According to section C.III.6 of the ethical guidelines of the German Psychological Society this experiment does not require explicit consent. The raters participated voluntarily.

3.4 From Annotations to Plausibilities

Having collected a set of annotations, we need to transform it to trainable data. Each instance may have received ratings for some actions from one or more raters.

The key idea is to train the network to match the plausibility distribution of the raters for each instance. Not every instance suggests clear actions and often multiple ratings seem plausible. By modeling the ground truth as a distribution over ratings we can incorporate a notion of uncertainty. This approach is different from image classification, where the ground truth distribution

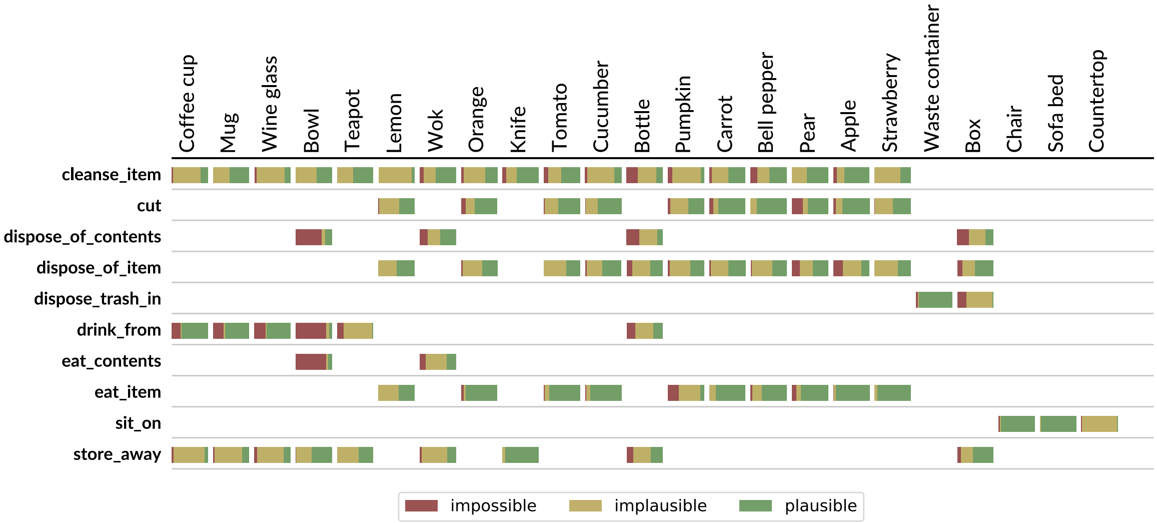


Figure 3: Rating distributions for all actions and frequent objects. Note, ratings are often not uniformly distributed for objects. Incompatible combinations of action and object are left blank.

accumulates all mass on a single label. In our case this happens only if all raters agree. Moreover, we predict ten actions per instance simultaneously.

Formally, for every instance $i \in \mathcal{I}$ (i.e. an object in a scene) we aggregate all associated ratings into a matrix $\mathbf{R}^{(i)} \in \mathbb{N}^{|\mathcal{A}|\times 3}$. Each element $\mathbf{R}^{(i)}_{a,r}$ denotes the count of ratings r for action a. In addition, a mask $\mathbf{v}^{(i)} \in \{0,1\}^{|\mathcal{A}|}$ is computed that indicates which rows (actions) of $\mathbf{R}^{(i)}$ are valid for an instance. This is necessary, because in the training set, annotations can be incomplete. Since raters can freely choose, which instances to annotate, there is no guarantee that for a given instance all possible actions are actually rated. The values of unrated yet compatible actions in $\mathbf{R}^{(i)}$ are not informative and therefore must be excluded from the computation of the loss. Thus, later, we will use $\mathbf{v}^{(i)}$ to exclude such undefined actions from being considered in the loss. Note, *compatibility* is manually defined by us based on the object names while *validity* depends on the annotations provided by the raters. A specific action of an instance can be *invalid* if it is *compatible* but did not receive any ratings. Next, the ground truth plausibility matrix $\mathbf{P}^{(i)}$ is generated from $\mathbf{R}^{(i)}$.

$$\mathbf{P}^{(i)}_a = \begin{cases} \dfrac{\mathbf{R}^{(i)}_a}{\sum_r \mathbf{R}^{(i)}_{a,r}} & a \text{ is compatible with instance } i \\ [1,0,0] & \text{otherwise} \end{cases} \tag{1}$$

Here the vector $[1,0,0]$ is used to assign the rating *impossible* to all incompatible actions (as described above).

4 Methods

4.1 Loss

Given an input image $\mathbf{I}$ of an instance, the network f predicts a matrix that assigns a probability to each rating for all actions. The rating probabilities for each action must sum to one. The loss is calculated by the cross entropy (see Section 4.5 for a definition of cross entropy) between each action's predicted rating distribution and the actual distribution obtained from the raters, denoted by $\mathbf{P}^{(i)}$.

$$\mathcal{L}^{(i)} = \frac{1}{\sum_{a \in \mathcal{A}} \mathbf{v}_a^{(i)}} \sum_{a \in \mathcal{A}} \mathrm{CE}(f(\mathbf{I}^{(i)})_a, \mathbf{P}_a^{(i)}) \mathbf{v}_a^{(i)}$$

In case an action is required but not provided, the value of $\mathbf{P}$ is invalid and should not contribute to the error expressed by the loss. This is realized by using the validity mask $\mathbf{v}^{(i)}$.

Data Augmentation Since we have to cope with limited training data, we apply different forms of data augmentation in order to increase the robustness of the classifier. This involves random cropping, adding Gaussian blur, changing gamma and colors of the image. For the sake of simplicity, we control the strength of these operations with a single integer value a. The optimal value for a is determined experimentally (see Table 2). Note, on the validation and test set images, no augmentation is used. For details on the implementation of augmentation we refer to the Appendix or the source code.

Implementation We employ batch normalization [19] and early stopping after 3 epochs without improvement of the validation loss. Weight updates are carried out with ADAM [20]. The code is implemented based on the PyTorch [32] framework.

4.2 Models

We use state-of-the-art convolutional neural networks architectures that have proven to work well for image recognition tasks. These include different variations of ResNet [17] and InceptionV4 [41]. Instead of training from scratch, we initialize the networks weights from pre-training on ImageNet [8] unless otherwise stated.

4.3 Baselines

We start our analysis by introducing two baselines:

- The **mode baseline** always predicts the most common rating for the depicted object. This is somewhat unfair since the baseline uses object labels other models do not have. However, it provides us with insights about how strongly the prediction of an action is tied to the underlying object class.

- The **no input baseline** is identical to a normal model but does not receive any image as input. Hence, the only way it can minimize loss is to learn the dataset distribution. This baseline provides us with a reference to relate other scores with. If a model does not perform better than this baseline, it has not learned anything but the biases present in the dataset.

4.4 Context Representations

As stated above, instances are objects within larger scenes. Hence, it might be useful to make the rest of the scene accessible to the model. For incorporation of this kind of context, we differentiate between multiple ways, which we describe in the following. Context representations that involve a "+" imply two inputs (instance image + some context image) to the model and thus require two separate encoder networks.

setting	1st input	2nd input
ignore	black image	-
img+mask	instance image	context image with instance being masked
img+full	instance image	context image (no masking)
only-masked	context image with instance being masked	-
only-full	context image (no masking)	-

Table 1: Input data for the different context settings

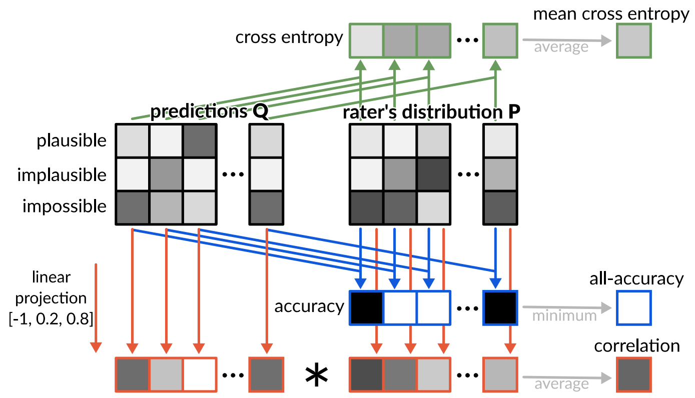


Figure 4: Illustration of how metrics are computed for one instance. Predictions of the network and ratings of the annotators are collected in two matrices $\mathbf{Q}$ and $\mathbf{P}$. By comparing the most likely ratings for each the accuracy is obtained.

- The trivial case *ignore* means ignoring the context entirely and considering only the instance's object.
- In the *img+masked* setting, we mask the object bounding box with a black rectangle. Hence, the network has no access to the object's visual features but has to rely only on contextual cues. Additionally, as a second input, the instance image is shown, too.
- In the *img+full* setting, the entire context is shown (without masking the instance) and the instance image is provided as a second input.
- In the *only-masked* setting, the entire context with the instance being masked is shown.
- In the *only-full* setting, the entire context is shown.

In Table 1 we provide an overview of the different input data types in the context settings.

4.5 Metrics

Obtaining quantitative scores for performance is a challenging task because the model's predictions and the ground truth are proper probability distributions. This is different from image classification, where the ground truth distribution has only one non-zero element. Furthermore, for each instance, all ten actions are predicted simultaneously.

For calculating performance metrics, we compare the ground truth $\mathbf{P}^{(i)}$ with $\mathbf{Q}^{(i)}_{a,r}$, which represents the network's predictions for action a and plausibility rating r of an instance i. The rating distribution sums to one, i.e. $\sum_{r\in\mathcal{R}} \mathbf{Q}^{(i)}_{a,r} = 1$.

All-action Accuracy (A-Acc) A straightforward choice to assess how well the predictions of a model are aligned with user annotated ratings is *accuracy*. If the highest mass rating is identical for prediction and ground truth, an instance is considered to be classified correctly. Note, the highest mass rating of the ground truth means the rating (impossible, implausible or plausible) that was most frequently assigned by the raters. We consider accuracy in two settings: 1) Independently for each action as described above and 2) for all actions of an instance. In the latter case, successful classification requires the correct prediction of all actions. While accuracy is easy to interpret, its disadvantage is its sole dependency on the maximum: The actual distribution over the ratings impossible, implausible and plausible is ignored. E.g. a confident prediction that puts all weight on plausible is treated equally to an uncertain, close-to-uniform prediction (where plausible happens to be the maximum by a small margin). This means, accuracy fails to represent the plausibility distribution as a whole.

$$\text{A-Acc}_a(\mathbf{P}^{(i)}, \mathbf{Q}^{(i)}) = \begin{cases} 1 & \text{if } \arg\max(\mathbf{P}^{(i)}_a) = \arg\max(\mathbf{Q}^{(i)}_a) \\ 0 & \text{otherwise} \end{cases}$$

Since we require all actions to be correctly annotated, we apply a $\min$ function on all action-wise accuracies. By averaging over all actions, we obtain the single A-Acc score:

$$\text{A-Acc} = \frac{1}{|\mathcal{I}|} \sum_{i\in\mathcal{I}} \min_{a\in\mathcal{A}} \text{A-Acc}_a(\mathbf{P}^{(i)}, \mathbf{Q}^{(i)})$$

Cross Entropy (CE) Since we need to compare probability distributions, we can make use of divergence measures, which express how similar probability distributions are. While many of such measures exist, a natural choice is to use cross entropy that is also used as the objective to train the network. We compute cross entropy for each action by:

$$\text{CE}_a(\mathbf{P}^{(i)}, \mathbf{Q}^{(i)}) = -\sum_{r\in\mathcal{R}} \mathbf{P}^{(i)}_{a,r} \log \mathbf{Q}^{(i)}_{a,r}$$

Then a single score is obtained by averaging individual cross entropies CE_a over all actions:

$$\text{CE} = \frac{1}{|\mathcal{I}|} \frac{1}{|\mathcal{A}|} \sum_{i\in\mathcal{I}} \sum_{a\in\mathcal{A}} \text{CE}_a(\mathbf{P}^{(i)}, \mathbf{Q}^{(i)})$$

A small cross entropy indicates high similarity between prediction and ground truth and is therefore desirable. In contrast to accuracy (A-Acc), CE is not intuitively interpretable (how good is a CE of say 0.2?) but it captures differences in the non-maximum parts of the distributions. Comparison with respect to CE is enabled by considering the CE of one setting relative to others, with low CE scores being desirable.

Correlation (Corr) The annotated data is ordinal, i.e. there exists an order from impossible over implausible to plausible. By defining a distance between the three ratings we can transform a plausibility distribution to a continuous, scalar value. This is done by a linear projection with a fixed vector $\mathbf{l} = [-1, 0.2, 0.8]$ that expresses the distances between the ordinal values. For correlation, we do not compute action-wise scores but consider instances and actions jointly. Let the index j iterate over instances as well as actions (hence the mappings $i(j)$ and $a(j)$), then scores for an action can be computed by: $\mathbf{r}^{(j)} = \max(0, \langle \mathbf{l}, \mathbf{P}^{i(j)}_{a(j)} \rangle)$ and $\mathbf{q}^{(j)} = \max(0, \langle \mathbf{l}, \mathbf{Q}^{i(j)}_{a(j)} \rangle)$

Now that predictions and ground truths are mapped to a sequence of scalars, we can access the quality of the model's predictions by employing Pearson's correlation coefficient. The resulting score indicates to which degree predicted and ground truth scores are linearly related. We consider this a good measure as it is normalized between -1 and 1 and the top score of 1 or 100% is only attained if scores are identical, except for a scaling factor. In practice, if scores are normalized, the scaling factor becomes irrelevant.

The correlation coefficient is defined as follows:

$$\text{Corr} = \frac{\sum_j (\mathbf{q}^{(j)} - \bar{\mathbf{q}}) * (\mathbf{r}^{(j)} - \bar{\mathbf{r}})}{\sqrt{\sum_j (\mathbf{q}^{(j)} - \bar{\mathbf{q}})^2} \sqrt{\sum_i (\mathbf{r}^{(j)} - \bar{\mathbf{r}})^2}}$$

A problem of the correlation score is that it requires the variance to be computable. If all predictions (or all ground truth scores) are identical, the term $(\mathbf{r}^{(j)} - \bar{\mathbf{r}})$ is zero and causes division by zero. In fact, this case rarely occurs in our experiments, we indicate it by "-". The correlation coefficient is both easy to interpret and captures differences across distributions. However, one might argue that the projection vector is somewhat arbitrary.

5 Experiments

Next, we conduct a series of experiments assessing the quality of the trained networks and relating them to meaningful baselines. First, we show some qualitative results, involving both instance only and context. Quantitatively, we analyze performance concerning context, architecture and training settings using the metrics defined above. For this, we follow the original splits of OpenImages and apply the data processing described above, yielding 12262 training samples, 542 validation samples and 471 test samples. Each of these subsets contains individual images, so the test scores are computed on images that have not been seen before. Samples of the dataset can be found online[1]. While a test set of 542 scenes might seem small, it should be considered that we required all compatible actions to be annotated by the raters (unlike for validation and training samples).

5.1 Qualitative Evaluation

In Figure 5 we present a set of instance images with their associated action plausibilities computed using the single-image InceptionV4-based model as well as the 2xRN50 model which uses the instance image in conjunction with img+full context. Note the variety in the presented samples, ranging from an outdoor cherry tree to different cup close-ups having vastly different illuminations.

The presented samples indicate that the trained model generates useful predictions of the plausibilities of the actions on these unseen samples (own photographs). In the top two rows, most predictions are correct, while only some of them are questionable. The full cup of the coffee should probably neither be stored away nor cleansed. Yet, "drink from" correctly received the largest predicted plausibility for this sample. The same is true for the cleanse item and the empty cups. If a definitive decision is required, one could apply a minimal threshold and then pick the most likely action. We observe that the plausibilities are strongly dependent on the object class. However,

[1] https://storage.googleapis.com/openimages/web/visualizer/index.html

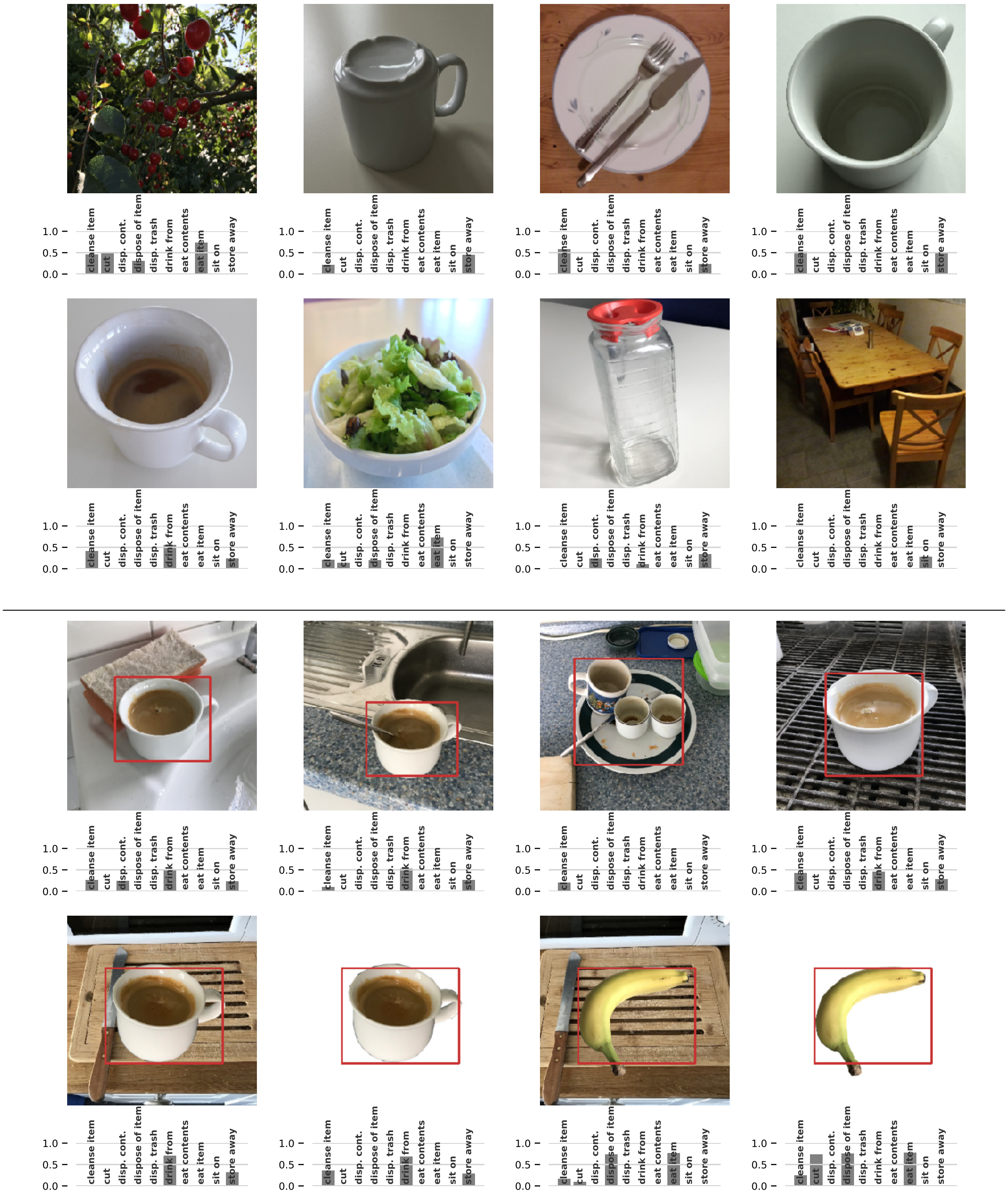


Figure 5: Top two rows: Qualitative samples generated using the InceptionV4-based network. Bottom row: Samples generated using img+full context and the 2xRN50 network (the instance image is indicated in red). All of these images were taken by us; thus they are neither part of any subset of OpenImages nor any other dataset.

this is not true for all cases. For example, the plausibility for drinking is zero for the empty cup while it is the most likely action for the filled cup. Additionally, while the object class often seems to determine the presence of plausible actions, there are fine-grained differences in the individual plausibilities. These differences represent a crucial aspect of the visual common-sense knowledge about household scenes that has been learned. In a robotic context, such differences could be used to compare plausibilities of a given action across multiple objects and then pick the most suitable object.

The qualitative samples that involve context (two bottom rows) suggest that the context has an inhibitory effect on action plausibilities, i.e. the predicted plausibilities tend to be smaller. This can be seen especially in the bottom row that involves the same object on different backgrounds. Here the model predicts more potential actions and assigns slightly higher plausibilities if the background is white compared to the real-world background.

5.2 Quantitative Evaluation

Based on the previously defined baselines and metrics, we begin our quantitative analysis by comparing various training settings, augmentation strengths, and encoder architectures. In subsequent experiments we address special questions investigating how many samples are sufficient, the role of context, the impact of the encoder architecture and several design choices as part of an ablation. Additionally, human performance using the same metrics is assessed and related to the computational models.

5.2.1 Ablation

Training Setting and Augmentation First we assess the impact of several training parameters, introduced above, on the performance. The corresponding results are reported in Table 2. MR refers to the minimal number of ratings required for a sample. While this is per default 1, in case of MR $= 2$ the dataset size is reduced but samples are more reliable. Same rating (SR) means that samples are only accepted when the raters agree (which only makes sense for MR > 1). Moreover, we find that both, pre-training on ImageNet and the validity mask $\mathbf{v}$ in the loss function are crucial for performance. In both cases, performance decreases compared to single rater samples. This suggests that the increased variance introduced by a large dataset weighs more than the increased reliability of multiple ratings per instance. This finding is in accordance with the work of Mahajan et al. [30] where image classifiers were successfully trained on hashtags, despite strong label noise of the latter. In augmentation we find a moderate strength of 2 to perform best.

Encoder The comparison of different encoder architectures, presented in Table 3, indicates that larger models tend to perform better. We attribute this to two reasons: First, they can capture more complex features. Second, their object detection performance is better. Given reliable object detection, it is easier to exploit dataset biases. For a more detailed discussion of this we refer to Sec. 5.2.3. Furthermore, we find all models to exhibit fast inference (25ms to 1s), allowing real-time use for example in robotics.

Besides the shown experiments, we found the batch size to play a critical role for performance and, thus, suggest keeping the batch size as large as possible. Additionally, we tried to use larger images to improve performance without success. We hypothesize that the reason for this is that models strongly benefit from the pre-trained ImageNet weights. This pre-training was done for a fixed image size and the ImageNet dataset is fairly consistent with respect to scale. Hence, the

[1] on an Intel Core i7-2600 CPU with 3.4 GHz, image size: 256x256, batch-size: 1, using PyTorch 1.1.0

Table 2: Ablation of different training settings (top) and augmentation strengths (bottom). Both use the InceptionV4 encoder and context is ignored. VM: using a validity mask $\mathbf{v}$, MR: minimal number of ratings, SR: same ratings only, PT: pre-trained, σ: standard deviation.

Training settings / Augmentation

MR	aug	SR	PT	mask	A-Acc (σ)	CE (σ)	Corr (σ)
-	2	-	✓	✓	**45.3** (1.4)	**0.253** (0.008)	**72.5** (1.6)
2	2	-	✓	✓	39.3 (1.4)	0.290 (0.006)	65.6 (2.0)
2	2	✓	✓	✓	38.4 (1.2)	0.329 (0.007)	60.9 (2.2)
-	2	-	-	✓	24.2 (2.2)	0.410 (0.022)	48.4 (4.0)
-	2	-	✓	-	37.4 (1.1)	0.442 (0.018)	43.4 (1.8)
-	0	-	✓	✓	45.3 (2.2)	0.263 (0.016)	73.7 (0.9)
-	1	-	✓	✓	46.2 (1.0)	0.257 (0.009)	**74.3** (0.9)
-	3	-	✓	✓	**46.5** (1.2)	0.266 (0.016)	74.1 (1.1)
-	5	-	✓	✓	46.0 (0.6)	0.264 (0.016)	73.8 (1.1)
-	7	-	✓	✓	45.9 (2.5)	**0.254** (0.011)	73.6 (1.0)
-	9	-	✓	✓	45.8 (1.2)	0.261 (0.014)	73.4 (1.0)

Table 3: Comparison of different encoders. Context is ignored, and augmentation set to 2. The last line depicts the same setting as the first line of Table 2. T: Single sample inference time[1].

Encoders

model	A-Acc	CE	Corr
SqueezeNet [18]	38.8	0.326	60.2
RN18 [17]	45.1	0.277	70.3
RN50 [17]	44.9	0.300	68.9
RN101 [17]	44.4	0.265	70.9
RN152 [17]	45.3	0.275	73.5
Xception [5]	47.0	0.282	71.7
Inc3 [40]	41.1	0.321	67.8
Inc4 [41]	46.7	0.263	74.9

Table 4: Evaluation per action for selected models (right) as well as comparison to baseline performances (left). The mode baseline does not receive the image as an input but has access to the name of the object shown. "No inp." refers to a RN50 network where all input information is removed by multiplying with zero. σ indicates the standard deviation after executing the training and test ten times for each score. On the right, we show correlation scores for each action individually. [2]

					Corr (per-action)									
model	context	A-Acc (σ)	CE (σ)	Corr (σ)	cleanse-item	cut	dispose-of-contents	dispose-of-item	dispose-trash-in	drink-from	eat-contents	eat-item	sit-on	store-away
Inc4	ignore	46.1 (0.9)	0.257 (0.010)	74.0 (1.1)	54.0	85.3	16.7	70.9	84.2	76.4	23.7	94.0	90.9	65.7
No Inp.	ignore	43.0 (1.4)	0.280 (0.010)	70.1 (1.6)	52.1	76.6	21.8	68.9	78.7	69.7	15.4	91.3	87.3	61.0
2xRN50	img+full	43.5 (2.1)	0.301 (0.019)	70.1 (1.8)	52.8	78.0	18.2	67.8	82.8	66.7	5.4	91.9	86.2	61.8
RN50	ignore	7.0 (0.0)	2.205 (0.023)	-	-	-	-	-	-	-	-	-	-	-
Mode	ignore	50.7 (0.0)	0.641 (0.000)	75.9 (0.0)	59.9	88.3	16.8	71.6	86.8	65.5	-	96.7	94.9	73.0

features encoded by the weights are optimized for this specific size. Possibly, our dataset is too small to cause substantial changes in the features and hence it benefits from objects being provided at the original scale.

5.2.2 Action-wise Evaluation and Comparison to Baselines

The results presented in Table 4 show improvement over the no-input-image baseline. This means the models indeed use information from the presented images to improve predictions. In fact, the no-input-image baseline considers all actions implausible, which is the best guess without knowing the image. However, the mode baseline outperforms our methods in Acc and Corr while our method achieves better CE. This means that our method has advantages at predicting fine-grained differences in the rating distribution, while for coarse accuracy, which neglects details in the distribution, the mode baseline is good enough.

The class-wise scores give more insights. For most action classes, our methods yield a worse accuracy (Acc) than the mode baseline. However, drink-from is a notable exception as it performs much better than mode. This suggests that for drink-from, the image content is crucial and must be considered to make a decision.

Considering the good performance of the mode baseline, it should be noted that it benefits from several factors: First, it knows the object class being depicted, an information that other models have no access to. Evidence for the importance of this is found in the gap between ImageNet pre-trained and untrained models in the ablation (Table 2). Second, it knows the modes of the rating distributions. Since these distributions are far from being uniform, the mode alone often is a powerful predictor for the most likely rating. This means that the mode baseline has an unfair advantage over our method: In practice the information about the object class being shown is obviously not available as it would require a perfect object recognizer. Plugging in a sub-optimal

[2] In some cases the models predicted eat-contents always as impossible. These were ignored when computing the score for eat-contents.

Table 5: Performance on four selected objects: bottle, bowl, wok and box. "No inp." refers to a RN50 network where all input information is removed by multiplying with zero.

model	context	A-Acc	CE	Corr
Mode	ignore	3.0	0.766	62.2
No Inp.	ignore	0.0	3.068	47.2
Inc4	ignore	15.2	0.510	60.3
RN50	ignore	18.2	0.518	60.4
2xRN50	img+full	15.2	0.548	57.1

object recognizer would diminish the performance. Nonetheless, the mode baseline serves as a useful anchor to relate scores to.

When we consider all ratings in the CE metric, the mode baseline does not perform as good anymore. To some extent this is not surprising because the mode baseline always generates one-hot distributions, i.e. vectors where all elements are zero except for one element that is one. Still these fine-grained differences in plausibilities are crucial for many applications in robotics since they enable the comparison and selection across different potential actions.

5.2.3 Selected Objects and Raters

In many cases the rating distribution is highly dependent on the object class, i.e. given the object class we can make the correct prediction without having looked at the image. While this is a natural phenomenon, it interferes with our analysis since we are particularly interested in cases where the image content matters. Hence, we conduct an analysis with a subset of objects whose plausibility rating distribution has a higher entropy. Concretely, these object classes are: bottle, bowl, wok and box. The corresponding results are shown in Table 5. We see that the mode baseline is strongly outperformed in terms of Acc and slightly outperformed on CE. This indicates that the good performance of the mode baseline is an artifact of unbalanced rating distributions.

Similar to picking specific object classes, we can also limit the training data to specific raters. In the most extreme case, we train and test only on data provided by a single rater. Corresponding performance is reported in Table 6. Here we observe substantial improvements over the whole dataset, despite the smaller training sets. Note that we did not cherry-pick this special rater, but this observation remains consistent also when using other raters. However, this suggests a high level of inter-individual differences between raters. From a human-robot interaction viewpoint this is interesting and suggests that person-specific training might be needed to avoid clashes in plausibility assessments of human versus robot. Potentially, it might be sufficient to *condition at runtime* on specific persons by taking personal preferences and habits into account.

Additionally, we select a subset of 3 raters having an average pairwise agreement of 73.4. When we use this set for training and test, we obtain the scores reported in Table 7. Here we see substantially better performance in terms of CE. In addition, the gap between mode baseline and our methods is larger. From the results presented in Tab 6 and Table 7 we conclude that consistency of training and test data is a crucial property. Thus, to achieve better overall performance when using more raters, data must be gathered in a more consistent way.

5.2.4 Rater Reliability

Having only compared scores obtained from different computational methods so far, a natural question is: How consistent are the ratings provided by humans? For this, we apply the metrics in-

Table 6: Performance of all eight individual raters with Inc4 model ignoring context. D denotes dataset size.

D	A-Acc	CE	Corr
2894	73.9	0.092	88.6
2621	59.4	0.168	82.0
2365	70.6	0.109	88.4
1841	66.7	0.160	78.1
3300	72.2	0.086	80.9
1874	56.5	0.200	71.8
1914	73.0	0.097	85.7
1575	44.4	0.193	88.9

Table 7: Performance on three selected raters having high agreement.

model	context	A-Acc	CE	Corr
Mode	ignore	67.1	0.611	83.0
No Inp.	ignore	1.8	2.087	-
Inc4	ignore	59.3	0.190	82.8
RN50	ignore	50.9	0.212	77.6
2xRN50	img+full	56.3	0.208	77.8

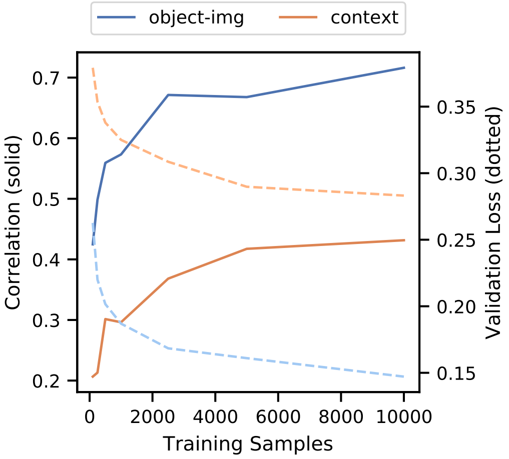


Figure 6: Performance for different numbers of training samples.

troduced above on pairs of human raters. By averaging all pairwise scores we obtain the following: Acc of 42.0, CE of 0.347 and a Corr of 44.8. While Acc is comparable to some models, in terms of CE and Corr the raters perform significantly worse than the computational methods. If we require a minimal intersection of 100 instances to compensate for statistically unreliable data points, we obtain slightly better scores.

We also tracked the self-consistency of the raters by presenting selected instances twice within the collection of all instances. Since the raters were free to select which samples they annotate, not all of them annotated these instances. However, across those who did, the self-consistency varies between 0.77 and 1.0 with an average of 0.90. The number of samples that were annotated twice ranges from 1 to 26 with an average of 13.1.

5.2.5 Scalability

The number of training samples is a quantity that normally has a strong impact on the performance. Since we are collecting the data, it is crucial to understand the effect of the training sample size to avoid an insufficiently small dataset. Figure 6 provides an overview on the relationship between training samples and performance in terms of correlation. It suggests that the dataset is large

Table 8: Comparison of different context representations. σ indicates the standard deviation after executing training and testing ten times for each score. BS: batch size.

model	context	BS	A-Acc (σ)	CE-mean (σ)	Corr (σ)
RN50	ignore	32	42.1 (1.6)	0.301 (0.017)	70.3 (0.9)
Inc4	ignore	32	44.9 (1.6)	0.267 (0.012)	72.3 (1.0)
RN50	only-masked	32	30.4 (2.4)	0.425 (0.012)	49.3 (2.4)
RN50	only-full	32	36.8 (1.3)	0.374 (0.015)	62.4 (2.2)
2xRN50	img+full	24	43.9 (1.8)	0.304 (0.015)	71.2 (1.3)
2xRN50	img+masked	24	43.4 (2.2)	0.309 (0.021)	69.1 (2.6)
2xRN101	img+full	24	41.4 (2.2)	0.314 (0.013)	70.0 (1.6)
2xRN101	img+masked	24	42.3 (2.0)	0.305 (0.013)	69.4 (2.2)

enough and no major improvements could be expected from gathering more data.

We find that already a fairly small number of annotated scenes (around 3000) allows models to attain a good performance. This is indicated by a high correlation of around 0.65 between predictions and ground truth probabilities in Figure 6. More samples further improve on performance, although at a smaller rate.

A straightforward way to obtain more ratings would be to hire more annotators. While this would incur some cost, this clearly is possible due to the linear relationship between cost and number of annotations. To extend the number of action categories, it would be necessary to define compatibility with all 600 object classes from OpenImages for each new action. However, since the set of useful actions (possibly in the hundreds) is limited we consider this feasible, too.

5.2.6 Context

Not only the appearance of an object is relevant for actions, potentially also the context can give hints about the status of an object. Having introduced context representations in Sec. 4.4, here we run an explicit comparison of the representations.

From Table 8 we observe that context with the instance object being masked (only-masked) helps to predict actions but does not achieve the performance of showing the object itself (ignore context). When the instance image is combined with a context representation (img+full), the Corr and A-Acc slightly improve compared to RN50 ignoring context. However, this improvement is fairly small. This is probably due to small parts of the context being included in the image itself. The information that can be extracted from a bigger context is therefore negligible and does not outweigh the problems of having more parameters. Relying exclusively on the context does not seem to be a good idea. This is not surprising because the object appearance clearly gives hints about possible actions. Using the larger 2xRN101 model does not improve on performance. This is possibly due to its larger number of parameters making it more prone to overfitting.

5.3 Comparison with State-of-the-Art

As discussed in 2, our method can not be compared as a whole to state-of-the-art methods due to the use of different labels for actions. Possibly, the most similar method to ours is the work by Vu et al. [44]. Their approach shares with ours the goal of predicting actions from static images. We use the CNNs from our method and train them on their SUNAction dataset. We report corresponding scores in Tab. 9. The CNNs we employ in our models clearly outperform the bag-of-word and fisher-vector-based methods proposed by Vu et al. [44], when the CNNs are initialized

Table 9: Comparison of our classification model against state-of-the-art approaches. No augmentation is used. The numbers in brackets indicate standard deviation over five runs.

model	PT	location	mAP (σ)
RN18	-	indoor	70.4 (18.4)
RN18	✓	indoor	79.4 (6.9)
RN50	✓	indoor	85.2 (7.1)
Inc4	✓	indoor	84.9 (5.6)
RN18	-	outdoor	73.4 (14.4)
RN18	✓	outdoor	82.7 (4.0)
RN50	✓	outdoor	78.4 (4.5)
Inc4	✓	outdoor	81.7 (8.6)
Sift BoW [44]		indoor	40.9
HOG BoW + HOG FV + CSIFT FV [44]		indoor	61.0
Sift BoW [44]		outdoor	31.8
HOG BoW + HOG FV + CSIFT FV [44]		outdoor	52.0

with weights obtained through ImageNet pre-training. Note, this finding is largely unsurprising given the dominance of CNNs for image recognition. Interestingly, the advantage of ImageNet pre-training is fairly small although it strongly reduces variance. However, it links our approach to the existing body of research.

6 Conclusion

Summary In this paper, we established a framework of how to gather action plausibility ratings, creating a dataset called ”PlausiblAct”, transform them to train neural networks, and evaluate the corresponding results.

After defining a set of ten actions and three ratings, we presented our sparse data collection method relying on web techniques allowing for a fast and comparatively effortless data annotation. Next, these ratings of object instances were transformed into distributions on which a neural network could be trained to make action-oriented predictions. To assess the quality of these predictions we proposed three metrics capturing complementary quality aspects.

In our comparison of state-of-the-art feature encoders, we find the InceptionV4 network to be suited best for the task. The experiments suggest that object-classification performance is still a crucial factor for scoring action plausibilities. Combinations with context seem to improve the performance slightly while context alone, ignoring the actual objects’ appearances, leads to fairly bad action plausibility predictions. The experiments demonstrated that our system can operate on *data-in-the-wild*. This means, that our system works well on images that were not intentionally shot for our purpose but randomly chosen.

Limitations and Future Work The presented approach has some limitations. So far, the system is limited to generating a plausibility distribution over a set of ten actions based on image input, which needs to be extended in future works. We also found that inter-rater reliability is quite low and might be improved. Consequently, a key question for follow-up work is how to design data collection paradigms that enforce reliability and consistency across the raters more rigorously than the presented one while extending the set of considered actions. Furthermore, the models we employed are simple image classification models that are not specifically designed for reasoning.

Future work might involve reasoning-oriented models, e.g. the relation network [38]. So far, we excluded scenes depicting humans from the data as far as possible. As a potential next step, showing humans could increase the complexity of this or similar approaches as intentions would need to be estimated, too. If humans are visible, action plausibility could be learned by observing human behavior in video. While this would eliminate the necessity for labeling scenes, it might come with its own challenges like action detection. Nonetheless, it is an alternative approach which might be worth pursuing. For the actual robotic execution of a task, the desired poses of the respective objects would need to be calculated and trajectories of the robot have to be generated, too. For both of these problems working algorithms exist, hence here we had exclusively focused on inferring actions in this paper not attempting to actually execute them on a machine.

In an interactive scenario, our method could be complemented by using reinforcement learning, where our pre-trained models (involving strong augmentation) can help by providing a good prior for selecting actions (the policy) that can later be refined in an interactive environment.

Such a supervised training of policies for reinforcement learning, specifically mapping from images to motor torques, was shown to work well in previous research [26]. Another approach employs simulated images in a supervised training setting to detect objects with the goal of using this system as a policy for reinforcement learning [42]. Böhmer et al. [2] carried out a survey on slow feature analysis- and autoencoders-based methods that learn state representations from visual data for reinforcement learning.

Extensibility Currently, symbolic intermediate representations are entirely evaded by our method. Yet, our system could be extended to be partially symbolic by explicitly detecting objects and their states. Such symbolic information could then be processed together with instance images to infer action plausibilities. However, this would require enumerating many possible states rather and contradicts the original idea of a direct mapping.

An advantage of the proposed method is that it can be combined with other robotic algorithms. This also holds true for the following example: Assume a scene involving a dirty and a clean cup and the instruction "put cup into dishwasher". Although it is obvious to humans that the instruction refers to the dirty cup, this common-sense knowledge is not available to the machine. By using our method, the system can evaluate images of both cups and then pick the one for which the action "cleanse" is more plausible. Thus, potential applications, where we expect action plausibilities to be helpful, concern robotic action planning, where our method allows better disentangling action preconditions needed by the planning operators. Thus, in conclusion, we believe that the here-presented method has many potential use-cases and that it should be possible to extend it without too much effort.

Acknowledgements

We thank the reviewers for their valuable comments and feedback. The research leading to these results has received funding from the European Community's H2020 Programme ICT-2016-2017/H2020-ICT-2016-1 under grant agreement no. 73761, IMAGINE.

References

[1] D. A. Allport. Selection for action: Some behavioral and neurophysiological considerations of attention and action. *Perspectives on perception and action*, 15:395–419, 1987. 4

[2] W. Böhmer, J. T. Springenberg, J. Boedecker, M. Riedmiller, and K. Obermayer. Autonomous learning of state representations for control: An emerging field aims to autonomously learn state representations for reinforcement learning agents from their real-world sensor observations. *KI - Künstliche Intelligenz*, 29 (4):353–362, Nov 2015. ISSN 1610-1987. doi: 10.1007/s13218-015-0356-1. URL `https://doi.org/10.1007/s13218-015-0356-1`. 21

[3] A. M. Borghi. Affordances, context and sociality. *Synthese*, pages 1–31, 2018. 4

[4] Y.-W. Chao, J. Yang, B. Price, S. Cohen, and J. Deng. Forecasting human dynamics from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 548–556, 2017. 4

[5] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017. 15

[6] P. Cisek. Cortical mechanisms of action selection: The affordance competition hypothesis. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1485):1585–1599, 2007. 4

[7] S. H. Creem and D. R. Proffitt. Grasping objects by their handles: a necessary interaction between cognition and action. *Journal of experimental psychology: Human Perception and Performance*, 27(1):218, 2001. 4

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009. 9

[9] T.-T. Do, A. Nguyen, and I. Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *International Conference on Robotics and Automation (ICRA)*, 2018. 4

[10] N. El-Sourani, I. Trempler, M. F. Wurm, G. R. Fink, and R. I. Schubotz. Predictive impact of contextual objects during action observation: Evidence from functional magnetic resonance imaging. *Journal of Cognitive Neuroscience*, 0(0):1–13, 2019. doi: 10.1162/jocn_a_01480. URL `https://doi.org/10.1162/jocn_a_01480`. PMID: 31617822. 4

[11] D. F. Fouhey and C. L. Zitnick. Predicting object dynamics in scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 4

[12] J. Franchak and K. Adolph. Affordances as probabilistic functions: Implications for development, perception, and decisions for action. *Ecological Psychology*, 26(1-2), 2014. 2

[13] J. J. Gibson. *The Senses Considered as Perceptual Systems*. Houghton Mifflin, 1966. 4

[14] J. J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979. 4

[15] H. Grabner, J. Gall, and L. Van Gool. What makes a chair a chair? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011. 4

[16] T. C. Handy, S. T. Grafton, N. M. Shroff, S. Ketay, and M. S. Gazzaniga. Graspable objects grab attention when the potential for action is recognized. *Nature neuroscience*, 6(4):421, 2003. 4

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 9, 15

[18] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 15

[19] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, International Conference on Machine Learning (ICML), pages 448–456. JMLR.org, 2015. URL `http://dl.acm.org/citation.cfm?id=3045118.3045167`. 9

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 9

[21] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. In *Robotics: Science and Systems*, 2013. 3

[22] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013. ISSN 0278-3649, 1741-3176. doi: 10.1177/0278364913478446. CAD120 dataset, CAD 120. 4

[23] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, S. Kamali, M. Malloci, J. Pont-Tuset, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017. 2, 5, 7

[24] N. Krüger, C. Geib, J. Piater, R. Petrick, M. Steedman, F. Wörgötter, A. Ude, T. Asfour, D. Kraft, D. Omrcen, A. Agostini, and R. Dillmann. Object–action complexes: Grounded abstractions of sensory–motor processes. *Robotics and Autonomous Systems (RAS)*, 59(10):740 – 757, 2011. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2011.05.009. URL http://www.sciencedirect.com/science/article/pii/S0921889011000935. 4

[25] T. Lan, T.-C. Chen, and S. Savarese. A hierarchical representation for future action prediction. In *European Conference on Computer Vision*, pages 689–704. Springer, 2014. 3

[26] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research (JMLR)*, 17(1):1334–1373, 2016. 21

[27] O. Lindemann, P. Stenneken, H. T. Van Schie, and H. Bekkering. Semantic activation in action planning. *Journal of Experimental Psychology: Human Perception and Performance*, 32(3):633, 2006. 4

[28] T. Lüddecke and F. Wörgötter. Learning to segment affordances. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 769–776, 2017. 4

[29] T. Lüddecke, T. Kulvicius, and F. Wörgötter. Context-based affordance segmentation from 2d images for robot action. *Robotics and Autonomous Systems (RAS)*, 2019. 4

[30] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten. Exploring the limits of weakly supervised pretraining. In *European Conference on Computer Vision (ECCV)*, pages 181–196, 2018. 14

[31] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos. Affordance detection of tool parts from geometric features. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015. 4

[32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems Workshops*, 2017. 9

[33] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu. Learning human-object interactions by graph parsing neural networks. In *European Conference on Computer Vision (ECCV)*, 2018. 4

[34] N. Rhinehart and K. M. Kitani. Learning action maps of large environments via first-person vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–588, 2016. 4

[35] M. J. Riddoch, G. W. Humphreys, S. Edwards, T. Baker, and K. Willson. Seeing the action: Neuropsychological evidence for action-based effects on object selection. *Nature neuroscience*, 6(1):82, 2003. 4

[36] K. L. Roberts and G. W. Humphreys. Action relations facilitate the identification of briefly-presented objects. *Attention, Perception, & Psychophysics*, 73(2):597–612, 2011. 4

[37] A. Roy and S. Todorovic. A multi-scale cnn for affordance segmentation in rgb images. In *European Conference on Computer Vision (ECCV)*. Springer, 2016. 4

[38] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4967–4976, 2017. 21

[39] A. K. Seth. The ecology of action selection: Insights from artificial life. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1485):1545–1558, 2007. 4

[40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 15

[41] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 4, 2017. 9, 15

[42] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30. IEEE, 2017. 21

[43] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating visual representations from unlabeled video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[44] T.-H. Vu, C. Olsson, I. Laptev, A. Oliva, and J. Sivic. Predicting actions from static scenes. In *European Conference on Computer Vision (ECCV)*. Springer, 2014. 4, 19, 20

[45] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference Computer Vision (ECCV)*, 2016. 4

[46] X. Wang, R. Girdhar, and A. Gupta. Binge watching: Scaling affordance learning from sitcoms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4

[47] F. Wörgötter, A. Agostini, N. Krüger, N. Shylo, and B. Porr. Cognitive agents—a procedural perspective relying on the predictability of object-action-complexes (oacs). *Robotics and Autonomous Systems (RAS)*, 57(4):420–432, 2009. 4

[48] C. Ye, Y. Yang, C. Fermüller, and Y. Aloimonos. What can i do around here? deep functional scene understanding for cognitive robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 4

[49] B. Zhou, A. Andonian, A. Oliva, and A. Torralba. Temporal relational reasoning in videos. In *European Conference on Computer Vision (ECCV)*, pages 803–818, 2018. 3

[50] Y. Zhu, A. Fathi, and L. Fei-Fei. Reasoning about object affordances in a knowledge base representation. In *European Conference on Computer Vision (ECCV)*, pages 408–424. Springer, 2014. 4

[51] M. Zolfaghari, K. Singh, and T. Brox. Eco: Efficient convolutional network for online video understanding. In *European Conference on Computer Vision (ECCV)*, 2018. 3

Appendix

Augmentation

In the following we describe the operations used for augmentation. Here, a is an integer that generally expresses the strength of augmentation to reduce complexity. $\mathcal{N}(\mu, \sigma)$ denotes a Gaussian distribution with mean μ and standard deviation σ:

Horizonal Flip	Flip image horizontally with a probability of 0.5
Crop	Sample random crop of size (H', W') from original image of size (H, W) $W' = W - \texttt{randint}(1, 0.05 \cdot a \cdot W)$ and $H' = H - \texttt{randint}(1, 0.05 \cdot a \cdot H)$
Gamma correction	Change image gamma by channel-wise random values G_r, G_g, G_b which are computed according to: $G_r, G_g, G_b \sim \mathcal{N}(1, 0.05 \cdot a) + G_{all}$ with $G_{all} \sim \mathcal{N}(1, 0.1 \cdot a)$ Each value is clipped to be in the interval [0.1, 1.9] .
Color offset	A channel-wise offset O is added to the image (values ranging between 0 and 255) with $O \sim \mathcal{N}(0, 4 \cdot a)$. The offset is applied after the gamma correction.

6.1 Object-Action Compatibility

	drink from	cut	sit on	eat item	eat contents	cleanse item	store away	dispose of item	dispose of contents	dispose trash in
Toothbrush	-	-	-	-	-	✓	-	✓	-	-
Apple	-	✓	-	✓	-	✓	-	✓	-	-
Chopsticks	-	-	-	-	-	✓	✓	-	-	-
Croissant	-	✓	-	✓	-	-	-	✓	-	-
Cucumber	-	✓	-	✓	-	✓	-	✓	-	-
Radish	-	✓	-	✓	-	✓	-	✓	-	-
Hot dog	-	✓	-	✓	-	-	-	✓	-	-
Waffle	-	✓	-	✓	-	-	-	✓	-	-
Pancake	-	✓	-	✓	-	-	-	✓	-	-
Pretzel	-	✓	-	✓	-	-	-	✓	-	-
Bagel	-	✓	-	✓	-	-	-	✓	-	-
Teapot	✓	-	-	-	-	✓	✓	-	-	-
Popcorn	-	-	-	✓	-	-	-	✓	-	-
Burrito	-	✓	-	✓	-	-	-	✓	-	-
Scissors	-	-	-	-	-	-	✓	-	-	-
Chair	-	-	✓	-	-	-	-	-	-	-
Muffin	-	✓	-	✓	-	-	-	✓	-	-
Cookie	-	✓	-	✓	-	-	-	✓	-	-
Calculator	-	-	-	-	-	-	✓	-	-	-
Box	-	-	-	-	-	-	✓	✓	✓	✓
Stapler	-	-	-	-	-	-	✓	-	-	-
Studio couch	-	-	✓	-	-	-	-	-	-	-

Zucchini	-	✓	-	✓	-	✓	-	✓	-	-
Ladle	✓	-	-	-	-	✓	✓	-	-	-
Winter melon	-	✓	-	✓	-	✓	-	✓	-	-
Spatula	-	-	-	-	-	✓	✓	-	-	-
Pencil sharpener	-	-	-	-	-	-	✓	-	-	-
Eraser	-	-	-	-	-	-	✓	-	-	-
Tin can	✓	-	-	-	-	-	✓	✓	✓	-
Mug	✓	-	-	-	-	✓	✓	-	-	-
Can opener	-	-	-	-	-	✓	✓	-	-	-
Coffee cup	✓	-	-	-	-	✓	✓	-	-	-
Cutting board	-	-	-	-	-	✓	✓	-	-	-
Vase	-	-	-	-	-	✓	✓	-	-	-
Slow cooker	-	-	-	-	-	✓	✓	-	✓	-
Whisk	-	-	-	-	-	✓	✓	-	-	-
Salt and pepper shakers	-	-	-	-	-	✓	✓	-	-	-
French fries	-	✓	-	✓	-	-	-	✓	-	-
Tart	-	✓	-	✓	-	-	-	✓	-	-
Egg	-	-	-	✓	-	-	-	✓	-	-
Grape	-	✓	-	✓	-	✓	-	✓	-	-
Mixing bowl	✓	-	-	-	-	✓	✓	-	-	-
Hammer	-	-	-	-	-	-	✓	-	-	-
Sofa bed	-	-	✓	-	-	-	-	-	-	-
Adhesive tape	-	-	-	-	-	-	✓	-	-	-
Saucer	-	-	-	-	✓	✓	✓	-	✓	-
Drinking straw	-	-	-	-	-	✓	✓	-	-	-
Common fig	-	✓	-	✓	-	✓	-	✓	-	-
Cocktail shaker	✓	-	-	-	-	✓	✓	-	-	-
Artichoke	-	✓	-	✓	-	✓	-	✓	-	-
Knife	-	-	-	-	-	✓	✓	-	-	-
Bottle	✓	-	-	-	-	✓	✓	✓	✓	-
Bottle opener	-	-	-	-	-	✓	✓	-	-	-
Bowl	✓	-	-	-	✓	✓	✓	-	✓	-
Frying pan	-	-	-	-	✓	✓	✓	-	✓	-
Ring binder	-	-	-	-	-	-	✓	-	-	-
Plate	-	-	-	-	✓	✓	✓	-	✓	-
Pitcher	✓	-	-	-	-	✓	✓	-	-	-
Pencil case	-	-	-	-	-	-	✓	-	-	-
Kitchen knife	-	-	-	-	-	✓	✓	-	-	-
Plastic bag	-	-	-	-	-	-	✓	✓	✓	✓
Potato	-	✓	-	✓	-	✓	-	✓	-	-
Pasta	-	-	-	✓	-	-	-	✓	-	-
Pumpkin	-	✓	-	✓	-	✓	-	✓	-	-
Pear	-	✓	-	✓	-	✓	-	✓	-	-
Infant bed	-	-	✓	-	-	-	-	-	-	-
Pizza	-	✓	-	✓	-	-	-	✓	-	-
Submarine sandwich	-	-	-	✓	-	-	-	✓	-	-
Loveseat	-	-	✓	-	-	-	-	-	-	-
Coffee table	-	-	✓	-	-	-	-	-	-	-
Taco	-	-	-	✓	-	-	-	✓	-	-
Strawberry	-	✓	-	✓	-	✓	-	✓	-	-
Tomato	-	✓	-	✓	-	✓	-	✓	-	-
Measuring cup	-	-	-	-	-	✓	✓	-	-	-
Paper cutter	-	-	-	-	-	-	✓	-	-	-
Wok	-	-	-	-	✓	✓	✓	-	✓	-
Jug	-	-	-	-	-	✓	✓	-	-	-
Pizza cutter	-	-	-	-	-	✓	✓	-	-	-
Bread	-	✓	-	✓	-	-	-	✓	-	-
Platter	-	-	-	-	-	✓	✓	-	-	-
Toilet paper	-	-	-	-	-	-	✓	-	-	-
Lemon	-	✓	-	✓	-	✓	-	✓	-	-
Banana	-	✓	-	✓	-	✓	-	✓	-	-
Wine glass	✓	-	-	-	-	✓	✓	-	-	-

Countertop	-	-	✓	-	-	-	-	-	-	-
Waste container	-	-	-	-	-	-	-	-	-	✓
Book	-	-	-	-	-	-	✓	-	-	-
Hamburger	-	-	-	✓	-	-	-	✓	-	-
Asparagus	-	✓	-	✓	-	✓	-	✓	-	-
Spoon	-	-	-	-	✓	✓	✓	-	✓	-
Oyster	-	-	-	✓	-	-	-	✓	-	-
Ice cream	-	-	-	✓	-	-	-	✓	-	-
Orange	-	✓	-	✓	-	✓	-	✓	-	-
Beaker	✓	-	-	-	-	-	-	✓	-	-
Peach	-	✓	-	✓	-	✓	-	✓	-	-
Fork	-	-	-	-	✓	✓	✓	-	✓	-
Cabbage	-	✓	-	✓	-	✓	-	✓	-	-
Carrot	-	✓	-	✓	-	✓	-	✓	-	-
Mango	-	✓	-	✓	-	✓	-	✓	-	-
Pineapple	-	✓	-	✓	-	✓	-	✓	-	-
Stool	-	-	✓	-	-	-	-	-	-	-
Envelope	-	-	-	-	-	-	✓	✓	-	-
Cake	-	-	-	✓	-	-	-	✓	-	-
Candy	-	-	-	✓	-	-	-	✓	-	-
Salad	-	✓	-	✓	-	✓	-	✓	-	-
Serving tray	-	-	-	-	-	✓	✓	-	-	-
Kitchen and dining room table	-	-	✓	-	-	-	-	-	-	-
Cake stand	-	-	-	-	-	✓	✓	-	-	-
Broccoli	-	✓	-	✓	-	✓	-	✓	-	-
Grapefruit	-	✓	-	✓	-	✓	-	✓	-	-
Bell pepper	-	✓	-	✓	-	✓	-	✓	-	-
Pomegranate	-	✓	-	✓	-	✓	-	✓	-	-
Doughnut	-	✓	-	✓	-	-	-	✓	-	-
Pen	-	-	-	-	-	-	✓	-	-	-
Watermelon	-	✓	-	✓	-	✓	-	✓	-	-
Cantaloupe	-	✓	-	✓	-	✓	-	✓	-	-