UNIVERSITÄT BONN

WEAKLY AND SEMI SUPERVISED SEMANTIC SEGMENTATION OF RGB IMAGES

DISSERTATION

zur Erlangung des Doktorgrades (*Dr. rer. nat.*)

der Mathematisch-Naturwissenschaftlichen Fakultät

der Rheinischen Friedrich–Wilhelms–Universität, Bonn

vorgelegt von

JOHANN SAWATZKY

aus

Barnaul, Russland

Bonn, 2020

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich–Wilhelms–Universität Bonn

1. Gutachter / 1st Advisor: Prof. Dr. Juergen Gall
2. Gutachter / 2nd Advisor: Prof. Dr. Fred Hamprecht
Tag der Promotion / Day of Promotion: 15.12.2020
Erscheinungsjahr / Year of Publication: 2021

Abstract

by Johann Sawatzky

for the degree of

Doctor rerum naturalium

Teaching machines semantic scene understanding from RGB images received a lot of attention in recent years, since this ability is crucial for several applications like autonomous driving, robotics or video surveillance. Large datasets with dense annotations provided by humans and deep learning methods trained on them boosted the performance in semantic segmentation from mediocre to human level. Still, these methods suffer from a major shortcoming. They require expensive human annotations as soon as a new semantic class has to be learnt.

To reduce the annotation effort by orders of magnitude, one can follow the weakly supervised semantic segmentation paradigm and reduce the cost per image by using cheaper localisation cues like keypoints or bounding boxes instead of precise polygons. Alternatively one can only annotate a fraction of the images in the training set and learn from them as well as the unlabeled ones which would constitute a semi supervised approach.

The first part of this thesis concerns object part affordance (functional attribute) segmentation using keypoints as supervision cues. To this end, we introduce a custom dataset with affordance annotations on a pixel level. Additionally, we propose a method that performs significantly better than weakly supervised semantic segmentation methods originally designed for objects. Interestingly, our method generalizes to affordances of novel object classes not present in the train set. Subsequently, we improve upon this method with a second one. One of the strengths of it is the stochastic approximation of the Jaccard index which allows for proper hyper parameter choice even in the absence of ground truth for precise cross validation.

The second part of the thesis treats a setup where object level bounding boxes are given and object part affordances have to be segmented. We propose to annotate the affordances for a tiny number of example objects and then propagate them to the rest of the training set. This way, approximations to ground truth can be obtained for a constant cost.

After this we leave the domain of object part affordances and tackle weakly supervised semantic segmentation of object classes using image captions as supervision cues. Image captions not only provide additional object localization cues in form of object attributes but are also freely available on the internet. Using images and their corresponding captions, we train a multi-modal learning approach to locate arbitrary text snippets in an image. We then use it to provide high confidence object class areas in training images which are superior to those obtained from manually curated image tags.

Finally we consider a semi supervised semantic segmentation setup with pixel-wise labels given for a small fraction of images and no supervision cues of any kind for the rest. We propose a method which discovers latent classes maximizing the information gain about the semantic classes on labeled data. On unlabeled data, we use the consistency between

the latent classes and the semantic classes as a supervision signal. We show that supervision through latent classes is complementary to other consistency signals like neural discriminators. Furthermore, we show that latent classes learned automatically are superior to manually defined supercategories.

All approaches are compared to contemporary state-of-the-art methods and show an improvement compared to them.
Keywords: Semantic Segmentation, Weakly Supervised Learning, Semi-Supervised Learning, Affordances

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

An alphabetically sorted list of abbreviations used in the thesis:

CRF	Conditional random field
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
DNN	Deep Neural Network
EM	Expectation Maximization
GAN	Generative Adversarial Network
IoU	Intersection over Union
MAP	Maximum-a-Posteriori
SGD	Stochastic gradient descent

Frequently Used Symbols

$\mathbf{I}$	Image
$\mathcal{G}$	Graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$
$\mathcal{V}$	Vertices of graph $\mathcal{G}$
$\mathcal{E}$	Edges of graph $\mathcal{G}$
E	Energy function
$\mathcal{L}$	set of latent classes
L	loss
λ	weight of loss term
P	Probability
S	Segmentation prediction
Z	observed variables (weak supervision cues)
X	geometrical pixel positions
M	moving average
D	discriminator network
f	network features
τ	threshold hyper parameter for spatial distance of pixels
α, β	hyper parameters depending on context
h, w	height and width
$\phi(\cdot)$	Unary potential
$\psi(\cdot, \cdot)$	Pairwise potential
θ	Parameters of the model
c	semantic class
l	latent class

y	binary label
$\mathbf{e}$	semantic embedding vector

List of Publications

- **Discovering Latent Classes for Semi-Supervised Semantic Segmentation**
  Zatsarynna O., Sawatzky J.* and Gall J.*
  GCPR 2020

- **Harvesting Information from Captions for Weakly Supervised Semantic Segmentation**
  Sawatzky J., Banerjee D., and Gall J.
  Cross-Modal Learning in Real World Workshop in conjuction with ICCV 2019

- **Ex Paucis Plura: Learning Affordance Segmentation from Very Few Examples**
  Sawatzky J., Garbade M., and Gall J.
  GCPR 2018

- **Adaptive Binarization for Weakly Supervised Affordance Segmentation**
  Sawatzky J. and Gall J.
  International Workshop on Assistive Computer Vision and Robotics
  ICCV 2017

- **Weakly Supervised Affordance Detection**
  Sawatzky J., Srikantha A.*, and Gall J.*
  CVPR 2017

Acknowledgements

Approaching the end of the way towards my PhD I would like to thank the many people who made it possible with their support. Above all, it is my PhD supervisor, Prof. Juergen Gall. He provided support by generating ideas and giving strategic feedback during the early stages of the research projects. During conference submission deadlines, he has done a tremendous amount of work to decisively improve our submissions.

Besides I would like to thank Prof. Fred Hamprecht, Prof. Thomas Schultz and Prof. Stefan Linden for accepting to be part of my PhD evaluation committee.

Prof. Erhardt Barth played an important role ushering me into the world of computer vision and deep learning during his lectures and my master thesis.

Next I would like to thank my colleagues in the group for advice, discussions and the enjoyable work atmosphere. My work would not have been possible without our IT infrastructure. Many thanks to Martin Garbade and Julian Tanke who beared the load of maintaining it as well as Christian Grund who volunteered to help reviving it after the movement of the institute in April 2018. Not to forget the scheduling tool of Alexander Richard which boosted the performance of our clusters.

A special thank goes to my co-authors: Abhulash Srikantha, who gave me a head start into the PhD, Olga Zatsarynna, who in turn took over my last work and of course Yaser Souri who was working with me for one and a half years on a project that was considered as *the real deep learning* by some at CVPR 2019 but unfortunately did not fit thematically into this thesis. With Martin Garbade we even cooperated on two papers and had very interesting insights when doing this.

Last but not least I would like to thank my family and friends for backing me during the periods of uncertainty.

CHAPTER 1

Introduction

Contents

1.1 Motivation

One of the most important technological achievements of the last 2 decades are algorithms for semantic data analysis. These include for example search engines or recommendation systems based on user profiling. A crucial prerequisite for these algorithms are huge amounts of data, which emerged as the Internet became more and more popular. A huge portion of the data is human understandable, i.e texts in human language, images, videos or audio files, which means that doing data mining on these files essentially means teaching a machine to read, see and listen like humans.

So far, intelligent algorithms are mainly confined to the virtual world. However, a machine able to semantically understand real world scenes as well as a human can substitute his eyes and more importantly judgement just as the steam engine substituted his manual effort. The most promising autonomous agents right now are probably autonomous cars.

An autonomous agent needs a granular semantic understanding of the scene: It needs to assign a class and instance label to each pixel in the scene. The task of semantic segmentation deals with the former. A legitimate question is why we aim at segmenting RGB images instead of *e.g.* RGBD images or infrared images. At the end of the day, an autonomous agent does not need to use the same modality as a human and instead use the additional information of *e.g.* depth. The justification of RGB images is the abundance of data and the low price and ubiquity of RGB cameras compared to other imaging sensors. A semantic segmentation method for RGB images can train on a much bigger corpus of data and be deployed anywhere you can find a camera without additional costs for e.g a depth sensor or an infra red sensor.

Apart from being a fundamental component for the scene understanding of autonomous agents, semantic segmentation is applicable in a number of domains on its own. This is the case whenever physical matter needs to be classified into multiple categories, like in medical imaging, quality control or aerial surveillance.

While being extremely useful, semantic segmentation is also extremely costly: To train a machine in a vanilla fully supervised setup, one first has to provide it with data, where a class label is assigned to each pixel. To get such annotations, human annotators draw polygons around regions with the same semantic class. One accurate polygon takes at least half a minute and if we assume that a street

scene is composed of 20 polygons (which is a modest estimate), a human can finish 6 such images in an hour. Then, to creating a dataset covering 1000 cities with 1000 images each requires 166,666 hours of human work which amounts to ca. 300000$ on Amazon Turk. This calculation does not take quality control into account or the fact that for every new semantic category the whole dataset needs to be relabeled. Obtaining a dataset of reasonable size for autonomous driving takes a substantial amount of money equivalent to hiring two machine learners for a year.

Given this financial burden, it is imperative to examine what is achievable with sparser amount of annotation. In weakly supervised semantic segmentation, one provides cheaper supervision cues for all images in the dataset. These can be image tags indicating the present semantic classes or some types of localization cues, like keypoints, scribbles or bounding boxes. The supervision cues can also be related to the semantic classes but not indicating them, like *e.g.* bounding boxes around objects which parts need to be segmented or generic image captions describing the scene and not targeting any specific semantic classes. The latter setup is especially interesting, since these captions are available on the Internet for free. In semi supervised semantic segmentation one provides pixel-wise labels for a fraction of the dataset (*e.g.* 10% or 5%) and no annotations or cheap supervision cues for the remaining data. The challenge here is to provide some supervision signal on images without any human annotation.

1.2 Challenges

Even in a fully supervised setup, semantic segmentation remains challenging. As a logical extension of image classification, it inherited all the problems from it, like high intra-class appearance variation, different view points, scales and illumination conditions, truncation and occlusion. Often the objects of interest (*e.g.* distant pedestrians in street scene datasets) are so small that even a human would not be able to classify them from the interior of the bounding box around the object but would rely on context instead. These problems become even more severe since semantic segmentation has not only to detect the semantic class but estimate its precise location. Typically, methods for image classification are designed to be invariant to translations and rotations of the image content. However, exactly this property is not desirable for semantic segmentation systems. Tubes holding traffic signs are a good example: Detecting their rough position is not enough, the prediction must closely match the ground truth, and since the tubes are extremely thin, each pixel matters.

Semantic segmentation is not solved yet. The impact of a high number of classes, small objects only recognizable by context and a high intra-class appearance variation was demonstrated on the COCO stuff dataset (*Caesar et al.*, 2018). Methods giving over 80% Jaccard index on the Pascal VOC 2012 dataset (*Everingham et al.*, 2014) drop below 40%.

Semantic segmentation benchmarks like Pascal VOC 2012 (*Everingham et al.*, 2014) or COCO stuff (*Caesar et al.*, 2018) typically assume that the semantic classes are mutually exclusive, non ambiguous and are not organized in a hierarchy. All of these assumptions are simplifications. In reality, a pixel can belong to multiple semantic classes at the same time (*e.g.* have multiple attributes), words can be ambiguous (*e.g.* the mouse can refer to the rodent as well as the device), exhibit a hierarchy (cat and Egyptian cat). Datasets reflecting the complex structure of the human mind are yet to be collected and annotated and corresponding models are yet to be trained.

Specifically for weakly supervised semantic segmentation, the main challenge is that some cues about the content are given but the precise spatial extent of the semantic classes is not provided. Us-

ing image class tags as supervision cues is a very popular approach. Generally, these methods learn what distinctive patterns of the image correlate with a certain tag. These regions are used as high confidence seeds which are then refined using color similarity, spatial proximity or objectness/saliency for guidance. These approaches work well on iconic images where the classes of interest are objects and a typical image shows only a very small number of classes, which is the case on Pascal VOC 2012 (*Everingham et al.*, 2014). However, *e.g.* for autonomous driving datasets, certain classes (like road or sky) appear in almost all images and for affordance datasets like the CAD 120 affordance dataset (*Sawatzky et al.*, 2017), the affordances of an object almost always appear together. In such setups, the relation between class tags and distinctive regions can not be established. Additionally, saliency or objectness are not reliable cues for highly cluttered datasets like CAD 120 affordance dataset (*Sawatzky et al.*, 2017). Finally, color similarity also has limitations when it comes to object part segmentation: The color of the fur of a cat is not related to its body parts. We address these weaknesses in the first three works of this thesis.

Clean image level supervision cues are cheaper than full annotations but not for free contrary to image tags or captions on the Internet. Using these imposes two challenges. First, the noise in the image tags and captions needs to be addressed. There is no guarantee that the image tag or caption describes the image content completely. And second, in case of captions, the challenge is to make use of the rest of the text apart from the class names itself, *e.g.* attributes of the objects or the description of their location.

In cases where only a tiny fraction of the images in the dataset are annotated at all, the usage of the unlabeled data gets into the focus. The challenge here is to come up with a supervision signal on the unlabeled data.

1.3 Contributions

This dissertation aims at reducing the performance gap of semantic segmentation models trained in a fully supervised setup compared to models trained with cheaper annotations. It explores several setups differing in data domain and type of supervision cue. The different supervision levels are shown in Figure 1.1. All works presented in this thesis constituted the state of the art at the moment of completion.

- In our first work presented in Chapter 4 we focus on weakly supervised object part affordance segmentation. Affordances of object parts differ in several aspects from objects or stuff. A technical difference is that the affordances of an object or its part need not be mutually exclusive, for instance the blade of a knife can be used to cut or to balance powder on it, so it is "cuttable" and "supportable" at the same time. Contrary, the semantic classes on popular benchmarks are mutually exclusive. Another difference is that affordances are more abstract than object classes. This leads to a high intra-class variation in appearance. For example, the interior of a mug and of a bowl can both be used to store liquids, however their visual appearance is very different.

  While benchmarks for semantic segmentation of full objects or stuff are publicly available, a dataset containing realistic images with object part affordances annotated on pixel level was still missing in 2016. This is why we introduced the CAD 120 affordance dataset. It comprises over 3000 images with over 9000 object instances and each object can have multiple affor-

dances. An important feature of the dataset is the object split: It ensures that object classes used in the train set do not occur in the test set. By evaluating on the object split, one can test the ability of a method to recognize the functionality of unseen object classes.

We provide a baseline for learning affordances in a fully supervised setup. Additionally, we introduce a method to learn affordances from sparse keypoints only. It is the first method for weakly supervised affordance segmentation and it outperforms methods designed for weakly supervised semantic segmentation of object classes.

- Chapter 5 is an extension of the first work. This time, we entirely focus on weakly supervised affordance segmentation. We provide an algorithm which beats the previous by a huge margin becoming the state-of-the-art. Additionally, it has less components and a significantly lower training time. In general, the hyper parameters of a weakly supervised learning algorithm can not be chosen by cross validation on a validation set, since by definition, ground truth data is not available. However, our algorithm allows for a stochastic approximation of the Jaccard index, which can be used for cross validation. This way, we can automatically tailor the hyper parameters to the dataset.

- Chapter 6 constitutes our last contribution to weakly supervised affordance segmentation. While in the previous works we eliminated human effort by reducing the annotation cost per image, here we cut the number of annotated images. While pixel-wise affordance labels are scarce, there exist large scale datasets with bounding boxes given for household objects. We suggest to annotate the affordances for a tiny number of object instances. Then, we use the bounding boxes and a modification of an unsupervised semantic alignment method to transfer these annotations to all other object instances of the same class in the train set. This makes the training of a fully supervised model possible. Qualitatively, this approach allows to learn the correct assignment of affordances within the object. This advantage allows it to become the state of the art on the IIT-AFF dataset *Nguyen et al.* (2017b).

- In Chapter 7 we continue with exploiting image captions as a supervision signal for weakly supervised semantic segmentation. Using curated image tags indicating the presence of certain object classes in an image is a very popular supervision setup in weakly supervised semantic segmentation of objects. The majority of the current state-of-the-art methods first obtain rough localization cues from these tags and uses these cues for training. These image tags are assumed to be cleaned by human annotators. On the Internet however, image captions describing an image but potentially not mentioning all object classes are abundant. We propose to use the rich object context provided in an image caption as an additional supervision cue. For example, the color of a "black cat" might simplify its location in an image.

  Being more concrete, we train a multi-modal model to locate arbitrary text snippets inside an image. After that, we parse the captions of images for objects and attributes describing them and locate these in the training images. These localization cues are superior to localization cues from clean image tags. This fact allows us to outperform the previous state-of-the-art on the COCO dataset *Lin et al.* (2014).

- Finally Chapter 8 examines a semi-supervised setup where pixel-wise labels are given for a small fraction of the data and no supervision cues of any type are given for the remaining

images. The general idea is to introduce an auxiliary task which is related to semantic segmentation but simpler than it. Since the task is simpler, the annotated data might suffice to learn it reasonably well. The predictions for the simpler task then in turn serve as a supervision signal on the unlabeled data. One natural choice in case of semantic segmentation would be to learn the supercategories of the classes of interest. However, we show that one can do substantially better by letting the system discover latent supercategories. These latent classes are chosen so that the latent class of a pixel reduces the uncertainty about the semantic class as much as possible. The supervision signal from latent classes turns out to be complementary to signals from adversarial networks leading to an approach performing on par with the state-of-the-art on Pascal VOC 2012 *Everingham et al.* (2014) and being the new state-of-the-art on Cityscapes *Cordts et al.* (2016).

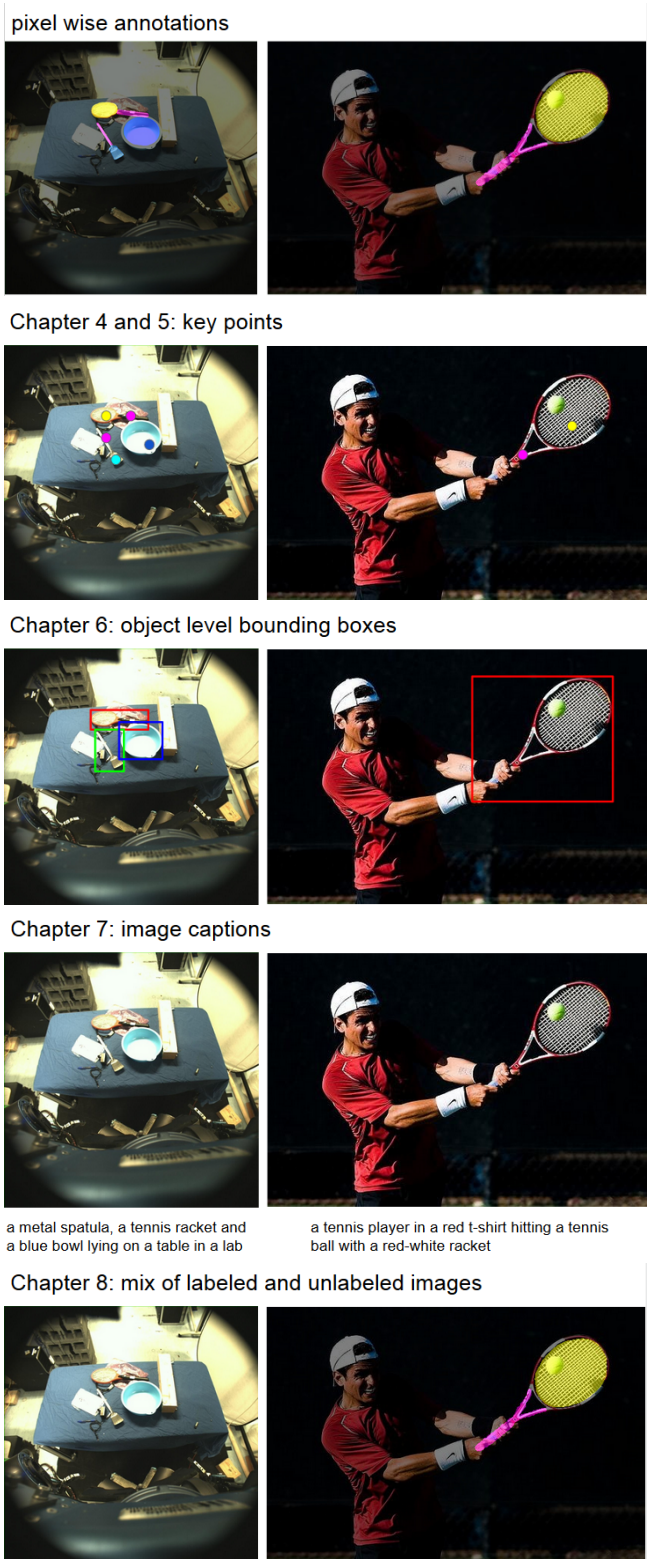


Figure 1.1: Providing the class label for each pixel on the train data allows to learn models with reasonable performance. However, these labels are extremely costly. This thesis examines how well we can do with cheaper annotation cues. Chapters 4 and 5 deal with affordance segmentation from keypoints, in Chapter 6 we use tight bounding boxes around objects for the same purpose. In Chapter 7 we examine image captions as a source of supervision. We conclude the thesis with a work on semi supervised semantic segmentation where we boost the performance by using images without any labels.

CHAPTER 2

Preliminaries

We will now discuss the machine learning techniques this thesis depends on. Convolutional neural networks and label refinement methods are, regardless of the supervision mode, the current workhorses of semantic segmentation. The most popular method for color and proximity based refinement is by modeling the pixel labels via conditional random fields (CRF). In the first work, we also use Grabcut (*Rother et al.*, 2004) It is a method for interactive foreground refinement which also heavily relies on color similarity and spatial proximity.

The task of weakly supervised semantic segmentation can be modeled as an Expectation Maximization problem. The supervision cues are the visible variables and the pixel-wise labels are the hidden ones. In the fourth work of this thesis, semantic information gathered from captions is used as a supervision cue. To convert human words into numerical data and preserve semantic proximity in the numerical space, we use the Word2Vec encoding (*Mikolov et al.*, 2013). Finally, the last work relies on conditional entropy as a measure of information gain and a discriminator network.

Contents

2.1 Convolutional Neural Network (CNN)

Convolutional neural networks are the paramount models when it comes to extracting semantic content from RGB images. These models are parametrized computational graphs. The nodes comprise data tensors, so called feature maps, and parametrized computational nodes, so called layers, mapping one or multiple input data tensors to one or multiple output data tensors. The layers typically

used in such networks are (dilated) convolutions, ReLUs, pooling, batch norm and deconvolution layers. We will describe each of them in more detail below.

In case of semantic segmentation, the CNNs take an image as input and return the pixel-wise label probability distributions as output. The first feature map corresponds to the RGB image and the last to the activations for the semantic classes. The higher the activation the more probable the class. If the classes are mutually exclusive, these activations f are converted to pixel-wise probability distributions P using the softmax:

$$P_{i,h,w,c}(\theta) = \frac{exp(-f_{i,h,w,c}(\theta))}{\sum_{\hat{c} \in C} exp(-f_{i,h,w,\hat{c}}(\theta))} \tag{2.1}$$

where i denotes the image, h and w are the spatial position of the pixel, c denotes the semantic class and θ are the parameters of the CNN.

If the labels are not mutually exclusive, a sigmoid function σ can map activations to probabilities:

$$P_{i,h,w,c}(\theta) = \sigma(f_{i,h,w,c}) = \frac{1}{1 + exp(-f_{i,h,w,c}(\theta))} \tag{2.2}$$

2.1.1 Convolutions

Convolutional layers map a single 3-dimensional tensor $T^{in} \in \mathbb{R}^{W_{in} \times H_{in} \times C_{in}}$ to another 3-dimensional tensor $T^{out} \in \mathbb{R}^{W_{out} \times H_{out} \times C_{out}}$. H and W are the spatial dimensions of the tensors (width and height). C is the channel dimension. The mapping is done using the kernel $K \in \mathbb{R}^{k \times k \times C_{in} \times C_{out}}$ and the output channel bias $b \in \mathbb{R}^{C_{out}}$

$$T^{out}_{h,w,c_{out}} = \sum_{c_{in}} \sum_{i,j=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} T^{in}_{h-i,w-j,c_{in}} K_{i,j,c_{in},c_{out}} + b_{c_{out}}. \tag{2.3}$$

The values of K and b are the parameters to optimize. Typically, k is a tiny odd integer, so the values of the output feature map only depend on a small spatial input, which is a reasonable feature for visual data. In 2.3, the resolution of the output tensor remains identical to the resolution of the input tensor. The number of channels in a tensor is typically very high, so in order to reduce the memory foot print, the tensors need to be down sampled in the spatial dimension. This is done using the stride s

$$T^{out}_{h,w,c_{out}} = \sum_{c_{in}} \sum_{i,j=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} T^{in}_{sh-i,sw-j,c_{in}} K_{i,j,c_{in},c_{out}} + b_{c_{out}}. \tag{2.4}$$

Dilated convolutions use kernels with the same number of parameters but sub sample the spatial positions in the input tensor the kernel is applied to. They maintain a wide receptive field with a low number of parameters. The general formula for dilation d and stride s is therefore:

$$T^{out}_{h,w,c_{out}} = \sum_{c_{in}} \sum_{i,j=-k/2}^{k/2} T^{in}_{sh-di,sw-dj,c_{in}} K_{i,j,c_{in},c_{out}} + b_{c_{out}}. \tag{2.5}$$

2.1.2 ReLU

Neural nets can not consist of convolutions only, since each convolution can be represented by a matrix multiplication and therefore consecutive convolutions correspond to a single linear operation. To go beyond a single linear operation, convolutional layers are typically followed by non-linearities applied element-wise to the output tensor $T \in \mathbb{R}^{W_{out} \times H_{out} \times c_{out}}$. The by far most popular non-linearity for semantic segmentation is ReLU:

$$ReLU(f) = max(f, 0) \tag{2.6}$$

2.1.3 Pooling

In order to propagate salient features over a local neighborhood in the feature map without introducing new parameters to the model, max pooling

$$T_{max\_pool,h,w,c_{out}} = \max_{i,j \in [-k/2, k/2]} T_{sh+i,sw+j,c_{out}}. \tag{2.7}$$

or average pooling

$$T_{avg\_pool,h,w,c_{out}} = \frac{1}{k^2} \sum_{i,j=-k/2}^{k/2} T_{sh+i,sw+j,c_{out}}. \tag{2.8}$$

is typically used. As for the convolutional layer, k denotes the kernel size and s denotes the stride. The motivation to use pooling operations is two fold: Firstly, it serves as a regularizer requiring that small spatial variations shall not have a big effect on the output features. Second, if memory footprint is an issue, it is a parameter free way to downsample the feature map while preserving the most salient features.

2.1.4 Deconvolution

A deconvolution layer first upsamples the input feature map filling zeros between the actual values. Then a convolution operation is applied. The deconvolution layer is used to learn an upsampling operation for a feature map instead of applying data agnostic alternatives like bilinear upsampling. An alternative name for deconvolution is transposed convolution. The name stems from the fact that the matrix of the deconvolution has the same structure as the transposed matrix of the respective convolution (the convolution which inverts the upsampling operation).

2.1.5 Batch Norm

The batch norm layer first normalizes the values of features using the mean μ_c and standard deviation σ_c of these features in the channel c:

$$\hat{f}_{h,w,c} = \frac{f_{h,w,c} - \mu_c}{\sigma_c} \tag{2.9}$$

after this a linear mapping is applied to these features

$$y_{h,w,c} = \alpha_c \hat{f}_{h,w,c} + \beta_c \tag{2.10}$$

α_c and β_c are parameters of the model that are learned during training.

The batch norm layer was introduced by (*Ioffe and Szegedy*, 2015) and empirically turned out to be useful in practice. The author reckoned that this is due to the reduction of the internal covariate shift, *i.e.* the order of magnitude of the features remains constant during training. However, the usefulness of this property for model training was so far not rigorously mathematically proved.

2.1.6 Training

The parameters of the CNN θ are estimated by maximum likelihood maximization of the pixel-wise labels in the train set. This is equivalent to minimizing the cross entropy loss:

$$L_{ce} = -\sum_{i=0}^{N}\sum_{h=0}^{H}\sum_{w=0}^{W} \log\left(P_{i,h,w,c_{gt}(i,h,w)}(\theta)\right) \tag{2.11}$$

where i iterates over the training images and $c_{gt}(i,h,w)$ is the semantic label of the pixel located at h, w in image i.

In general there does not exist an analytical solution to this minimization problem nor does there exist a numerical algorithm which is guaranteed to converge to a global minimum. However, one can apply gradient descent to at least arrive at a local minimum:

$$\theta_{t+1} = \theta_t - \lambda \nabla_{\theta_t} L \tag{2.12}$$

where λ is a hyper parameter called learning rate.

One practical problem with this approach is that one can not load the complete training set into the memory. Accumulating the gradient over the whole data by iteratively computing the gradient for subsets of images is also prohibitively slow. Therefore, one uses minibatch stochastic gradient descent, obtaining the gradient in 2.12 from tiny subsets of training data. Stochastic refers to the fact that the subsets are sampled at random.

The search for the global minimum with stochastic gradient descent often ends up in a local minimum or saddle point. The usage of momentum (*Rumelhart et al.*, 1986) is a powerful heuristic to avoid this. Essentially, the gradient $\nabla_{\theta_{old}}$ in 2.12 is exchanged for its exponential moving average V_t

$$V_t = \beta \nabla_{\theta_{t-1}} L + (1-\beta) V_{t-1} \tag{2.13}$$

This allows to jump out of local minima or saddle points and follow larger scale slopes.

There exist a number of works which try to further improve upon this heuristic. For example, a widely used algorithm is ADAM (*Kingma and Ba*, 2015). It additionally calculates the exponential moving average of the variance of the gradient:

$$U_t = \beta_{var}\left(\nabla_{\theta_{t-1}} L\right)^2 + (1-\beta_{var}) U_{t-1} \tag{2.14}$$

The square is taken element-wise. Then, it updates the parameters by

$$\theta_{t+1} = \theta_t - \lambda \frac{V_t}{U_t + \varepsilon} \tag{2.15}$$

In case of a convex optimization problem, it can be mathematically proved that ADAM converges faster than SGD with momentum. However, since the optimization of deep convolutional neural networks is a non-convex problem, these methods have not eclipsed SGD with momentum in practice.

2.1.7 Overfitting

Reducing the loss on training data does not necessary mean that the model will generalize well on test data as well. This happens if the number of parameters in the model is too high for the given amount of data. There exist several strategies to alleviate this issue.

Data Augmentation The images as well as the corresponding segmentation maps are randomly geometrically transformed (cropped, scaled, flipped) before being fed into the CNN. This somehow increases the amount of available data (although the samples are correlated of course).

Dropout Layer A dropout layer multiplies the values in the input feature map with 0 at random. This can be seen as data augmentation on the feature map level, since it provides the subsequent layers with noisy data. The dropout layer applied to visual data was one of the major contributions in the AlexNet paper (*Krizhevsky et al.*, 2012).

Regularization Another approach is to add the L_2 or L_1 norm of the weights to the loss term or apply an exponential decay to the weights itself. In any case, the goal is to prevent large weight values. It can be even shown for convex problems that these two approaches yield equivalent results.

Pretraining One very powerful technique is to train a CNN for the same or closely related task on a far bigger dataset with similar images. Subsequently, one exchanges the few last layers for randomly initialized ones while keeping the learned parameters of the deeper layers. Then one trains the network on the actual dataset of interest. When doing this, the learning rate is typically far higher for the layers initialized from scratch than for the deeper ones. The later can be even set equal to 0. This process is called “fine-tuning”.

2.2 Conditional Random Field

Typically, the semantic labels of two spatially close pixels with the same color are the same. This information is not explicitly incorporated into CNNs for semantic segmentation. Therefore segmentation results are usually refined using conditional random fields. The work of (*Krahenbuhl and Koltun*, 2011) is very popular in practice and shall be described here.

In a conditional random field, the probability distribution of pixel-wise labels $\mathbf{c}$ given an image I is given by

$$P(\mathbf{c}|\mathbf{I}) = \frac{1}{Z(\mathbf{I})} \exp\Big(- \sum_{\kappa \in \kappa_{\mathcal{G}}} \Phi_\kappa(\mathbf{c}|\mathbf{I}) \Big) \tag{2.16}$$

where $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is a graph on $\mathbf{c}$, κ is a clique in the set of cliques $\kappa_{\mathcal{G}}$, Φ_c is a potential belonging to that clique, $Z(\mathbf{I})$ is a normalizing factor. In the case of (*Krahenbuhl and Koltun*, 2011), the graph is supposed to be fully connected, so there exists only one clique and the probability distribution is

given by

$$P(\mathbf{c}|\mathbf{I}) = \frac{1}{Z(\mathbf{I})} \exp\Big(- \Phi(\mathbf{c}|\mathbf{I})\Big) \tag{2.17}$$

The overall potential comprises unary potentials Ψ_u and binary potentials Ψ_p defined at each pixel

$$\Phi(\mathbf{c}|\mathbf{I}) = \sum_{i}^{N} \Psi_u(c_i) + \sum_{i<j} \Psi_p(c_i, c_j) \tag{2.18}$$

The unary potential represents the relation between pixels and local features, it must be a monotonically decreasing function of the probability predicted by the CNN for this pixel. The binary potential reflects that pixels close in geometrical or color space are likely to have the same label. therefore, they are set to

$$\Psi_p(c_i, c_j) = \mu(c_i, c_j) \left(w^1 \exp\Big(- \frac{|x_i - x_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\Big) + w^2 \exp\Big(- \frac{|x_i - x_j|^2}{2\theta_\gamma^2}\Big)\right) \tag{2.19}$$

where $\mu(c_i, c_j) = [c_i \neq c_j]$, I_i, I_j denote the color vectors, x_i, x_j are the spatial coordinates of pixel i and j and θ_α and θ_β θ_γ are hyper parameters.

It is intractable to compute the probability distribution $P(\mathbf{c})$ itself. Instead, the authors look for a distribution which can be expressed as a product of its marginals $\mathbf{Q} = \prod_i Q_i$ and then search for parameters of Q which minimize the KL-divergence between Q and P. This optimization is an iterative procedure which involves convolutions with a Gaussian kernel in a feature space of spatial positions and colors. One major contribution of this paper was to perform these convolutions in a timely efficient manner.

Once $\mathbf{Q}$ is estimated, the refined pixel-wise labels are obtained by taking the argmax over Q_i for pixel i.

2.3 Grabcut

Originally, Grabcut (*Rother et al.*, 2004) was an algorithm for interactive foreground vs background segmentation. As a minimum requirement, the user has to provide example pixels of the background and optionally example pixels of foreground. These pixels are used as a seed to segment the image into a foreground and background based on spatial proximity and color similarity. Optionally, the user can correct the segmentations interactively. In the original paper, the example pixels for background were those on the edges of a loose box separating the foreground from the background. In our first work, we estimate high confidence foreground regions and background regions with a neural network and initialize Grabcut from them.

Being more specific, Grabcut models the color distribution of foreground and background pixels with 2 separate Gaussian Mixture Models with $\mathbf{k}$ components each. I denotes the color of pixel n $\alpha_n \in \{0, 1\}$ its label and $k_n \in \{1, ...K\}$ the component it belongs to. $\pi(\alpha_n, k_n)$ denotes the weight of Gaussian component k_n of the foreground model ($\alpha_n = 1$) or the background model ($\alpha_n = 0$). $\mu(\alpha_n, k_n)$ and $\Sigma(\alpha_n, k_n)$ are the mean and variance of the respective component. θ summarizes the

weights, means and variances of the components. The energy E is a function of the model parameters as well as the labeling:

$$E(\alpha, \mathbf{k}, \theta, \mathbf{I}) = \sum_n D(\alpha_n, k_n, \theta, I_n) + V(\alpha, \mathbf{I}) \tag{2.20}$$

where $D(\alpha_n, k_n, \theta, I_n) = -\log p(I_n|\alpha_n, k_n, \theta) - \log \pi(\alpha_n, k_n)$ is the negative log-likelihood of the color value given the model and V denotes the graph energy:

$$V(\alpha, \mathbf{I}) = \gamma \sum_{(m,n)\in\mathbf{C}} [\alpha_n \neq \alpha_m] exp\left(-\beta \left\|I_m - I_n\right\|^2\right) \tag{2.21}$$

where $\mathbf{C}$ denotes the neighborhood of the pixel, and γ and β are hyper parameters.

After a random initialization of the background GMM with the pixels marked as background by the user (set T_B) and the foreground GMM using the remaining pixels (set T_U), one minimizes the energy by two alternating steps. One is the estimation of the labeling α minimizing the energy while keeping the GMM parameters frozen. This is done with the mincut algorithm. The pixels are treated as the graph nodes and the penalty terms in 2.21 as the edge weights. The other is the estimation of the parameters of the GMM which maximize the likelihood of the labeling. This essentially amounts to assigning the pixels labeled as background to the Gaussian components, estimating the weights as the share of the background pixels assigned to the component and calculating the mean and variance of the pixels inside the component. The same procedure is applied to the foreground.

2.4 Expectation Maximization

The training of a model parametrized by θ using the cross entropy loss in a fully supervised setup is equivalent to maximizing the log likelihood $L(\theta) = \sum_{i=0}^{N} \log P(\mathbf{C_i}, \mathbf{I}_i|\theta)$ where $\mathbf{C_i}$ is the labeling of image $\mathbf{I}_i$. In a weakly supervised setup, $\mathbf{C_i}$ is unknown, *i.e.* a hidden random variable while the supervision cues are the visible variables Z_i in image $\mathbf{I}_i$. Now the goal is to maximize the log-likelihood $L(\theta) = \sum_{i=0}^{N} \log P(Z_i, \mathbf{I}_i|\theta)$. This can be done using the Expectation-Maximization (EM) algorithm (*Dempster et al.*, 1977). To do this, in iteration t, in the Expectation step, one calculates the function

$$Q(\theta|\theta_t) = \sum_{\mathbf{C}} P(\mathbf{C}|\mathbf{I}, Z, \theta_t) \log(P(\mathbf{C}, \mathbf{I}, Z, |\theta)) \tag{2.22}$$

and maximizes it in the Maximization step with respect to θ, with θ_t being the estimate of the model parameters from the previous iteration.

One can show that in each step, L increases by at least the amount that Q increases:

$$\begin{aligned}
&\log P(\mathbf{I}, Z|\theta_{t+1}) - \log P(\mathbf{I}, Z|\theta_t) = \sum_{\mathbf{C}} P(\mathbf{C}|\mathbf{I}, Z, \theta_t)(\log P(\mathbf{I}, Z|\theta_{t+1}) - \log P(\mathbf{I}, Z|\theta_t)) \\
&= \sum_{\mathbf{C}} P(\mathbf{C}|\mathbf{I}, Z, \theta_t)(\log P(\mathbf{C}, \mathbf{I}, Z|\theta_{t+1}) - \log P(\mathbf{C}|\mathbf{I}, Z, \theta_{t+1})) \\
&- \sum_{\mathbf{C}} P(\mathbf{C}|\mathbf{I}, Z, \theta_t)(\log P(\mathbf{C}, \mathbf{I}, Z|\theta_t) - \log P(\mathbf{C}|\mathbf{I}, Z, \theta_t)) \\
&= Q(\theta_{t+1}|\theta_t) - Q(\theta_t|\theta_t) - \sum_{\mathbf{C}} P(\mathbf{C}|\mathbf{I}, Z, \theta_t)(\log P(\mathbf{C}|\mathbf{I}, Z, \theta_{t+1}) - \log P(\mathbf{C}|\mathbf{I}, Z, \theta_t)) \\
&= (Q(\theta_{t+1}|\theta_t) - Q(\theta_t|\theta_t)) + KL(P(\mathbf{C}|\mathbf{I}, Z, \theta_t)||P(\mathbf{C}|\mathbf{I}, Z, \theta_{t+1}))
\end{aligned} \tag{2.23}$$

Since the Kullback-Leibler-divergence $KL(P(\mathbf{C}|\mathbf{I}, Z, \theta_t)||P(\mathbf{C}|\mathbf{I}, Z, \theta_{(t+1)}))$ is always positive, the increase in the log-likelihood is indeed at least as high as the increase in Q. Since the log-likelihood monotonically increases and has an upper bound 0, the value of Q will converge. If additionally the sequence of θ_t is chosen so that

$$\frac{\partial}{\partial \theta_{t+1}} Q(\theta_{t+1}|\theta_t) = 0 \tag{2.24}$$

for all t and $\frac{\partial^2}{\partial^2 \theta} Q(\theta|\theta_t)$ is negative definite and bounded away from 0 for all θ lying on the path between consecutive θ_t, the algorithm will converge to a θ^* where $\frac{\partial}{\partial \theta^*} L(\theta^*) = 0$. So if the function Q is concave, the solution will converge to a local maximum or saddle point of the log-likelihood (minimum is excluded due to the monotonic increase).

2.5 Conditional Entropy

The entropy of the distribution of a random variable X is defined as

$$H(X) = \sum_x p(x) \log p(x) \tag{2.25}$$

From an information theory perspective, it measures the uncertainty about the random variable X. In his seminal paper, (*Shannon*, 1948) proves that the entropy definition above is the only function (up to a scaling factor) which satisfies the properties necessary for an information measure $F(p_1, p_2, ...p_n)$

- F should be a continuous function
- For uniformly distributed random variables, F should grow monotonically with the number of outcomes
- If a random experiment is broken down into 2 steps, the measure of information shall not change: $F(p_1, p_2, ...p_n) = F_{first\_step} + E_{first\_step}(F_{second\_step})$

For two random variables X and Y, the conditional entropy $H(X|Y)$ is the expectation value of

the entropy of the conditional distribution $P(X|Y)$:

$$\begin{aligned} H(X|Y) &= E_Y \left(\sum_x p(x|y) \log p(x|y) \right) \\ &= \sum_y \sum_x p(y)p(x|y) \log p(x|y) \\ &= \sum_y \sum_x p(x,y) \log p(x|y) \end{aligned} \tag{2.26}$$

The closer the relationship between the X and Y, the smaller $H(X|Y)$. If Y determines X, $H(X|Y) = 0$. Conversely, if they are independent random variables $H(X|Y) = H(X)$.

In the last work of this thesis we let our model automatically discover latent classes which reduce the uncertainty about the semantic class of a pixel as much as possible. Therefore, we use the conditional entropy of the semantic classes given the latent class as our loss function.

2.6 Word2Vec

In 7, we use image captions as supervision cues. To convert arbitrary words to fixed size semantic embedding vectors, we use the Word2Vec model proposed in (*Mikolov et al.*, 2013).

The model itself is a shallow neural net consisting of 3 layers. The first 2 layers are linear layers and the last one is a softmax layer. For a corpus containing V words, the input to the network are one-hot vectors $v \in \mathbb{R}^V$. An input vector is mapped to a semantic embedding of dimension $e \in \mathbb{R}^D$, via $e = W_1 v$ where D is a hyper parameter and $W_1 \in \mathbb{R}^{D \times V}$ are the weights of the first layer. Then, the second layer maps the embedding to the activation vector $a = W_2 e$, $W_2 \in \mathbb{R}^{V \times D}$. The final layer takes the softmax of a estimating the probability for each word from the corpus to occur in the vicinity of the input word.

To train this model, the authors use snippets of sentences of length 10. For each sentence, they form all pairs of words occurring in the snippet. For each pair, one of the words is the input and the one hot encoding of the other the ground truth. Words typically occurring in the same context are mapped to similar semantic embeddings while words occurring in different contexts exhibit different embeddings. Furthermore, the semantic embedding of a particular word is just a column of W_1, since the input to the neural network are one hot vectors essentially selecting a column of W_1.

CHAPTER 3

Related Work

Now we review the work related to this thesis. We start with research on affordances which are our target domain in the first half of the thesis. From there we move on to fully supervised semantic segmentation. After this, we cover weakly supervised semantic segmentation and weakly supervised visual grounding, a task we borrow ideas from. Finally we review the smaller area of semi supervised semantic segmentation. Curriculum learning is of interest, since our method for semi supervised semantic segmentation is related to it.

Contents

3.1 Affordances

3.1.1 Attributes of Objects

Properties of objects can be described at various levels of abstraction by a variety of attributes including visual properties (*Parikh and Grauman*, 2011; *Khan et al.*, 2012; *Farhadi et al.*, 2009; *Lampert et al.*, 2009), *e.g.* object color, shape and object parts, physical properties (*Ferrari and Zisserman*, 2007; *Zhu et al.*, 2014b), *e.g.* weight, size and material characteristics, and categorical properties (*Akata et al.*, 2015; *Deng et al.*, 2012). Object affordances, which describe potential uses of an object, can also be considered as other attributes. For instance, (*Chao et al.*, 2015) describes affordances by object-action pairs whose plausibility is determined either by mining word co-occurrences in textual data or by measuring visual consistency in images returned by an image search. (*Zhu et al.*, 2014b) propose to represent objects in a densely connected graph structure. While a node represents one of the various visual, categorical, physical or functional aspects of the object, an edge indicates

the plausibility of both node entities to occur jointly. Upon querying the graph with observed information, *e.g. {round, red}*, the result is a set of most likely nodes, *e.g. {tomato, edible, 10-100gm, pizza}*.

3.1.2 Affordances as Auxiliary Representation

Affordances have also been used as an intermediate representation for higher level tasks. In (*Castellini et al.*, 2011), object functionality is defined in terms of the poses of relevant hand-grasps during interaction. Object recognition is performed by combining individual classifiers based on object appearance and hand pose. (*Zhu et al.*, 2015) use affordances as a part of a task oriented object modeling. They formulate a generative framework that encapsulates the underlying physics, functions and causality of objects being used as tools. Their representation combines extrinsic factors that include human pose sequences and physical forces such as velocity and pressure and intrinsic factors that represent object part affordances. (*Koppula and Saxena*, 2016) models action segments using CRFs which are described by human pose, object affordance and their appearances. Using a particle filter framework, future actions are anticipated by sampling from a pool of possible CRFs thereby performing a temporal segmentation of action labels and object affordances. (*Kjellström et al.*, 2011) jointly models object appearance and hand pose during interactions. They demonstrate simultaneous hand action localization and object detection by implicitly modeling affordances. (*Fowler et al.*, 2018) predict affordances of objects in the scene from human activity observed with a 360 degree camera. These affordances then improve 3D scene completion.

3.1.3 Localizing Affordances

Localizing object affordances based on supervised learning has been addressed in particular in the context of robotics applications by classical hand crafted approaches. (*Katz et al.*, 2014) performs robotic manipulations on objects based on affordances which are inferred from the orientations of object surfaces. (*Kim and Sukhatme*, 2014) learn a discriminative model to perform affordance segmentation of point clouds based on surface geometry. (*Myers et al.*, 2015) use RGB-D data to learn pixel-wise labeling of affordances for common household objects. They explore two different features: one based on a hierarchical matching pursuit and another based on normal and curvature features derived from RGB-D data. (*Hermans et al.*, 2011) learn to infer object level affordance labels based on attributes derived from appearance features. In (*Desai and Ramanan*, 2013), pixel-wise affordance labels of objects are obtained by warping the query image to the K-nearest training images based on part locations inferred using deformable part models. (*Song et al.*, 2016) combine top-down object pose based affordance labels with those obtained from bottom-up appearance based features to infer part-based object affordances. Top-down approaches for affordance labeling have been explored in (*Grabner et al.*, 2011; *Jiang et al.*, 2013b) where scene labeling is performed by observing possible interactions between scene geometry and hallucinated human poses. Localizing object affordances based on human context has been also studied in (*Koppula and Saxena*, 2014). In a more recent work, (*Siam et al.*, 2019) model affordances of unseen objects in a teacher-student setup and learn affordances from instructional videos.

Meanwhile, the field of affordance localization was conquered by CNNs. Since grasp locations are of special interest, (*Lenz et al.*, 2015) propose a two stage cascade approach based on RGB-D data to regress potential grasp locations of objects. (*Kumra and Kanan*, 2017) combine a deep

neural net to extract features and a shallow one to obtain the grasp configuration. (*Akizuki and Aoki*, 2018) log human tactile interaction with objects to infer optimal grasping locations from it. (*Nguyen et al.*, 2016) generalized this approach to multiple affordances. (*Roy and Todorovic*, 2016) use CNNs to estimate a depth map and surface normals for a scene and a single-label CNN for semantic segmentation. The feature maps are then merged to predict affordances maps. Localizing objects and affordances jointly turned out to be a powerful approach. (*Nguyen et al.*, 2017b) used two separate nets to first locate the objects and subsequently localize them. (*He et al.*, 2017) propose a region alignment layer to align the input image space with the feature map space. It is an end to end approach which performs object detection and instance wise semantic segmentation. (*Do et al.*, 2018) modified this seminal architecture to detect multiple affordance classes in the object, instead of binary classes as in (*He et al.*, 2017). (*Chu et al.*, 2019) detect and rank affordances simultaneously. (*Lüddecke et al.*, 2019) localize multiple affordances per object.

Various image domains have been explored for affordance detection or segmentation. The context of affordances also strongly differs depending on the task such as understanding human body parts (*Lin et al.*, 2017), classifying environment affordances (*Roy and Todorovic*, 2016; *Pham et al.*, 2018), or detecting affordances of real world objects that robots interact with (*Myers et al.*, 2015). (*Schoeler and Wörgötter*, 2016) use predefined primary tools to infer object functionalities from 3D point clouds. (*Kjellström et al.*, 2011) detect object affordances by observing object-action interactions performed by humans. Similarly, in their more recent work (*Fang et al.*, 2018) propose to learn to detect affordances from demo videos. Recently, (*Li et al.*, 2019c) introduced a synthetic 3D scene dataset with human poses. Affordances for human poses are predicted from geometry.

3.1.4 RGB Image Datasets with Pixel Level Affordance Annotation

Before the CAD 120 affordance dataset was intorduced in this work, there already existed the UMD affordance dataset introduced by (*Myers et al.*, 2015). It showed single isolated objects on a table colored in a distinctive blue color. The affordances were not mutually exclusive, but ranked by their plausibility for each object part. It was an important pioneering step, but the images were not sufficiently realistic. After our CAD 120 affordance dataset was published, (*Nguyen et al.*, 2017b) created the IIT 2017 affordance dataset. It comprises several thousand images taken from the ImageNet challenge (*Russakovsky et al.*, 2015) as well as images taken in the lab by the team. Contrary to our dataset, the affordances are selected so that they are mutually exclusive.

3.2 Semantic Segmentation

Since this field is huge, only the most influential and recent works are presented in this section. Current semantic segmentation methods use fully convolutional neural nets. (*Shelhamer et al.*, 2015) popularized this approach. In their seminal work (*Chen et al.*, 2015) introduce atrous convolutions which increased the spatial dimensions of a convolutional kernel without increasing the number of parameters in it. Another early work experimenting with atrous convolutions is (*Yu and Koltun*, 2016). The atrous convolution was extended to atrous spatial pooling pyramids in the follow up work (*Chen et al.*, 2018b). In the last work from this series (*Chen et al.*, 2017), the atrous convolution blocks were augmented with globally pooled features and batch norm layers between subsequent atrous convolution layers. The sparse sampling of information by atrous convolutions has been iden-

tified as a weakness in (*Wang et al.*, 2018a). Consequently, the authors introduce policies for dilation rates to ensure full coverage of all pixels inside the image region which corresponds to a particular spatial location in the last feature map..

While atrous convolutions increase the field of view on features, another direction of research successfully incorporates class co-occurrence information into the final prediction. (*Zhao et al.*, 2017) aggregated class context information with a spatial pyramid pooling module, thereby providing a widely used architecture. Recently, (*He et al.*, 2019b) extended this approach to an adaptive context module. While in (*Zhao et al.*, 2017) the pooling assigns equal weight to all spatial positions, the recent work (*He et al.*, 2019b) introduces the adaptive context module to make the pooling weight depending on local features as well as global context. (*Ding et al.*, 2019b) predict labels from features of multiple levels and use class information from higher level features as context to suppress wrong predictions from low level features. Another approach is to enforce class co-occurrence and shape structure in the output. It is common practice to apply the architecture agnostic conditional random field module proposed by (*Krahenbuhl and Koltun*, 2011) on top of the predictions. It penalizes segmentations which assign different semantic classes to pixels which are close spatially and in color space. (*Zheng et al.*, 2015) noted that the CRF module can be formulated as an RNN which allowed to learn the CRF parameters like all the other parameters in the network. In these works, the measure for pixel similarity is rather simplistic and handcrafted. Furthermore, the pairwise potential is assumed to be the sum of products of a pixel similarity term and a class compatibility term. A number of works (*Liu et al.*, 2015; *Lin et al.*, 2016b; *Vemulapalli et al.*, 2016) predict the pairwise potentials of pixels with deep neural nets. Another feature of these approaches is that class compatibility now depends on the relative spatial position of the respective pixels. (*Ke et al.*, 2018) enforced low KL distance for neighbour pixels belonging to the same class and high KL distance if the classes are different. The range of the neighborhood is adjusted to individual classes in a min max game. In their very recent work, (*Hwang et al.*, 2019) propose to enhance the structure differences between the ground truth output and the predictions with an analyzer network. The segmentation and the analyzer networks then play a min max game. (*Zhou et al.*, 2019) formulate the incorporation of context as a reinforcement learning problem. The quality of the segmentation maps is the state which defines the reward, while the context module is the player seeking to improve the segmentation.

The higher and therefore more abstract feature layers of a segmentation convnet have typically a smaller spatial resolution then the lower ones. Before atrous convolutions, this was to increase the image region connected to a pixel in the feature map. After their introduction this is still done to reduce the memory footprint of the feature maps. If the down scaling is achieved by pooling operations, it also has the advantage of a better object recognition performance. There exists a track of research which seeks to combine the semantically rich low resolution high level features with spatially precise low level features. Some works introduce new modules which directly combine low level feature maps with high level feature maps (*Lin et al.*, 2017; *Ronneberger et al.*, 2015; *Pohlen et al.*, 2017; *Chen et al.*, 2018c). (*Pohlen et al.*, 2017) for example keep a residual stream at full resolution and a pooling stream. Both streams have an equal number of feature maps, and the feature maps of one level use the feature maps from the previous level from both streams for an update. The residual stream preserves the low level details while the pooling stream improves object recognition. (*Badrinarayanan et al.*, 2017) propose to reconstruct the spatial resolution by storing the argmax index in the max pooling operations and use it to enrich higher level maps with spatial information.

Others apply deconvolution modules to the high level feature maps to reconstruct the fine grained

details (*Noh et al.*, 2015; *Wang et al.*, 2018a; *Tian et al.*, 2019). Recently, the attention module proposed in (*Vaswani et al.*, 2017) inspired attention based semantic segmentation (*Yu et al.*, 2018; *Zhao et al.*, 2018c; *Zhang et al.*, 2018a) methods. For example (*Yu et al.*, 2018; *Zhang et al.*, 2018a) introduce a module to suppress feature channels depending on global context. (*Zhao et al.*, 2018c) use attention to decide which spatial positions in a feature map are decisive for the next feature map. The recent works of (*Fu et al.*, 2019) use features as key and query values to aggregate transformed feature information. Although not explicitly stated (*Ding et al.*, 2019b) use a self-attention like approach to mask their convolution kernels locally conditioned on local features. (*Mou et al.*, 2019) use a query and key like mechanism to estimate cross channel and cross pixel relations in the feature maps and augment the feature map with this cross channel and cross pixel information. This direction of research continues to thrive in the very recent works of (*Zhu et al.*, 2019b; *Huang et al.*, 2019; *Zhang et al.*, 2019a; *Li et al.*, 2019b). Another branch of research tackles the task of time critical semantic segmentation (*Paszke et al.*, 2016; *Zhao et al.*, 2018b; *Zhuang et al.*, 2019) with (*Zhuang et al.*, 2019; *He et al.*, 2019c; *Chen et al.*, 2019b; *Li et al.*, 2019a; *Orsic et al.*, 2019; *Wang et al.*, 2019; *Marin et al.*, 2019a) being the most recent approaches. Examples of current research also include automatic search for an architecture (*Liu et al.*, 2019a; *Nekrasov et al.*, 2019; *Zhang et al.*, 2019b), domain adaptation methods (*Chang et al.*, 2019; *Vu et al.*, 2019; *Sun et al.*, 2019; *Li et al.*, 2019d; *Larsson et al.*, 2019; *Du et al.*, 2019; *Chen et al.*, 2019a; *Wu et al.*, 2019; *Lian et al.*, 2019; *Luo et al.*, 2019; *Choi et al.*, 2019), knowledge distillation (*Liu et al.*, 2019b; *He et al.*, 2019c) advanced semantic data augmentation methods (*Shetty et al.*, 2019), works focusing on the segmentation loss (*Marin et al.*, 2019b), data dependent upsample methods (*Tian et al.*, 2019), automatically learned pooling strategies (*Wei et al.*, 2019), adaptive filter sizes (*He et al.*, 2019a) more sophisticated strategies to fuse low and high level features (*Pang et al.*, 2019) or exploitation of object boundaries (*Ding et al.*, 2019a).

3.3 Weakly Supervised Semantic Segmentation and Visual Grounding

3.3.1 Approaches Based on Hand Crafted Features

Weakly-supervised learning for semantic image segmentation has been investigated in several works. In this context, training images are only annotated at the image-level and not at pixel-level. For instance, (*Vezhnevets and Buhmann*, 2010) formulates the weakly supervised segmentation task as a multiple instance and multitask learning problem. Further, (*Vezhnevets et al.*, 2011, 2012) incorporate latent correlations among superpixels that share the same labels but originate from different images. (*Xu et al.*, 2014) simplify the above formulation by a graphical model that simultaneously encodes semantic labels of superpixels and presence or absence of labels in images. (*Zhang et al.*, 2015) handle noisy labels from social images by using robust mid-level representations derived through topic modeling in a CRF framework.

3.3.2 Deep Learing Based Approaches

A natural step to less supervision are more coarse spatial cues like bounding boxes, keypoints and scribbles. (*Papandreou et al.*, 2015) use the expectation-maximization algorithm to perform weakly-supervised semantic segmentation based on annotated bounding boxes and image-level labels. An-

other more sophisticated approach based on bounding boxes was proposed in (*Khoreva et al.*, 2017). The authors use region proposal generated by Multiscale Combinatorial Grouping (MCG) (*Pont-Tuset et al.*, 2017) and Grabcut (*Rother et al.*, 2004) to localize the objects more precisely within the bounding box. (*Li et al.*, 2018b) used bounding boxes for object classes and image level supervision for stuff classes. (*Song et al.*, 2019) choose a similar path for object classes and design losses tailored to region refinement within the bounding box. (*Lin et al.*, 2016a) made use of a region-based graphical model, with scribbles providing ground-truth annotations to train the segmentation network. Scribbles also served as supervision for the works of (*Tang et al.*, 2018a,b), which investigate loss regularizations. Human annotated keypoints were used by (*Bearman et al.*, 2016) for weakly supervised class segmentation.

While the works mentioned above require some type of explicit spatial hints, others only rely on the list of present classes in the image. (*Qi et al.*, 2016) used proposals generated by (MCG) (*Pont-Tuset et al.*, 2017) to localize semantically meaningful objects. Recently, (*Fan et al.*, 2018) leveraged saliency to obtain object proposals and link objects of the same class across images with a graph partitioning approach. (*Pathak et al.*, 2015) addressed the weakly-supervised semantic segmentation problem by introducing a series of constraints.

In absence of explicit location cues provided by humans, class activation maps (CAMs) (*Zhou et al.*, 2016) have become a seminal supervision source. (*Pinheiro and Collobert*, 2015; *Shimoda and Yanai*, 2016) pioneered in this area. In (*Kolesnikov and Lampert*, 2016), three loss functions are designed to gradually expand the high confidence areas of CAMs. This approach was first improved by (*Wei et al.*, 2017a) who use an adversarial erasing scheme to acquire more meaningful regions that provide more accurate heuristic cues for training. Recently, (*Huang et al.*, 2018) proposed deep seeded region growing of CAMs with image level supervision. (*Wei et al.*, 2017b) presented a simple-to-complex framework which used saliency maps produced by the methods (*Cheng et al.*, 2015) and (*Jiang et al.*, 2013a) as initial guides. (*Hou et al.*, 2018b) advanced this approach by combining the saliency maps (*Hou et al.*, 2018a) with attention maps (*Zhang et al.*, 2016). (*Oh et al.*, 2017) and (*Chaudhry et al.*, 2017) considered linking saliency and attention cues together, but they adopted different strategies to acquire semantic objects. (*Zeng et al.*, 2019) jointly learn weakly supervised semantic segmentation and fully supervised saliency prediction. (*Roy and Todorovic*, 2017) leveraged both bottom-up and top-down attention cues and fused them via a conditional random field as a recurrent network. (*Ahn and Kwak*, 2018) use image level class labels to generate an initial set of CAMs and then propagate those CAMs by using random walk predictions from AffinityNet. (*Wei et al.*, 2018) use image level supervision with dilated convolutions with varying levels of dilations to generate weakly supervised segmentations. (*Wang et al.*, 2018b) use image level supervision along with a bottom-up and top-down framework, which alternatively expands object regions and optimizes segmentation network. (*Briq et al.*, 2018) use a convolutional simplex projection network for weakly supervised image segmentation. (*Tang et al.*, 2018b) integrate standard regularizers directly into the loss functions over partial input for semantic segmentation. (*Ge et al.*, 2018) use a four stage process that combines object localization with filtering and fusion of object categories. (*Hong et al.*, 2017) and (*Jin et al.*, 2017) tackle the weakly-supervised semantic segmentation problem using images or videos from the Internet. Some works focus on improving the class activation map itself. (*Li et al.*, 2018a) make the CAM an explicit component during training and use self guidance from the network. (*Lee et al.*, 2019) introduce a dedicated dropout layer to improve class activation maps. They use them to locate the less discriminative object parts. Many of these works iteratively refine

the predictions of the network on the data using conditional random fields. (*Shimoda and Yanai*, 2019) learn to predict the CRF refinement to learn the underlying logic of color similarity guided label propagation.

While these works address object segmentation, a few works focus on weakly supervised semantic part detection (*Krause et al.*, 2015) or segmentation (*Meng et al.*, 2017). It is even more challenging to tackle instance wise weakly supervised semantic segmentation. While (*Khoreva et al.*, 2017) pioneered in this area using bounding boxes, current state-of-the-art works (*Ahn et al.*, 2019; *Zhu et al.*, 2019a) rely on image level cues only and use the CAM paradigm. (*Shen et al.*, 2019) demonstrate that learning weakly supervised object detection and weakly supervised instance segmentation jointly is beneficial for both tasks.

3.3.3 Weakly Supervised Visual Grounding

The task of weakly supervised visual grounding is related but a different task. Instead of training a network for semantic image segmentation, the goal is to localize a given phrase in an image and the challenge is to handle phrases that are not part of the training data. This means that they require a phrase for inference, while in our case the captions are only given for training but not for inference on the test dataset. The output format of these methods varies. Some output bounding boxes (*Chen et al.*, 2018a; *Bouritsas et al.*, 2018; *Zhao et al.*, 2018a; *Akbari et al.*, 2019; *Datta et al.*, 2019), while others return heat maps for phrases from which a localization point can be derived (*Engilberge et al.*, 2018; *Durand et al.*, 2017; *Xiao et al.*, 2017).

3.4 Semi-Supervised Semantic Segmentation

Several of the works above can handle a setup where pixel-wise labels are given for a fraction of the data, while for the bigger part of the data only sparse cues are available. (*Kalluri et al.*, 2019) consider a semi supervised domain adaptation setup with a labeled source domain and an unlabeled target domain. However, the setting without any supervision cues on unlabeled images was so far only addressed by (*Hung et al.*, 2018) and (*Mittal et al.*, 2019). The former use a discriminator network to supervise the predictions on unlabeled data. Additionally, the discriminator indicates the regions with high segmentation confidence, which the authors then incorporate into their custom loss. The later improve upon the first using a better discriminator network. Additionally, the authors propose to train an image level classifier to suppress false positives on unlabeled data.

CHAPTER 4

Weakly Supervised Affordance Segmentation

Contents

4.1 Introduction

Our first work deals with weakly supervised segmentation of affordances. Affordances are important as they form the key representation to describe potential interactions. For instance, autonomous agents must recognize what areas are *drivable*, which objects are *movable*. On a finer functionality level, these agents need to recognize the purpose of objects and their parts, like *cuttable* for the blade of a knife. Furthermore, it is desirable that these autonomous systems are able to adapt to novel environments. In order to do so, they must generalize affordances to unseen classes, *e.g.* they must recognize that bowls can contain liquids just like mugs even if they never saw a bowl before.

Existing methods (*Katz et al.*, 2014; *Kim and Sukhatme*, 2014; *Myers et al.*, 2015; *Hermans et al.*, 2011; *Roy and Todorovic*, 2016) learn to infer pixel-wise affordance labels using supervised learning techniques. Since creating pixel-wise annotated datasets is heavily labor intensive, recent works focus mainly on coarse affordance classes like *walkable* or *reachable* which are at a scene level but not at an object level (*Roy and Todorovic*, 2016). An exception is the UMD part affordance dataset (*Myers et al.*, 2015), which provides annotations for objects. In order to simplify the annotation process, a turntable setting has been used to capture the objects. This setup, however, simplifies the task since each image contains only one object that is easy to segment. We therefore propose a more challenging dataset containing images captured in a kitchen environment. The dataset consists of 3090 images containing 9916 object instances. As in (*Myers et al.*, 2015), each pixel is annotated by none or several affordance classes. This is different to other semantic segmentation tasks where a

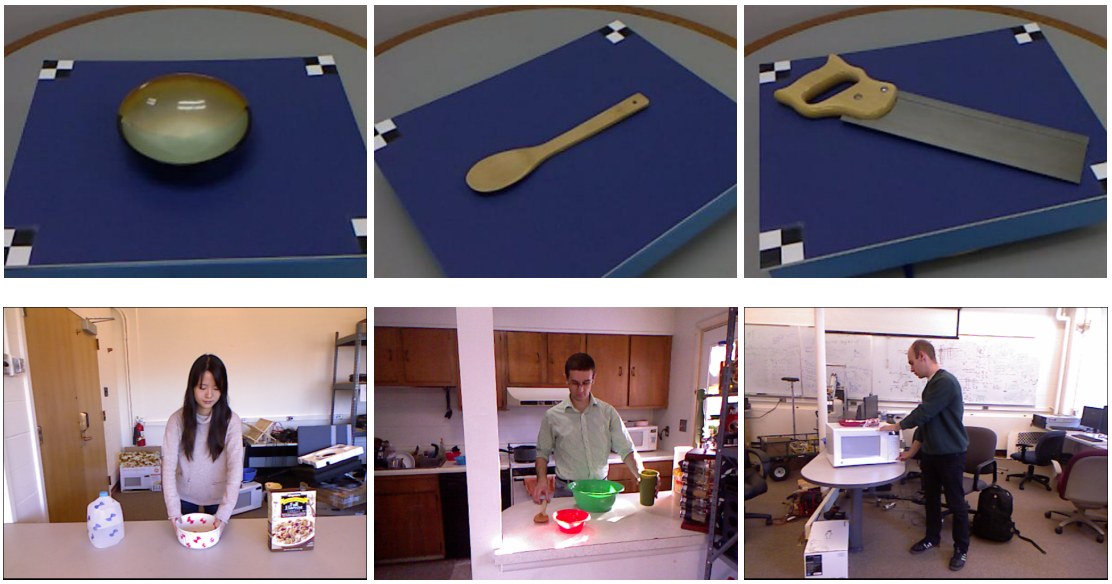


Figure 4.1: Example images from (top row) the UMD part affordance dataset and (bottom row) the CAD120 dataset.

pixel is usually labeled by only one semantic class. To address this, we extend a convolutional neural network (CNN) architecture for segmentation from singlelabel to multilabel classification.

Since CNNs require large amounts of annotated data, it is desirable to train them in a weakly supervised setting. In (*Bearman et al.*, 2016), supervision in form of keypoints has been proposed. Instead of providing segmentation masks, only a very small set of pixels in an image are annotated. We therefore propose an approach for affordance detection that can be learned by such keypoint annotations. In our experiments, we show that our approach outperforms (*Bearman et al.*, 2016) for affordance detection by a large margin. Our approach also achieves a higher affordance detection accuracy than other state-of-the-art methods that utilize weaker supervision at image level (*Papandreou et al.*, 2015; *Kolesnikov and Lampert*, 2016).

4.2 Affordance Datasets

There are not many datasets with pixel-wise affordance labels. Recently, the NYUv2 RGB-D dataset has been augmented with coarse affordance labels like *walkable* and *movable* for entire rooms instead of objects (*Roy and Todorovic*, 2016). In contrast, the publicly available RGB-D dataset proposed by (*Myers et al.*, 2015) focuses on part affordances of everyday tools. The dataset consisting of 28,074 images is collected using a Kinect sensor, which records RGB and depth images at a resolution of 640×480 pixels and provides 7-class pixel-wise affordance labels for objects from 17 categories. Each pixel may belong to multiple affordances at the same time. Each object is recorded on a revolving turntable to cover a full 360° range of views providing clutter-free images of the object as shown in Fig. 4.1. While such lab recordings provide images with high quality, they lack important contextual information such as human-interaction, other objects and typical background.

We therefore adopt a dataset that contains objects within the context of human-interactions in a

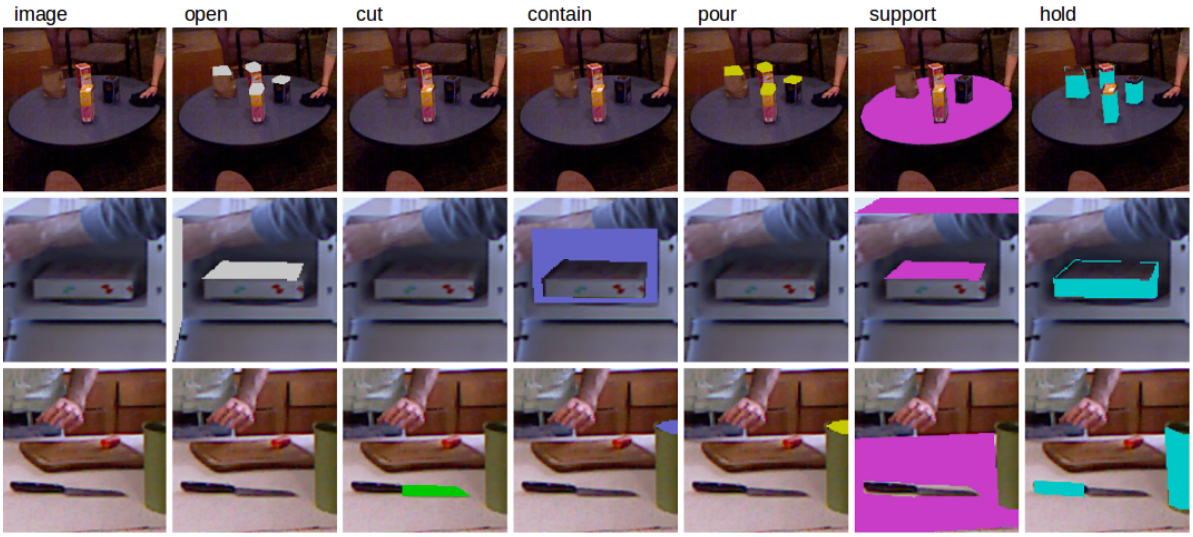


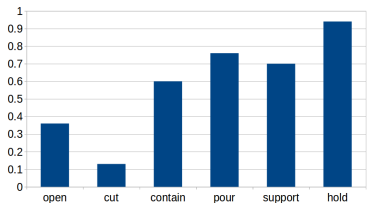
Figure 4.2: Example images with annotations from the proposed CAD120 affordance dataset. Pixels that do not belong to any affordance are considered as background.

more realistic environment. We found the CAD120 dataset (*Koppula and Saxena*, 2014) to be well tailored for our purpose. It consists of 215 videos in which 8 actors perform 14 different high-level activities. Each high-level activity is composed of sub-activities, which in turn involve one or more objects. In total, there are 32 different sub-activities and 35 object classes. A few images of the dataset are shown in Fig. 4.1. The dataset also provides frame wise annotation of the sub-activity, object bounding boxes and automatically extracted human pose.

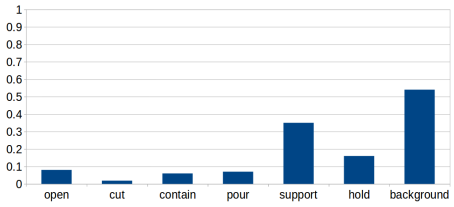
We annotate the affordance labels *openable, cuttable, containable, pourable, supportable, holdable* for every 10^{th} frame from sequences involving an active human-object interaction resulting in 3090 frames. Each frame contains between 1 and 12 object instances resulting in 9916 objects in total. We annotate all object instances with pixel-wise affordance labels. Since the object bounding boxes in the dataset are annotated, we perform all experiments on cropped images after extending the bounding boxes by 30 pixels if possible in each direction. A few annotated cropped images from the dataset are shown in Fig. 4.2. As can be seen, the appearance of affordances can vary significantly, *e.g.* visually distinct object parts like the lid of a box or the door of a microwave have the affordance label *openable*. Similarly, the knife handle and the boxes are *holdable*.

We report some statistics regarding the annotations with respect to the cropped images in Fig. 4.3. Fig. 4.3(a) shows that the generic affordance classes *supportable* and *holdable* occur frequently. The classes *pourable* and *containable* also occur quite often due to the kitchen environment. The class *cuttable* occurs rarely. Except of the background class and *supportable*, the classes cover only a small portion of a cropped image when they are present as shown in Fig. 4.3(b). Fig. 4.3(c) shows that most of the pixels are labeled as background, *i.e.* they are not labeled by any affordance class, but there are also many pixels labeled by two classes. The dataset is well balanced in terms of the number of images contributed by each actor with a median of 382 and a range of 227–606 images per actor. The dataset is publicly available.[1]

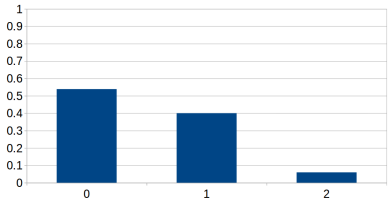
[1] `https://github.com/ykztawas/Weakly-Supervised-Affordance-Detection`



(a) Probability of an affordance class to occur in an image



(b) Image area occupied by an affordance class given its presence in an image



(c) Percentage of pixels that are annotated by 0, 1 or 2 affordance classes

Figure 4.3: Statistics of the dataset

4.3 Proposed Method

For semantic image segmentation, CNNs have shown very good results (*Chen et al.*, 2015, 2018b). For our experiments, we use the VGG-16 architecture as in (*Chen et al.*, 2015) and the ResNet-101 as in (*Chen et al.*, 2018b). In contrast to (*Chen et al.*, 2015, 2018b), we do not use an additional CRF. Since the models (*Chen et al.*, 2015, 2018b) do not handle the multilabel case, we have to modify the architecture. We first describe the learning procedure in the fully supervised setting and then discuss the weakly supervised setting.

4.3.1 Full Supervision

Given an image $\mathbf{I}$ with n pixels, we denote the image pixel positions as $X = \{x_1, x_2 \dots\}$ and the corresponding labeling as $Y = \{y_{i,c}\}$ where $y_{i,c} \in \{0, 1\}$ indicates if pixel at position x_i is labeled by affordance class c. We denote the set of affordances as $\mathcal{C}$.

In the fully supervised case, we train the CNN by optimizing the log likelihood given by

$$J(\theta) = \log P(Y|\mathbf{I};\theta) = \sum_{i=1}^{n} \sum_{c \in \mathcal{C}} \log P(y_{i,c}|\mathbf{I};\theta), \tag{4.1}$$

where θ are the parameters of the CNN. A common loss function for semantic image segmentation is the cross-entropy based on the output of a final softmax layer. This does not work in the multilabel case and we define the loss based on the sigmoid function:

$$P(y_{i,c}|\mathbf{I};\theta) = \frac{1}{1 + \exp\left(-f_{i,c}\left(y_{i,c}|\mathbf{I};\theta\right)\right)}, \tag{4.2}$$

where $f_{i,c}\left(y_{i,c}|\mathbf{I};\theta\right)$ is the output of the CNN at pixel position x_i and affordance c without the softmax layer.

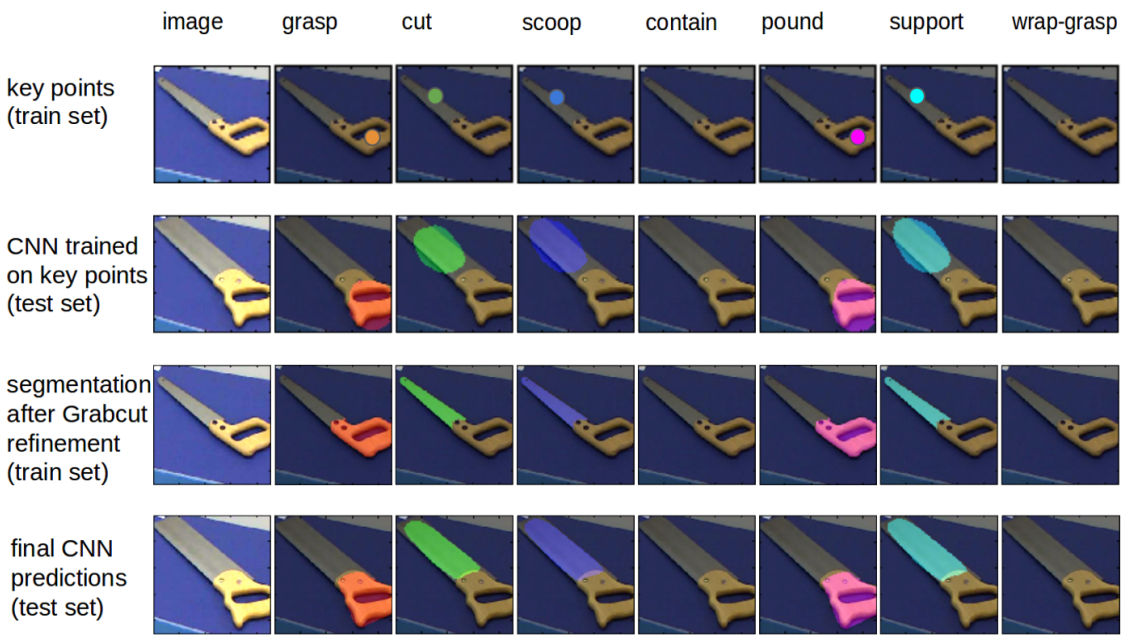


Figure 4.4: Illustration of our approach for weakly supervised affordance detection. The example images are taken from the UMD part affordance dataset (*Myers et al.*, 2015). The first row shows the weak annotations of the training images. The saw is annotated by five keypoints with affordance labels. The second row shows the prediction of the CNN on an image of the test set. If the CNN is only trained on the keypoint annotations, the predictions are not very precise. The third row shows the estimated annotation for the training image after the prediction of the CNN was refined by Grabcut for each affordance class. The last row shows the prediction of the CNN trained on the refined annotations of the training set. Compared to the second row, the affordances are precisely detected.

4.3.2 Weak Supervision

While all pixels are labeled in the fully supervised setting, we will have only very few pixels annotated in the weakly supervised setting. In our setup, weak supervision is provided in terms of keypoints as illustrated in the first row of Fig. 4.4. In this case, the observed variables are image data $\mathbf{I}$ and keypoints $Z = \{(c_k, x_k)\}$, where x_k is an annotated keypoint with label c_k, but the pixel level segmentations Y are latent variables.

The concept of weakly supervised learning consists of estimating Y for the training images while learning the parameters of the CNN.

We use the available pre-trained models on ImageNet for VGG-16 and ResNet-101 as initialization for the CNNs and initialize $\hat{Y}$ by

$$\hat{y}_{i,c} = \begin{cases} 1 & \text{if } |\{(z_k, x_k) \in Z : z_k = c \wedge |x_k - x_i| \leqslant \tau\}| > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

where Z are the available keypoint annotations. We chose $\tau = 40$ for the UMD dataset and $\tau = 50$ for the CAD dataset.

In order to learn the parameters θ of the CNN we maximize

$$\max_{\theta} \sum_{i=1}^{n} \sum_{l \in \mathcal{C}} \log P(\hat{y}_{i,c}|I;\theta). \tag{4.4}$$

The predictions of the learned CNN are reasonable but not very precise as illustrated in the second row of Fig. 4.4. We therefore add an additional training stage.

After updating the parameters of the CNN, we re-compute $P(Y|I;\theta)$ for the training images and compute the probability for the latent variable Y by

$$\begin{aligned} P(Y|I,Z) &= \sum_{c \in \mathcal{C}} P(Y,c|I,Z) && (4.5)\\ &= \sum_{c \in \mathcal{C}} P(Y|c,I,Z)P(c|I,Z) && (4.6)\\ &\approx \sum_{c \in \mathcal{C}} P(Y_c|I;\theta)P(c|Z). && (4.7) \end{aligned}$$

Since we know from Z if an affordance label c is present, we have $P(c|I,Z) = P(c|Z_x)$ and

$$P(c|Z) = \begin{cases} 1 & \text{if } |\{(z_k,x_k) \in Z : z_k = c\}| > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.8}$$

In (4.7), $P(Y_c|I;\theta)$ denotes the probabilities which have been predicted by the CNN for the affordance class c. In order to obtain the final annotation Y for the training images we binarize the predictions by setting

$$\hat{y}_{i,c} = \begin{cases} 1 & \text{if } P(y_{i,c}|I,Z) > 0.5 \\ 0 & \text{otherwise.} \end{cases} \tag{4.9}$$

While this could be already considered as the final estimate $\hat{Y}$ to update the CNN as described in (4.4), we use Grabcut (*Rother et al.*, 2004) to refine the labels for each affordance c independently. To model for each affordance c the color distribution of the affordance region and the background region, we use Gaussian mixture models with 6 components. The distribution for the affordance regions is initialized by the pixels with $\hat{y}_{i,c} = 1$ and distribution for the background by the pixels with $\hat{y}_{i,c} = 0$. The refinement by Grabcut is illustrated in the third row of Fig. 4.4. The final row shows the improved results of the CNN trained on the training images refined by Grabcut.

4.4 Experiments

We first evaluate the fully supervised approach (Section 4.3.1) and compare it with other fully supervised approaches for affordance detection. We then compare the discussed weakly supervised setting (Section 4.3.2) with the fully supervised baseline and state-of-the-art weakly supervised image segmentation methods.

For the UMD part affordance dataset, we use the two defined train-test splits for evaluation. For the first split, which is denoted by *category split*, the object classes are shared among the training and test set. In the second split, which is denoted by *novel split*, the object classes in the test set are not

UMD dataset (category split)	Grasp	Cut	Scoop	Contain	Pound	Support	Wgrasp	mean
Fully Supervised Ranked F-Measure category split								
HMP + SVM (*Myers et al.*, 2015)	0.15	0.04	0.05	0.17	**0.04**	0.03	0.10	0.08
DEP + SRF (*Myers et al.*, 2015)	0.13	0.03	0.10	0.14	0.03	0.04	0.09	0.08
Proposed (VGG)	0.23	**0.08**	**0.18**	**0.21**	**0.04**	0.08	**0.11**	0.13
Proposed (ResNet)	**0.24**	**0.08**	**0.18**	**0.21**	**0.04**	**0.09**	**0.11**	**0.14**
Fully Supervised IoU category split								
HMP + SVM (*Myers et al.*, 2015)	0.57	0.37	0.70	0.77	0.41	0.49	0.79	0.59
DEP + SRF (*Myers et al.*, 2015)	0.35	0.15	0.38	0.65	0.18	0.26	0.80	0.40
Proposed (VGG)	0.66	0.77	0.85	0.84	0.64	**0.73**	0.82	**0.76**
Proposed (ResNet)	**0.71**	**0.79**	**0.86**	**0.86**	**0.72**	0.55	**0.84**	**0.76**
Weakly Supervised IoU category split								
Proposed (VGG) without Grabcut (Train)	0.30	0.21	0.46	0.48	0.26	0.32	0.50	0.36
Proposed (VGG)	0.46	0.48	0.72	**0.78**	0.44	0.53	0.65	0.58
Proposed (VGG) + Grabcut (Test)	**0.57**	**0.68**	**0.73**	0.73	**0.60**	**0.66**	0.76	**0.67**
Proposed (ResNet) without Grabcut (Train)	0.29	0.21	0.47	0.50	0.28	0.33	0.50	0.37
Proposed (ResNet)	0.42	0.35	0.67	0.70	0.44	0.44	**0.77**	0.54
Proposed (ResNet) + Grabcut (Test)	0.52	0.56	0.72	0.72	0.51	0.64	0.76	0.63
Image label (*Papandreou et al.*, 2015)	0.06	0.19	0.04	0.22	0.12	0.02	0.08	0.10
Area constraints (*Papandreou et al.*, 2015)	0.06	0.04	0.10	0.14	0.22	0.04	0.37	0.14
SEC (*Kolesnikov and Lampert*, 2016)	0.39	0.16	0.27	0.13	0.35	0.19	0.07	0.22
WTP (*Bearman et al.*, 2016)	0.16	0.14	0.20	0.20	0.01	0.07	0.13	0.13

Table 4.1: Evaluation of fully and weakly supervised approaches for affordance detection on the UMD part affordance dataset (category split). Evaluation metrics are weighted F-measure and IoU.

present in the training set. The second protocol is more difficult and measures how well the methods generalize across object classes.

For the CAD120 affordance dataset, we also propose two splits. For the first split, which we denote by *actor split*, we reserve images from actors $\{5, 9\}$ as test set and use the images from actors $\{1, 6, 3, 7, 4, 8\}$ as training data. For the second split, which we denote by *object split*, the training set contains the object classes *table*, *plate*, *thermal cup*, *medicine box*, *microwave*, and *bowl* while the test set contains all other object classes.

In (*Myers et al.*, 2015) a ranked weighted F-measure was proposed for measuring the accuracy for affordance detection. The measure takes into account that a pixel can have multiple labels, but assumes that the labels can be ranked. Ranking the labels is often not very intuitive. We therefore also report the accuracy using per class intersection-over-union (IoU), which is also known as Jaccard index, for both datasets.

4.4.1 UMD Part Affordance Dataset

4.4.1.1 Supervised Setting

In (*Myers et al.*, 2015), two approaches have been presented for learning affordances from local appearance and geometric features. The first approach is based on features derived from a superpixel based hierarchical matching pursuit (HMP) together with a linear SVM and the second approach is based on curvature and normal features derived from depth data used within a structured random forest (SRF).

We compare two network architectures. The first one is based on the VGG-16 architecture (*Si-*

UMD dataset (novel split)	Grasp	Cut	Scoop	Contain	Pound	Support	Wgrasp	mean
Fully Supervised Ranked F-Measure novel split								
HMP + SVM (*Myers et al.*, 2015)	0.16	0.02	0.15	0.18	0.02	0.05	0.10	0.10
DEP + SRF (*Myers et al.*, 2015)	0.05	0.01	0.04	0.07	0.02	0.01	0.07	0.04
Proposed (VGG)	**0.18**	**0.05**	**0.18**	**0.20**	**0.03**	**0.07**	**0.11**	**0.12**
Proposed (ResNet)	0.16	**0.05**	**0.18**	0.19	0.02	0.06	**0.11**	0.11
Fully Supervised IoU novel split								
HMP + SVM (*Myers et al.*, 2015)	0.29	0.10	0.61	**0.74**	0.03	0.24	0.63	0.38
DEP + SRF (*Myers et al.*, 2015)	0.32	0.04	0.23	0.42	**0.16**	0.22	0.81	0.31
Proposed (VGG)	**0.37**	0.35	0.65	0.62	0.10	**0.52**	**0.85**	**0.50**
Proposed (ResNet)	0.33	**0.51**	**0.69**	0.52	0.09	0.51	**0.85**	**0.50**
Weakly Supervised IoU novel split								
Proposed (VGG) without Grabcut (Train)	0.16	0.14	0.43	0.45	0.02	0.37	0.40	0.28
Proposed (VGG)	0.27	0.14	0.55	0.58	0.02	0.37	0.67	0.37
Proposed (VGG) + Grabcut (Test)	**0.34**	0.34	0.65	**0.70**	0.08	0.54	**0.73**	0.48
Proposed (ResNet) without Grabcut (Train)	0.16	0.17	0.44	0.40	0.02	0.39	0.44	0.29
Proposed (ResNet)	0.25	0.21	0.62	0.50	0.08	0.43	0.67	0.40
Proposed (ResNet) + Grabcut (Test)	**0.34**	**0.70**	**0.78**	0.62	**0.09**	**0.72**	0.67	**0.56**
Image label (*Papandreou et al.*, 2015)	0.04	0.00	0.09	0.16	0.01	0.02	0.32	0.09
Area constraints (*Papandreou et al.*, 2015)	0.05	0.00	0.04	0.16	0.00	0.01	0.32	0.09
SEC (*Kolesnikov and Lampert*, 2016)	0.12	0.03	0.06	0.23	0.07	0.12	0.25	0.13
WTP (*Bearman et al.*, 2016)	0.11	0.03	0.18	0.11	0.00	0.02	0.23	0.10

Table 4.2: Evaluation of fully and weakly supervised approaches for affordance detection on the UMD part affordance dataset (novel split). Evaluation metrics are weighted F-measure and IoU.

monyan and Zisserman, 2015). For training, we use a mini-batch of 3 images and an initial learning rate of 0.001 (0.01 for the final classifier layer), multiplying the learning rate by 0.1 after every 2000 iterations. We use a momentum of 0.9, weight decay of 0.0005 and run for 6000 iterations. Additionally, we use the ResNet-101 architecture (*Chen et al.*, 2018b). Here we maintained all the hyperparameters from the original paper. The performance comparison on both IoU and ranked weighted F-measure metrics are shown in Tables 4.1 and 4.2.

As can be observed, the trend in performance is similar irrespective of the evaluation metric. The `HMP+SVM` outperforms the `DEP+SRF` combination, indicating that learning features from data is more effective than learning complex classifiers on handcrafted features. Our approach based on the `VGG` architecture as well as the `ResNet` architecture in turn outperform `HMP+SVM` confirming the effectiveness of end-to-end learning. In average, both architectures achieve similar results for both protocols.

When we compare the results in Tables 4.1 and 4.2, which correspond to the protocols *category split* and *novel split*, we observe a lower accuracy for the second protocol that evaluates the generalization across object classes. For the supervised case, the accuracy drops from 0.76 to 0.50. The affordance class *pound* has the largest drop. By looking at the data, we observe that only instances of the two object classes hammer and mallet are marked by *pound* in the training data. In the test data, the affordance appears for the object classes tenderizer, cup, and saw. While for hammer and mallet, the entire object is labeled by *pound*, the tenderizers are only partially labeled as *pound*. As a consequence, our approach tends to label also the entire tenderizer as *pound*. Our approach also does not label parts of a cup as *pound* since mugs, which are in the training set, are not labeled by *pound*. In general, the method needs to observe enough variation in the training data since it might

otherwise overfit to an object class.

4.4.1.2 Weakly Supervised Setting

In case of weak supervision, we evaluate our approach for the VGG architecture and the ResNet architecture. For the VGG architecture we used the same hyperparameters as for supervised learning. For ResNet, we reduced the number of iterations from 20000 to 5000 to reduce the training time.

First, we evaluate the impact of the additional Grabcut step during training as discussed in Section 4.3.2. We denote the results without the Grabcut step by `VGG without Grabcut(Train)` and `ResNet without Grabcut(Train)`. The accuracy drops drastically compared to our proposed method independently of the network architecture as shown in Tables 4.1 and 4.2. When we compare the network architectures VGG and ResNet, we observe that they perform similarly. While `VGG` performs slightly better for the *category split*, `ResNet` is slightly better for the *novel split*.

Since the Grabcut step is essential during training, we also evaluated if an additional refinement by Grabcut of the predictions of the CNN on the test images also improves the results. We denote this setting by `VGG+Grabcut(Test)` and `ResNet+Grabcut(Test)`. On the UMD dataset, this leads to a substantial improvement. For the *novel split*, the weakly supervised method `ResNet+Grabcut(Test)` even outperforms the ResNet trained with full supervision.

However, we will see in the next section that this is not the case for the more challenging CAD120 affordance dataset.

We also compare our approach to other methods that have been proposed for weakly supervised image segmentation.

The methods (*Kolesnikov and Lampert*, 2016; *Papandreou et al.*, 2015) rely on weaker supervision and use annotations at an image level, *i.e.* instead of keypoints only the classes that are present in an image are given without any additional localization of the classes. The method (*Papandreou et al.*, 2015) uses expectation-maximization to train a CNN. The image label based version rejects all classes proposed by the CNN during the E-step but not present in the training image. The area based version uses area priors for foreground and background. It also rejects classes not present in the image, but it also encourages that the background fills at least 40% of the image area and the foreground 20%, respectively. The approach did not always converge and oscillated instead. In these cases, we stopped after the 5th iteration. The SEC method (*Kolesnikov and Lampert*, 2016) uses attention heat maps from classification CNNs and conditional random fields. It is currently the best weakly supervised method on the Pascal VOC 2012 dataset, although it only uses image level supervision.

The method (*Bearman et al.*, 2016) uses the same amount of supervision as our approach, namely keypoints. The method exploits an objectness prior to improve the accuracy. We observed that we obtained better results after removing the dropout layer and replacing the upconvolution layer by the upsampling as it is used in (*Papandreou et al.*, 2015).

The results in terms of IoU are shown in Tables 4.1 and 4.2. SEC outperforms WTP despite of the weaker supervision. This is also consistent with the numbers reported in (*Kolesnikov and Lampert*, 2016; *Bearman et al.*, 2016). Our approach outperforms the other methods for affordance detection by a margin. While our approach requires more supervision than SEC and (*Papandreou et al.*, 2015), our approach also outperforms WTP, which also uses keypoints as annotations.

CAD120 affordance dataset (actor split)	Bck	Open	Cut	Contain	Pour	Support	Hold	Mean
Fully Supervised IoU category split								
Proposed (VGG)	0.81	0.67	0.00	0.54	0.42	0.70	0.64	0.54
Proposed (ResNet)	**0.86**	**0.71**	0.00	**0.61**	**0.45**	**0.79**	**0.70**	**0.59**
Weakly Supervised IoU category split								
Proposed (VGG) without Grabcut (Train)	0.58	0.37	0.10	0.19	0.18	0.18	0.41	0.29
Proposed (VGG)	**0.61**	0.33	0.00	**0.35**	**0.30**	0.22	**0.43**	**0.32**
Proposed (VGG) + Grabcut (Test)	0.60	0.23	**0.14**	0.33	0.28	**0.24**	0.42	**0.32**
Proposed (ResNet) without Grabcut (Train)	0.60	0.37	0.08	0.20	0.17	0.22	0.41	0.29
Proposed (ResNet)	0.60	0.25	0.00	**0.35**	**0.30**	0.17	0.42	0.30
Proposed (ResNet) + Grabcut (Test)	0.58	0.22	0.0	0.29	0.22	0.20	0.32	0.26
SEC (*Kolesnikov and Lampert*, 2016)	0.53	**0.43**	0.00	0.25	0.09	0.02	0.20	0.22
WTP (*Bearman et al.*, 2016)	0.53	0.13	0.00	0.10	0.08	0.11	0.22	0.17
Image label (*Papandreou et al.*, 2015)	0.55	0.05	0.01	0.09	0.10	0.02	0.21	0.15
Area constraints (*Papandreou et al.*, 2015)	0.53	0.11	0.02	0.09	0.09	0.07	0.15	0.15

Table 4.3: Evaluation of fully and weakly supervised approaches for affordance detection on the CAD120 affordance dataset (actor split). The evaluation metric used is IoU.

4.4.2 CAD120 Affordance Dataset

4.4.2.1 Supervised Setting

We first evaluate the fully supervised approaches on the proposed CAD120 affordance dataset, which is discussed in Section 4.2. The results are reported in Tables 4.3 and 4.4. If we compare the results for the *category split* for the UMD part affordance dataset, which is given in Table 4.1, with the *actor split* in Table 4.3, we observe that the accuracies on the proposed CAD120 affordance dataset are lower than the accuracies on UMD since the proposed CAD120 affordance dataset is more challenging, cf. Fig. 4.1. While on UMD both networks achieve 76% mean IoU, they achieve less than 60% on the proposed dataset. In contrast to UMD, the accuracy decreases only slightly when comparing the *actor split* and *object split* in Tables 4.3 and 4.4. This shows that the methods generalize very well across object classes for this dataset while on UMD the methods seem to overfit to the object categories of the training data due to the controlled recording setting. The larger drop in accuracy on UMD, however, can also be explained by annotation inconsistencies across object classes as it is discussed in Section 4.4.1.1.

4.4.2.2 Weakly Supervised Setting

We also evaluate our approach on the dataset in the weakly supervised setting. We perform the same experiments as on the UMD part affordance dataset. We first compare the results when Grabcut is removed from the training procedure, which is denoted by `VGG without Grabcut(Train)` and `ResNet without Grabcut(Train)`. The accuracy drops when Grabcut is omitted as shown in Tables 4.3 and 4.4. When we add Grabcut also for inference on the test images, denoted by `VGG+Grabcut(Test)` and `ResNet+Grabcut(Test)`, we observe that Grabcut does not improve the accuracy. This is in contrast to the UMD part affordance dataset where Grabcut during testing improved the results. The benefit of Grabcut during testing on UMD can be explained by the monotonous background as shown in Fig. 4.1, which simplifies the segmentation.

We also compare our approach to methods for weakly supervised image segmentation (*Bear-*

CAD120 affordance dataset (object split)	Bck	Open	Cut	Contain	Pour	Support	Hold	Mean
Fully Supervised IoU novel split								
Proposed (VGG)	0.76	0.10	0.27	0.60	0.45	0.66	**0.60**	0.49
Proposed (ResNet)	**0.80**	**0.22**	**0.50**	**0.62**	**0.48**	**0.75**	**0.60**	**0.57**
Weakly Supervised IoU novel split								
Proposed (VGG) without Grabcut (Train)	0.61	**0.13**	**0.15**	0.20	0.18	0.14	0.46	0.27
Proposed (VGG)	0.62	0.08	0.08	0.24	**0.22**	0.20	0.46	0.27
Proposed (VGG) + Grabcut (Test)	0.62	0.07	0.05	0.21	0.19	0.27	0.41	0.26
Proposed (ResNet) without Grabcut (Train)	0.60	0.10	0.10	0.16	0.16	0.18	0.38	0.24
Proposed (ResNet)	**0.69**	0.11	0.09	**0.28**	0.21	0.36	**0.56**	**0.33**
Proposed (ResNet) + Grabcut (Test)	**0.69**	0.09	0.04	0.20	0.18	**0.44**	0.48	0.30
SEC (*Kolesnikov and Lampert*, 2016)	0.54	0.04	0.09	0.13	0.09	0.08	0.13	0.16
WTP (*Bearman et al.*, 2016)	0.57	0.01	0.00	0.02	0.09	0.03	0.19	0.13
Image label (*Papandreou et al.*, 2015)	0.58	0.00	0.00	0.00	0.00	0.00	0.23	0.12
Area constraints (*Papandreou et al.*, 2015)	0.59	0.03	0.03	0.01	0.02	0.02	0.28	0.14

Table 4.4: Evaluation of fully and weakly supervised approaches for affordance detection on the CAD120 affordance dataset (object split). The evaluation metric used is IoU.

man et al., 2016; *Kolesnikov and Lampert*, 2016; *Papandreou et al.*, 2015). Among them, SEC (*Kolesnikov and Lampert*, 2016) performs best, yielding 22% mean IoU for the *actor split* and 16% for the *object split*. As for UMD, our approach outperforms SEC and the other methods. While `ResNet` achieves 30% mean IoU for the *actor split* and 33% for the *object split*, `VGG` achieves 32% and 27%, respectively. This is consistent with the UMD dataset where `ResNet` also generalizes better across object categories in comparison to `VGG`.

4.5 Conclusion

In this work, we have addressed the problem of weakly supervised affordance detection. To this end, we proposed a convolutional network that can be trained from weak keypoint annotations. In contrast to object detection and segmentation, affordance detection is a more difficult task due to the higher abstraction level compared to objects and the fact that a part can be associated with multiple affordances. For evaluation, we introduced a pixel-wise annotated affordance dataset containing 3090 images and 9916 object instances with rich contextual information which can be used to further investigate the impact of context on affordance segmentation.

To assess the quality of our method, we compared our approach to several state-of-the-art weakly supervised image segmentation methods on the proposed CAD120 affordance dataset and the UMD part affordance dataset (*Myers et al.*, 2015). On both datasets, our proposed method achieves state-of-the-art performance both in the fully supervised setting as well as in the weakly supervised setting.

CHAPTER 5

Adaptive Binarization for Weakly Supervised Affordance Segmentation

Contents

5.1 Introduction

In Chapter 4, we proposed an iterative approach alternating between updating the parameters of a convolutional neural network and estimating the unknown segmentation masks of the training images. During this process, the predictions of the network need to be binarized and refined with Grabcut. This approach heavily relies on color similarity within one affordance region and color contrast to other regions which might not be given for *e.g.* plastic spoons. Furthermore, the binarization step uses the threshold of 50%. It would be optimal if we trained on pixel-wise labels, but in a weakly supervised setup, the optimal threshold is unknown and varies from image to image.

In this work, we propose an adaptive approach that determines the threshold for binarization for each training image and affordance class. Our approach not only avoids the additional Grabcut refinement but also increases the affordance segmentation accuracy substantially. Since the initialization of the affordance segments based on the keypoints has a high impact on the accuracy, we show further how the parameters for initialization can be determined by cross validation using an approximation of the Jaccard index based on the given keypoints. We evaluate our approach on the CAD 120 affordance dataset we described in the Chapter before and the UMD part affordance dataset (*Myers et al.*, 2015) using two different network architectures. In all settings, our approach outperforms the results from Chapter 4. On the CAD 120 affordance dataset, the mean accuracy is increased by up to 17 percentage points.

5.2 Weakly Supervised Affordance Segmentation

Our approach for weakly supervised affordance segmentation extends the approach from Chapter 4 by adaptive binarization and approximated cross validation for estimating hyperparameters. We therefore briefly describe the previous algorithm first and then describe in Section 5.2.2 the adaptive binarization and in Section 5.2.3 approximated cross validation.

5.2.1 Method

The algorithm from 4 extends fully convolutional neural networks like (*Chen et al.*, 2015) or (*Chen et al.*, 2018b) for the task of affordance segmentation. In contrast to semantic image segmentation, where only one label per pixel needs to be predicted, affordance segmentation requires to predict a set of labels per pixel since an object region might contain multiple affordance types. The approach predicts $P(Y|I;\theta)$ where I denotes the input image, θ denotes the parameters of the model, *i.e.* the weights of the neural network, and $Y = \{y_{i,c}\}$ with $y_{i,c} \in \{0,1\}$ is the pixel-wise segmentation. If $y_{i,c} = 1$ the affordance type c is predicted for pixel i. Due to the multi-label problem, the network uses a sigmoid layer instead of a softmax layer (*Chen et al.*, 2015, 2018b):

$$P(y_{i,c} = 1|I;\theta) = \frac{1}{1 + \exp\left(-f_{i,c}\left(y_{i,c}|I;\theta\right)\right)}, \tag{5.1}$$

where $f_{i,c}$ is the value of the previous layer of the neural network. For segmentation, the predicted probabilities $P(y_{i,c}|I;\theta)$ need to be binarized. In Chapter 4, this was achieved by the standard 50% threshold:

$$\hat{y}_{i,c} = \begin{cases} 1 & \text{if } P(y_{i,c} = 1|I;\theta) \geqslant 0.5 \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

The model parameters θ are determined during training. In the strongly supervised setting, training means optimizing the log-likelihood:

$$J(\theta) = \log P(Y|I;\theta) = \sum_{i=1}^{n} \sum_{c \in \mathcal{C}} \log P(y_{i,c}|I;\theta). \tag{5.3}$$

In the weakly supervised setting, the log-likelihood can not be calculated since Y is not given during training. In Chapter 4, it was proposed to train the model only from a set of keypoints $Z = \{(c_k, x_k)\}$, which denote the presence of the affordance c_k at pixel position x_k, using expectation-maximization (EM). During training, both Y and θ need to be estimated from Z. The approach starts with an initial estimate $\hat{Y}$, which is derived from the keypoints Z by labeling all pixels within a radius of τ around a keypoint:

$$\hat{y}_{i,c} = \begin{cases} 1 & \text{if } |\{(c_k, x_k) \in Z : c_k = c \wedge |x_k - x_i| \leqslant \tau\}| > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{5.4}$$

In contrast to the method in Chapter 4 that uses fixed values for initialization, we discuss in Section 5.2.3 how τ can be estimated by approximated cross validation.

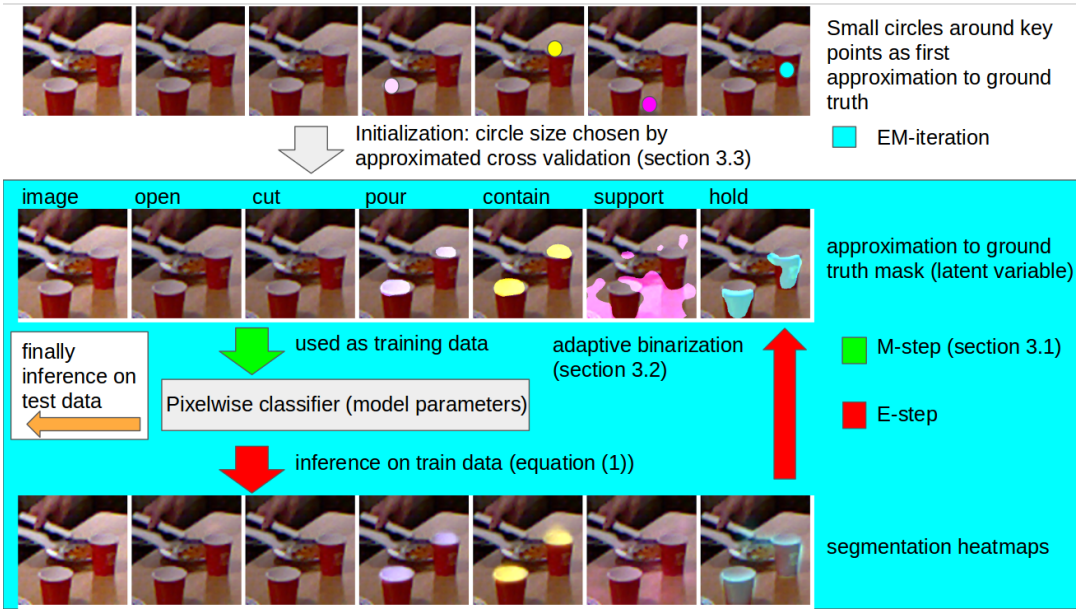


Figure 5.1: Illustration of our approach for affordance segmentation using keypoints as weak supervision. The CNN is trained by iteratively updating the segmentation masks for the training images (E-step) and the parameters of the network (M-step).

After $\hat{Y}$ is estimated, the weights of the network θ are updated by optimizing $J(\theta) = \log P(\hat{Y}|I;\theta)$. Given the new weights θ, the CNN predicts $P(y_{i,c}|I;\theta)$ for each training image and $\hat{Y}$ is refined by binarization of the CNN predictions. The 50% threshold used in (5.2), however, is only valid for the fully supervised setting. While in Chapter 4 an additional GrabCut step is used to address this issue, we propose an adaptive approach that determines the threshold for binarization for each training image and affordance class. This not only increases the accuracy, but it also reduces the training time since an additional GrabCut step is not needed anymore by our approach. The approach for adaptive binarization is discussed in Section 5.2.2. Our weakly supervised approach for affordance segmentation is illustrated in Figure 5.1.

To reduce overfitting and perform approximated cross validation as described in Section 5.2.3, we split the training set into three equally sized subsets A, B, and C. During the M-step, we train the convolutional network on each of the tuples (A,B), (B,C), and (C,A). During the E-step, each network predicts $P(y_{i,c}|I;\theta)$ for the set that was not used for training. As in the approach in Chapter 4, we use two EM iterations to obtain $\hat{Y}$ for all training images. The final CNN model is then obtained by optimizing $J(\theta) = \log P(\hat{Y}|I;\theta)$ on the entire training set.

5.2.2 Adaptive Binarization

We first want to explain why the binarization as described in Equation 5.2 is not optimal for the weakly supervised case. Let us first consider an optimal classifier that separates two classes perfectly in the training data. In this case, $P(y_{i,c} = 1|I;\theta) \geqslant 0.5$ if a pixel is annotated by $y_{i,c} = 1$ and $P(y_{i,c} = 1|I;\theta) < 0.5$ if it is annotated by $y_{i,c} = 0$.

Hence, using 50% as threshold for binarization is optimal. For weakly supervised learning, $\hat{Y}$ is in particular after the initialization only a poor estimation of the unknown ground truth segmentation

masks Y of the training data such that $\hat{y}_{i,c} \neq y_{i,c}$ for many pixels. This means that the optimal threshold is unknown. However, we can use the keypoints Z to obtain an estimate of the threshold:

$$\hat{y}_{i,c} = \begin{cases} 1 & \text{if } P(y_{i,c} = 1|I;\theta) \geqslant t \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

where

$$t = \min\left\{0.5, g\left(\{P(y_{i_k,c} = 1|I;\theta)\}_{c_k=c}\right)\right\}. \tag{5.6}$$

$\{P(y_{i_k,c} = 1|I;\theta)\}_{c_k=c}$ are the predictions of the classifier for all keypoints in the training image I with label c and g computes either the mean or median of $\{P(y_{i_k,c} = 1|I;\theta)\}_{c_k=c}$. In our default experimental setting, we will have only one keypoint for each affordance occurring in an image. In general, one can expect that the threshold is below 0.5 since the ratio $\frac{|\{i:y_{i,c}=0 \wedge \hat{y}_{i,c}=1\}|}{|\{i:y_{i,c}=0\}|}$ is usually lower than $\frac{|\{i:y_{i,c}=1 \wedge \hat{y}_{i,c}=0\}|}{|\{i:y_{i,c}=1\}|}$. As soon as the threshold reaches 0.5, we can replace the adaptive threshold by 0.5. We therefore limit the threshold by 0.5.

5.2.3 Approximated Cross Validation

In the fully supervised setup, hyperparameters can be optimized by cross-validation on the training set using the same measure that is also used for evaluation. Since the ground truth masks Y, however, are unknown in the weakly supervised setup, exact cross validation is not possible. We therefore propose to approximate the Jaccard index, which measures the intersection over union between the ground-truth Y and the prediction $\hat{Y}$, on the validation set. Since the Jaccard index is computed per affordance class c and then averaged over all classes, we discuss only the binary case with $y_i \in \{0, 1\}$. Let $P(y_i = 1) = \frac{|\{i:y_i=1\}|}{|\{i\}|}$ be the unknown percentage of pixels with $y_i = 1$ and $P(\hat{y}_i = 1) = \frac{|\{i:\hat{y}_i=1\}|}{|\{i\}|}$ the known percentage of pixels that have been classified with $\hat{y}_i = 1$. We can approximate $P(\hat{y}_i = 1|y_i = 1)$ by measuring how often a keypoint annotated by the affordance class has been correctly classified. Similarly, $P(\hat{y}_i = 1|y_i = 0)$ is given by the percentage of keypoints that have been misclassified. This gives the relation

$$\begin{aligned} P(\hat{y}_i = 1) =& P(\hat{y}_i = 1|y_i = 1)P(y_i = 1) \\ &+ P(\hat{y}_i = 1|y_i = 0)(1 - P(y_i = 1)) \end{aligned} \tag{5.7}$$

and thus

$$P(y_i = 1) = \frac{P(\hat{y}_i = 1) - P(\hat{y}_i = 1|y_i = 0)}{P(\hat{y}_i = 1|y_i = 1) - P(\hat{y}_i = 1|y_i = 0)}. \tag{5.8}$$

The Jaccard index which is

$$\eta = \frac{|\{i : y_i = 1 \wedge \hat{y}_i = 1\}|}{|\{i : y_i = 1\}| + |\{i : y_i = 0 \wedge \hat{y}_i = 1\}|} \tag{5.9}$$

can then be approximated by

$$\eta_{approx} = \frac{P(\hat{y}_i = 1|y_i = 1)P(y_i = 1)}{P(y_i = 1) + P(\hat{y}_i = 1|y_i = 0)(1 - P(y_i = 1))}. \tag{5.10}$$

As mentioned in Section 5.2.1, we split the training set into three subsets for approximate cross-validation.

CAD 120	Background	Open	Cut	Contain	Pour	Support	Hold	Mean
non-adaptive (VGG)	0.62	0.09	0.20	0.41	0.35	0.11	0.40	0.31
adaptive (VGG)	0.68	0.10	0.23	0.44	0.36	0.50	0.47	0.40

UMD	Grasp	Cut	Scoop	Contain	Pound	Support	Wgrasp	mean
non-adaptive (VGG)	0.32	0.12	0.48	0.46	0.08	0.33	0.69	0.36
adaptive (VGG)	0.31	0.18	0.56	0.49	0.08	0.41	0.66	0.38

Table 5.1: Comparison of adaptive binarization with non-adaptive binarization. The Jaccard index is reported for the object split of CAD 120 affordance dataset and the novel split of the UMD part affordance dataset.

CAD 120	Background	Open	Cut	Contain	Pour	Support	Hold	Mean
Max thres. 1.0 (VGG)	0.62	0.08	0.21	0.34	0.33	0.39	0.19	0.31
Max thres. 0.5 (VGG)	0.68	0.10	0.23	0.44	0.36	0.50	0.47	0.40

UMD	Grasp	Cut	Scoop	Contain	Pound	Support	Wgrasp	mean
Max thres. 1.0 (VGG)	0.32	0.04	0.36	0.42	0.05	0.23	0.64	0.29
Max thres. 0.5 (VGG)	0.31	0.18	0.56	0.49	0.08	0.41	0.66	0.38

Table 5.2: Impact of limiting the adaptive threshold (5.5) by 0.5. The Jaccard index is reported for the object split of the CAD 120 affordance dataset and the novel split of the UMD part affordance dataset.

5.3 Experiments

For evaluation, we use the CAD 120 affordance dataset we described in the Chapter before and the UMD part affordance dataset (*Myers et al.*, 2015). We use the splits separating the object classes (novel split on UMD and object split on CAD) and the splits which do not separate the object classes (category split on UMD and actor split on CAD). As measure, we use the Jaccard index. We report the results using the VGG architecture (*Chen et al.*, 2015) and the ResNet architecture (*Chen et al.*, 2018b) as underlying convolutional network. First, we conduct ablation experiments to show the impact of our two key components, adaptive binarization and approximated cross validation. Second, we compare our approach with other weakly supervised segmentation approaches. If not otherwise specified, we use our approach based on the VGG architecture with adaptive binarization and approximate cross validation to determine τ (5.4). As in Chapter 4, we use one keypoint per affordance class and training image. In Section 5.3.3, we also evaluate the impact of the number of keypoints.

5.3.1 Adaptive Binarization

First we evaluate the impact of adapting the binarization to each training image and affordance class in comparison to using a constant threshold for each affordance class. To this end, instead of using $\min\{0.5, g(\{P(y_{i_k,c}=1|I;\theta)\}_{c_k=c})\}$ as an individual threshold for each image I, we take the average of these thresholds over all images in the training set labeled with the affordance class c. Note that $g(\{P(y_{i_k,c}=1|I;\theta)\}_{c_k=c}) = P(y_{i_k,c}=1|I;\theta)$ in this experiment since we use only one keypoint per affordance c and image I.

The results for the object split of the CAD 120 affordance dataset and the novel split of the UMD part affordance dataset are shown in Table 5.1. Compared to the proposed adaptive binarization approach, the accuracy decreases for all affordance classes and the background, which are regions annotated without any affordance class. In average, the accuracy decreases by -9%. On UMD, the

	approx. Jaccard train			Jaccard test		
τ relative to image width	$0.03w$	$0.06w$	$0.12w$	$0.03w$	$0.06w$	$0.12w$
CAD actor split	0.38	**0.40**	0.30	0.41	**0.42**	0.37
CAD object split	0.48	**0.50**	0.39	0.38	**0.40**	0.35
UMD category split	0.57	**0.58**	0.44	**0.61**	0.59	0.44
UMD novel split	**0.66**	0.62	0.44	**0.38**	**0.38**	0.35

Table 5.3: Impact of τ (5.4). The second column contains the approximated Jaccard index (5.10) computed on the training data for three values of τ. The approximated Jaccard index is used to determine τ. The third column contains the Jaccard index computed on the test data for three values of τ.

decrease is smaller but still -2%. The effect on CAD is larger since the size of the affordance regions varies more across the training images in comparison to UMD.

As discussed in Section 5.2.2, we limit the adaptive threshold by 0.5, which is the optimal threshold for a fully supervised trained model. Table 5.2 shows the results when the threshold is not limited, *i.e.*, the adaptive threshold can even get close to one. As expected, the accuracy drops for both datasets by -9% since a threshold above 0.5 would produce even in the fully supervised case too small affordance segments.

5.3.2 Approximated Cross Validation

The initialization depends on the value τ, which determines the initial affordance segments around the keypoints (5.4). This is shown in the last column of Table 5.3 where we report the mean Jaccard index for three values of τ. Note that τ is set proportional to the image width w. The results show that the accuracy strongly depends on the initialization. The strongest variation can be observed for the category split of the UMD part affordance dataset where the accuracy varies between 0.44 to 0.61. The approximated Jaccard index computed from the keypoints in the training set, which is reported in the second column of Table 5.3, however, correlates with the Jaccard index on the test set. This shows that using approximated cross validation to determine τ works very well in practice. Note that the values between the Jaccard index and its approximation differ since the first measure is computed over the test set and the second over the training set. In all experiments except of Table 5.3, we have determined τ by approximated cross validation.

5.3.3 Varying Number of Keypoints

Our approach also works with multiple keypoints per affordance class in an image. In this case, we compare two functions for $g\left(\{P(y_{i_k,c}=1|I;\theta)\}_{c_k=c}\right)$ (5.5), namely taking the average or the median of $\{P(y_{i_k,c}=1|I;\theta)\}_{c_k=c}$. The results are reported in Figure 5.2. For the object split of the CAD 120 affordance dataset, average and median perform similar and the accuracy increases only slightly after three keypoints. A similar behavior can be observed for the novel split of the UMD part affordance dataset, but here average performs better than the median.

5.3.4 Comparison to the State-of-the-art

We finally compare our approach with other weakly supervised semantic segmentation approaches (*Kolesnikov and Lampert*, 2016; *Bearman et al.*, 2016; *Papandreou et al.*, 2015) and the

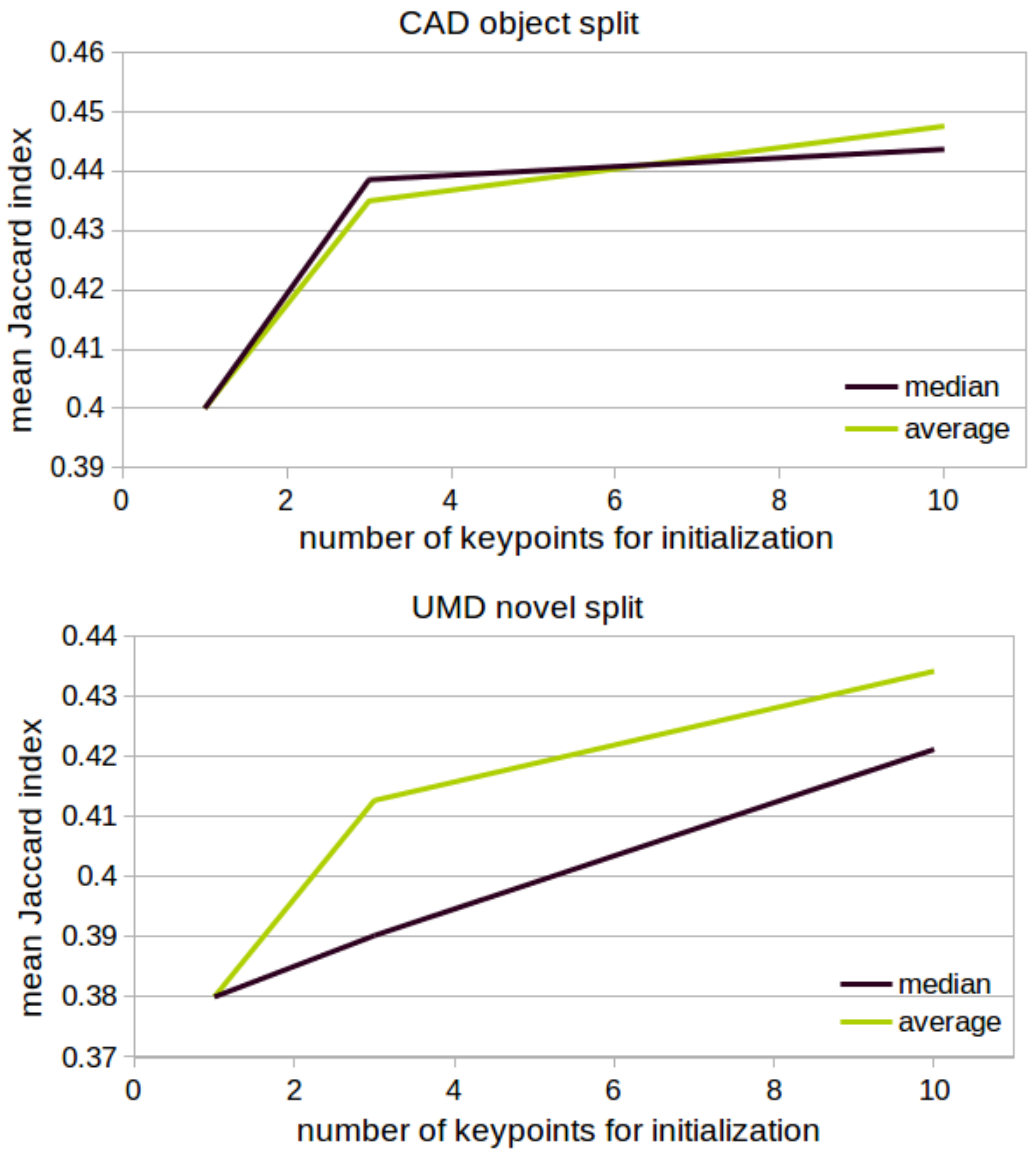


Figure 5.2: Affordance segmentation with more than one keypoint per image and affordance. For the function g (5.5), we compare average and median. The mean Jaccard index is plotted over the number of keypoints.

CAD 120	Background	Open	Cut	Contain	Pour	Support	Hold	Mean
image label supervision - actor split								
Area constraints (*Papandreou et al.*, 2015)	0.53	0.11	0.02	0.09	0.09	0.07	0.15	0.15
SEC (*Kolesnikov and Lampert*, 2016)	0.53	0.43	0.00	0.25	0.09	0.02	0.20	0.22
keypoint supervision - actor split								
WTP (*Bearman et al.*, 2016)	0.53	0.13	0.00	0.10	0.08	0.11	0.22	0.17
method in Chapter 4 (VGG)	0.61	0.33	0.0	0.35	0.30	0.22	0.43	0.32
Proposed (VGG)	0.71	0.47	0.0	0.36	0.37	0.56	0.49	0.42
method in Chapter 4 (ResNet)	0.60	0.25	0.00	0.35	0.30	0.17	0.42	0.30
Proposed (ResNet)	0.77	0.50	0.00	0.43	0.39	0.64	0.56	0.47
image label supervision - object split								
Area constraints (*Papandreou et al.*, 2015)	0.59	0.03	0.03	0.01	0.02	0.02	0.28	0.14
SEC (*Kolesnikov and Lampert*, 2016)	0.54	0.04	0.09	0.13	0.09	0.08	0.13	0.16
keypoint supervision - object split								
WTP (*Bearman et al.*, 2016)	0.57	0.01	0.00	0.02	0.09	0.03	0.19	0.13
method in Chapter 4 (VGG)	0.62	0.08	0.08	0.24	0.22	0.20	0.46	0.27
Proposed (VGG)	0.68	0.10	0.23	0.44	0.36	0.50	0.47	0.40
method in Chapter 4 (ResNet)	0.69	0.11	0.09	0.28	0.21	0.36	0.56	0.33
Proposed (ResNet)	0.74	0.15	0.21	0.45	0.37	0.61	0.54	0.44

Table 5.4: Comparison of our method to the state-of-the-art on the CAD 120 affordance dataset. The Jaccard index is reported.

method in Chapter 4. The results for both splits on the CAD 120 affordance dataset are reported in Table 5.4, while the results for the UMD part affordance dataset are reported in Table 5.5. The methods (*Kolesnikov and Lampert*, 2016; *Papandreou et al.*, 2015) use only image labels and therefore weaker supervision. It is therefore expected that methods that use more supervision in form of keypoints achieve a higher accuracy. For the methods (*Bearman et al.*, 2016), *method in Chapter 4* and our approach, we use one keypoint for each affordance class in an image. The parameter τ has been determined by approximated cross validation. We also report the results as in Chapter 4 for the VGG architecture and the ResNet architecture. Our approach outperforms *method in Chapter 4* and the other methods on both datasets. While our approach achieves with the ResNet architecture on all datasets and splits a better mean accuracy than VGG, this is not the case for the previous method where VGG is sometimes better. For the actor split of the CAD 120 affordance dataset, the mean accuracy is improved by $+17\%$ compared to Chapter 4. This shows the benefit of adaptive binarization for weakly supervised affordance segmentation. Qualitative results are shown in Figure 5.3.

5.4 Conclusion

In this work, we have proposed an approach for affordance segmentation that requires only weak supervision in the form of sparse keypoints. Our approach builds on the method introduced in the Chapter before, but it does not require an additional graph cut segmentation step. This has been achieved by an adaptive approach for binarizing the predictions of a convolutional neural network during training. By approximating the Jaccard index based on the keypoints, we are also able to optimize parameters for the initialization. This approach could also be used to optimize other hyperparameters. We evaluated our approach on the CAD 120 affordance and the UMD part affordance dataset. Our approach outperforms the state-of-the-art for weakly supervised affordance segmenta-

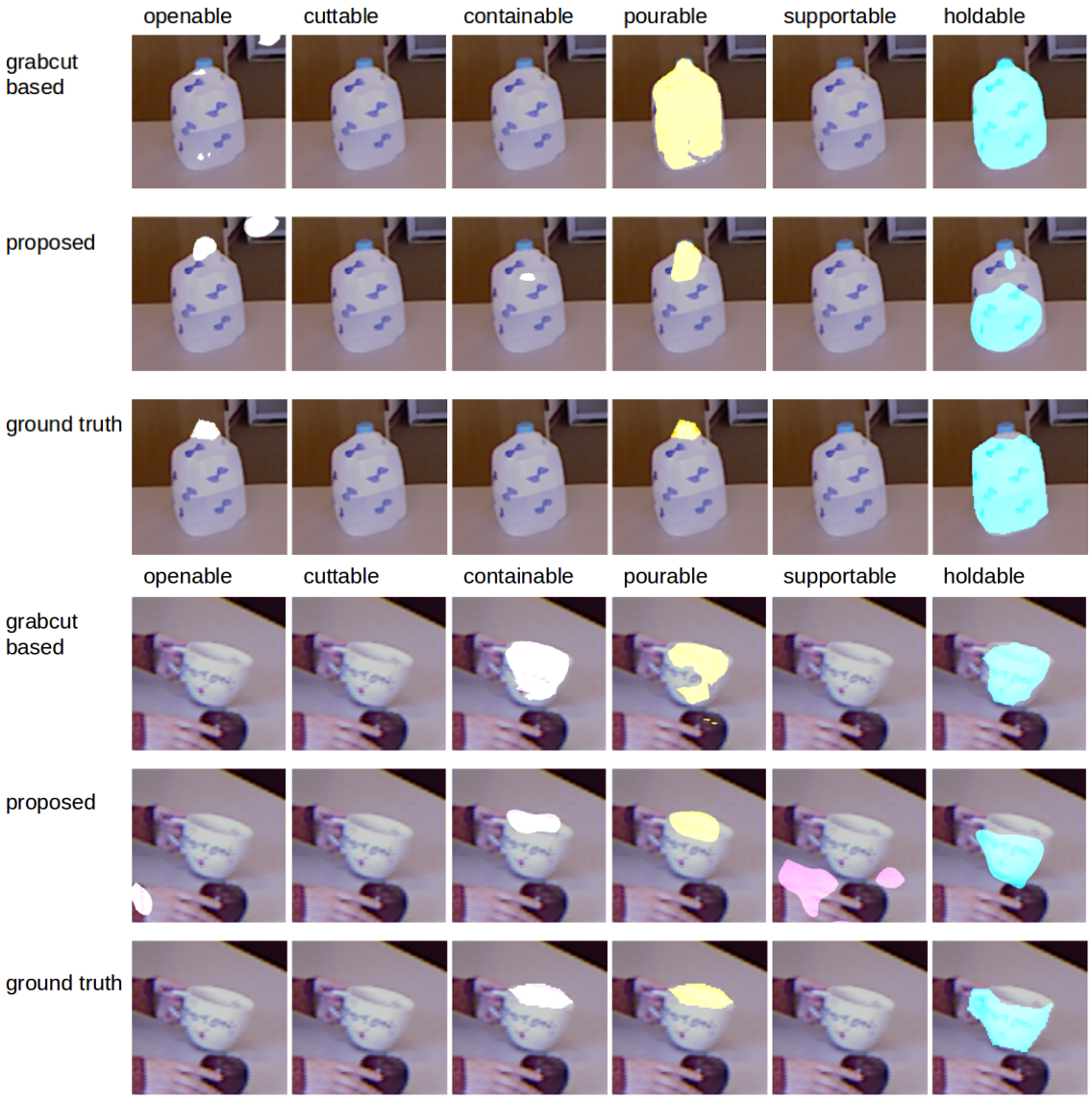


Figure 5.3: Qualitative comparison of our approach (second and fifth row) with the one from Chapter 4. Our approach localizes even small affordance parts while the Grabcut step in the earlier method merges the cap with the entire object.

UMD	Grasp	Cut	Scoop	Contain	Pound	Support	Wgrasp	mean
image label supervision - category split								
Area constraints (*Papandreou et al.*, 2015)	0.06	0.04	0.10	0.14	0.22	0.04	0.37	0.14
SEC (*Kolesnikov and Lampert*, 2016)	0.39	0.16	0.27	0.13	0.35	0.19	0.07	0.22
keypoint supervision - category split								
WTP (*Bearman et al.*, 2016)	0.16	0.14	0.20	0.20	0.01	0.07	0.13	0.13
method in Chapter 4 (VGG)	0.46	0.48	0.72	0.78	0.44	0.53	0.65	0.58
Proposed (VGG)	0.55	0.48	0.72	0.76	0.49	0.48	0.67	0.59
method in Chapter 4 (ResNet)	0.42	0.35	0.67	0.70	0.44	0.44	0.77	0.54
Proposed (ResNet)	0.57	0.54	0.71	0.70	0.43	0.54	0.69	0.60
image label supervision - novel split								
Area constraints (*Papandreou et al.*, 2015)	0.05	0.00	0.04	0.16	0.00	0.01	0.32	0.09
SEC (*Kolesnikov and Lampert*, 2016)	0.12	0.03	0.06	0.23	0.07	0.12	0.25	0.13
keypoint supervision - novel split								
WTP (*Bearman et al.*, 2016)	0.11	0.03	0.18	0.11	0.00	0.02	0.23	0.10
method in Chapter 4 (VGG)	0.27	0.14	0.55	0.58	0.02	0.37	0.67	0.37
Proposed (VGG)	0.31	0.18	0.56	0.49	0.08	0.41	0.66	0.38
method in Chapter 4 (ResNet)	0.25	0.21	0.62	0.50	0.08	0.43	0.67	0.40
Proposed (ResNet)	0.34	0.34	0.58	0.40	0.07	0.42	0.77	0.42

Table 5.5: Comparison of our method to the state-of-the-art on the UMD part affordance dataset. The Jaccard index is reported.

tion. On the CAD 120 affordance dataset, the mean accuracy is increased by up to 17 percentage points compared to the approach in Chapter 4.

CHAPTER 6

Learning Affordances from Very Few Examples

Contents

6.1 Introduction

In Chapters 4 and 5 we used keypoints on affordances as supervision cues. Although they are cheaper than polygons around image segments, the annotation cost still scales linearly with the number of affordance regions. However, one can do better and arrive at a constant cost: In this Chapter we show how to extend a tiny training set containing images with affordance annotations to make the training of a semantic segmentation CNN feasible. We assume that for the training set, we have only a handful of images per object category. For each image, the bounding box of the object and the the bounding boxes for all affordances of the object parts are given. Since training a CNN on such a small training set will be prone to overfitting, we make use of additional data where objects are already annotated by bounding boxes, but annotations of affordances or object parts are missing. We term the additional dataset as unlabeled since the images are not labeled in terms of affordances. Such data is already available, for instance, in form of object detection datasets.

In order to train the CNN on both datasets, *i.e.*, the small dataset with affordance annotations and the large dataset with only object annotations, we transfer the affordance annotations from the

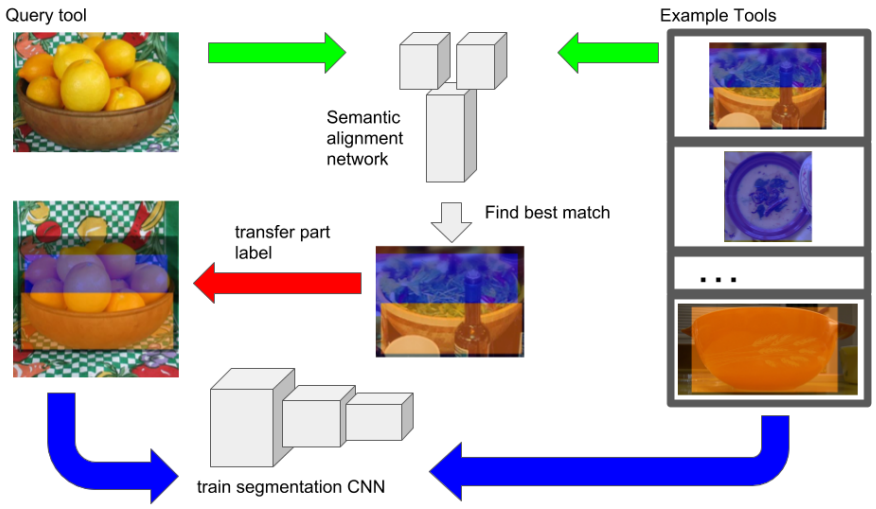


Figure 6.1: In order to train a network to segment affordances from a very small set of examples, we transfer labels to unlabeled images. The training data consists of a set of objects where affordances are annotated by bounding boxes (right). This training set is very small and comprises only a few examples per object category. We then collect more examples of objects from an object detection dataset, *i.e.*, the bounding box and the name of the object are given but not the affordances (left). To transfer the annotation labels of the training set to the new images, we use a semantic alignment network to find for each new image the most similar image in the training set. The bounding box annotations of the affordances are then transferred to the matched images and a CNN is then trained on all images.

small dataset to the unlabeled images of the large dataset. For the label transfer, we use a semantic alignment network, which is trained without supervision, to find for each unlabeled image the most similar labeled image. Despite of having only bounding box annotations of affordances, we then train a CNN for pixel-wise affordance segmentation weakly supervised on both datasets. We evaluate our approach on the IIT-AFF dataset (*Nguyen et al.*, 2017b) and the Pascal Parts dataset (*Chen et al.*, 2014) where our approach outperforms other segmentation approaches that are also trained weakly supervised.

6.2 Label Transfer for Affordance Segmentation

Since annotating affordances or parts of objects is time-consuming, our goal is to train a convolutional network that segments affordances in images on a very small set of annotated images and additional unlabeled images. An overview of the approach is given in Figure 6.1.

Our training set consists of a few example images for each object category where the affordances are annotated by bounding boxes. Since large datasets for object detection exist, we make use of them to extend the training set. These datasets, however, do not provide any annotations of affordances or parts but only bounding boxes for the objects. We therefore transfer the affordance labels from our training set to the objects from an object detection dataset. To this end, we first use a semantic alignment network to retrieve for each unlabeled image the most similar annotated training

image to transfer the annotations (Section 6.2.1). We then train a fully convolutional network on the original training set and the extended set with transferred labels and use this model for inference (Section 6.2.2).

6.2.1 Semantic Alignment Network for Similarity Estimation

For similarity matching between annotated example objects and unannotated query objects, we use the semantic alignment network proposed in (*Rocco et al.*, 2018). It takes two images $\mathbf{I}_s$ to $\mathbf{I}_t$ and predicts an affine transformation T_{aff} and a thin plate spline transformation T_{tps} whose concatenation semantically aligns $\mathbf{I}_s$ to $\mathbf{I}_t$. The transformations are subsequently predicted by two networks only differing in the final layer.

First the feature maps of both images $f^s_{ij:}$ and $f^t_{kl:}$, where i and j are the spatial coordinates in the source image $\mathbf{I}_s$ and k and l are the spatial coordinates in the target image $\mathbf{I}_t$, are extracted in two Siamese branches. Then, a 4D-tensor ν of space match scores is obtained via

$$\nu_{ijkl} = \frac{\langle f^s_{ij:}, f^t_{kl:} \rangle}{\sqrt{\sum_{a,b} \langle f^s_{ab:}, f^t_{kl:} \rangle^2}}. \tag{6.1}$$

Next, the parameters G of the geometrical transformation T_G are calculated from the space match score tensor S. This yields then the 4D inlier mask tensor M:

$$m_{ijkl} = \begin{cases} 1 & \text{if } d(T_G(i,j),(k,l)) < \tau \\ 0 & \text{otherwise.} \end{cases} \tag{6.2}$$

where d is the Euclidean distance. For τ, we use the same value as in (*Rocco et al.*, 2018). Combining M and S provides the soft inlier count, a measure for the quality of the alignment:

$$\kappa = \sum_{i,j,k,l} \nu_{ijkl} m_{ijkl}. \tag{6.3}$$

Intuitively, the feature vectors of pixels in the target and the warped image should be similar if the points are spatially close ($m_{ijkl} = 1$). Therefore, $-\kappa$ serves as a training loss.

We use the pre-trained model (*Rocco et al.*, 2018), which has been first trained on synthetic data obtained from the Pascal dataset (*Everingham et al.*, 2014) and then finetuned with image pairs from the PF-PASCAL dataset (*Ham et al.*, 2017). Since the loss does not require human supervision and the network does not explicitly take any note of the object class, the model generalizes to unseen object classes. We therefore can use the model trained on Pascal classes on the IIT-AFF dataset.

The approach, however, fails for large transformations. Already 2D rotations by more than 30 degrees lead to poor semantic alignments. We therefore augment each of the annotated examples by rotating it by 90, 180 and 270 degrees and flipping it. To find the best match in our annotated training set $\{\mathbf{I}_i\}_{i \in \{1,\dots n\}}$for a query image $\mathbf{J}$, we compute (6.3) for $\mathbf{J}$ and each image $\mathbf{I}_i$, which contains the same object class as $\mathbf{J}$. The best match for $\mathbf{J}$ is then given by the image $\mathbf{I}_k$ with the highest soft inlier count c.

6.2 shows some examples for the top 3 matches.

In order to transfer the affordance labels from $\mathbf{I}_k$ to $\mathbf{J}$, the estimated warping transformation could be used. However, our experiments reveal that the estimated transformations are not accurate enough for transferring the labels. Instead, we scale $\mathbf{I}_k$ to match the size of $\mathbf{J}$ and copy the annotations from $\mathbf{I}_k$ to $\mathbf{J}$.

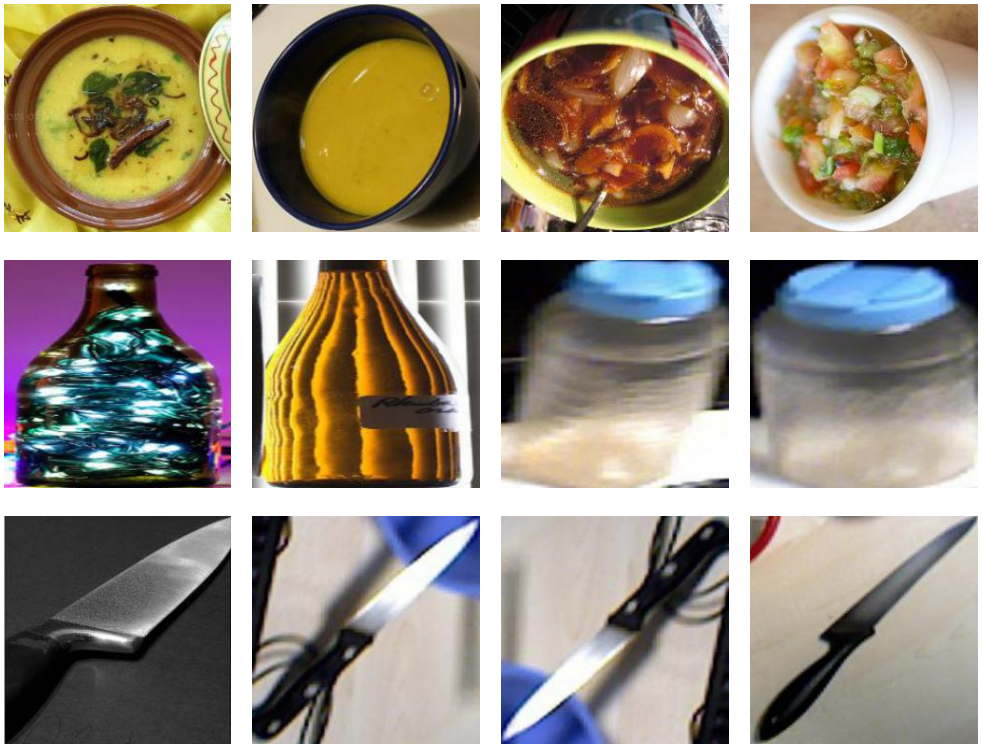


Figure 6.2: Some query tools (left column) and the top 3 matching example tools with decreasing proximity from left to right. Except for the second match for the knife, the matching procedure retrieves tools with seen from similar viewpoint and having same orientation.

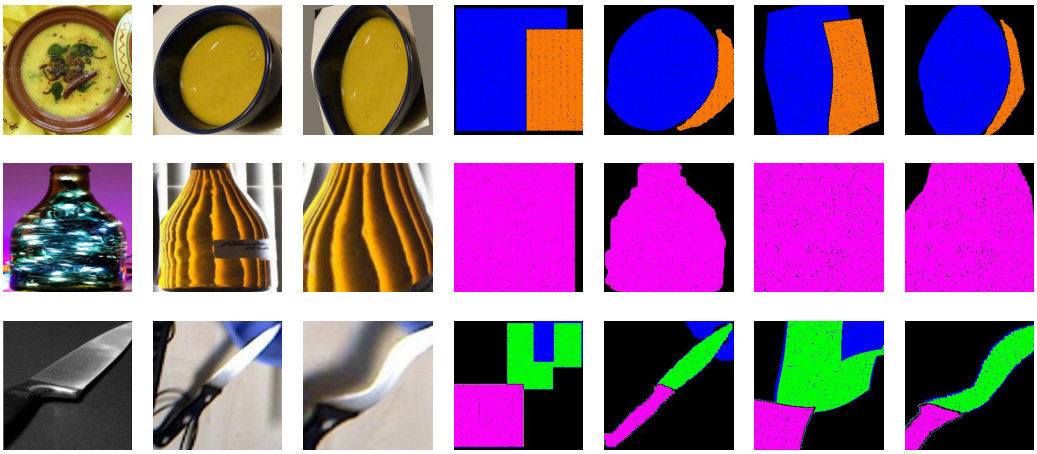


Figure 6.3: Illustration of our supervision levels and transfer strategies. From left to right: Query tool, matched example tool, aligned example tool, “bbox-copy” labels,“pixel-wise-copy” labels,“bbox-warp” labels,“pixel-wise-warp” labels.

6.2.2 Semantic Segmentation

In our experiments, we will investigate two supervision levels and two transfer strategies. In the first supervision setting, the affordances of example tools are pixel-wise annotated. In the second setting, the affordances of example tools are annotated by bounding boxes. In the later case, we obtain a rough pixel-wise annotation by setting all pixel labels inside an annotated bounding box to its affordance class. If a pixel is located inside multiple affordance bounding boxes, it receives the affordance label of the smallest bounding box. We refer to these supervision levels by “bbox” and “pixel-wise”. The “copy” strategy simple resizes and copies the labels of the example tool onto the query tool. The “warp” strategy warps the label of the example tool using the transformation predicted by the alignment network. For both transfer strategies, all pixels located outside the object bounding boxes are set to background and all pixel labels inside the object bounding boxes which were not assigned to an affordance class are ignored and thus do not contribute to the loss when training the semantic segmentation network. We combine the notations of supervision level and transfer strategy, for example“bbox-copy” means“bbox” supervision level and“copy” transfer method. 6.3 illustrates our supervision levels and transfer strategies. The proposed emthod assumes“bbox” for supervision and uses“copy” for label transfer. For semantic segmentation, we use the deeplab VGG architecture (*Chen et al.*, 2015), which is a fully convolutional network providing as output a feature map f with width and height equal to the input image and each channel corresponding to an affordance or background. We obtain the affordance probability by taking the pixel-wise softmax of f. During training, the loss for a particular pixel is computed individually, if the ground truth label is an affordance or background it equals the cross entropy between the ground truth label and the prediction, otherwise it is 0. The overall loss is the sum of the pixel-wise losses. During inference, we use the conditional random field layer of deeplab on top of the final feature map.

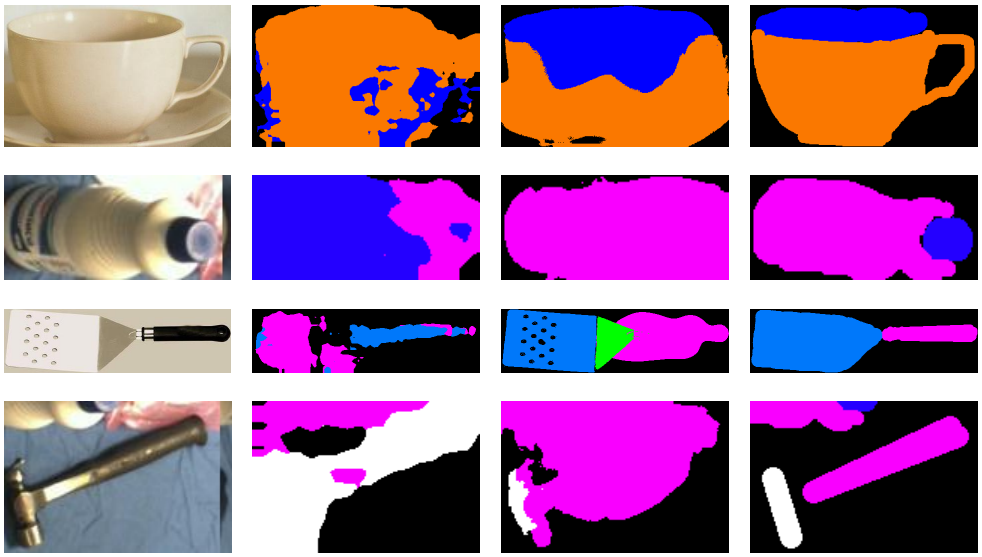


Figure 6.4: Qualitative results on bounding boxes: RGB input (first column), DCSP (*Chaudhry et al.*, 2017) results (second column), our results (third column), ground truth (last column). In contrast to DCSP, our method correctly associates the affordances with the respective object parts.

6.3 Experiments

We conduct our experiments on the IIT-AFF dataset introduced by (*Nguyen et al.*, 2017b). It consists of images showing 10 classes of tools in context, there are 6184 images in the trainval set and 2651 images in the test set. The images were collected in a robotics lab or come from the ImageNet dataset (*Russakovsky et al.*, 2015). Each tool is annotated with a bounding box. Additionally, each tool class has a predefined set of possible affordances. Tool parts serving an affordance are pixel-wise annotated with it. The tool classes with their affordances are: bowl (wrap, contain), tv (display), pan (contain, grasp), hammer (grasp, pound), knife (cut, grasp), cup (contain, wrap), drill (grasp, engine), racket (grasp, hit), spatula (support, grasp), bottle (grasp, contain).

Unless stated otherwise, in all our experiments our unlabeled set comprises the images from the IIT-AFF trainval set with 6 example tools per tool class randomly drawn from them to constitute the training set. We use the semantic alignment model trained on PF-PASCAL (*Ham et al.*, 2017) by (*Rocco et al.*, 2018). For training and inference with deeplab (*Chen et al.*, 2015), we use the same setup as in the original paper in the fully supervised setup on Pascal.

6.3.1 Comparison to state-of-the-art

To our knowledge there is no work on weakly supervised semantic segmentation which uses the same amount of supervision: A vast object dataset annotated on bounding box level but unlabeled in terms of object part affordances and a tiny dataset with bounding boxes provided for affordances. DCSP (*Chaudhry et al.*, 2017), which is the current state-of-the-art method for weakly supervised image segmentation on Pascal VOC 2012, uses a list of present classes in an image for supervision. We therefore train DCSP on the bounding boxes of tools from the unlabeled set as well as on the annotated bounding boxes of affordances from the training set. On the unlabeled set, the affordances

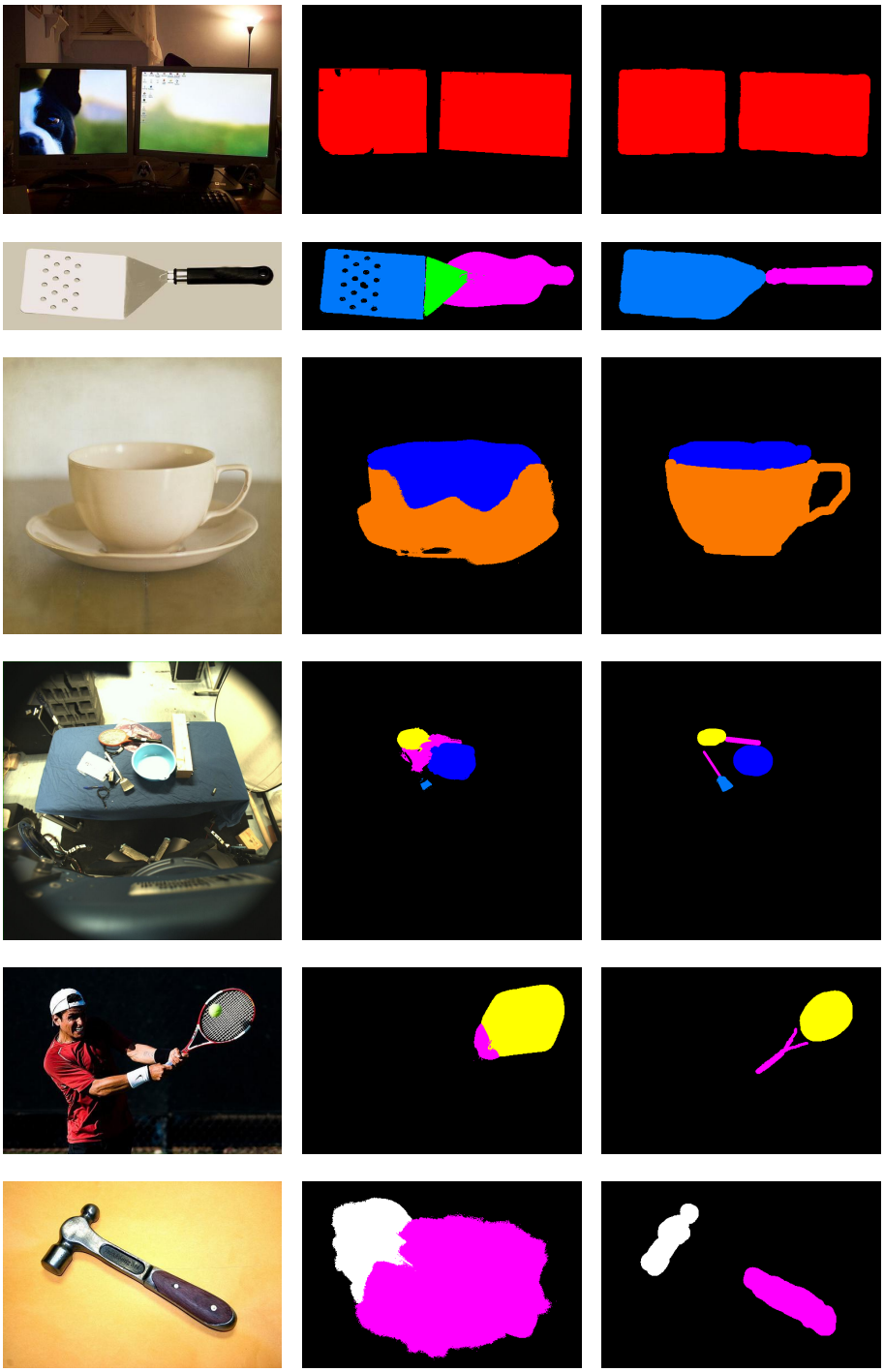


Figure 6.5: Qualitative results on IIT-AFF (*Nguyen et al.*, 2017b): RGB input (left), our results (middle), ground truth (right).

method	contain	cut	display	engine	grasp	hit	pound	support	wrap grasp	mean
DCSP (*Chaudhry et al.*, 2017)	0.340	0.179	0.602	0.214	0.259	0.548	0.242	0.085	0.205	0.297
proposed	0.616	0.209	0.811	0.364	0.328	0.633	0.345	0.335	0.510	0.461

Table 6.1: Comparison to DCSP (*Chaudhry et al.*, 2017), a method showing state-of-the-art results on Pascal VOC 2012. The metric used is intersection over union, evaluation on the IIT-AFF dataset (*Nguyen et al.*, 2017b). For a fair comparison, we train and evaluate DCSP on bounding box crops of tools and the affordance segments of example tools and evaluate ours on the bounding box crops only.

# examples	contain	cut	display	engine	grasp	hit	pound	support	wrap grasp	mean
1	0.480	0.152	0.760	0.305	0.293	0.575	0.101	0.134	0.387	0.354
2	0.518	0.191	0.744	0.336	0.267	0.590	0.248	0.096	0.426	0.380
3	0.522	0.182	0.716	0.331	0.269	0.645	0.245	0.136	0.417	0.385
6 (default)	0.564	0.180	0.723	0.329	0.288	0.596	0.295	0.323	0.469	0.419

Table 6.2: Evaluation of our method on the IIT-AFF dataset (*Nguyen et al.*, 2017b) for different number of example tools per tool class. We evaluate on full images and report IoU.

are inferred from the tool class and used as image labels for supervision. We keep the original training parameters of DCSP, but reduce the learning rate by a factor of 3 since it improved the results of DCSP.

For comparison, we evaluate both methods not on the entire images, but only within the annotated bounding boxes surrounding the objects, since we are interested in how well both methods segment the affordances within a bounding box. The results are reported in Table 6.1: DCSP achieves a mean IoU of 29.7% while our approach yields 46.1%, thus outperforming DCSP by more then 16%. To analyze if the difficulty for DCSP stems from the localization of affordances on the tool or from the pixel-wise segmentation of the tool itself, we performed an additional experiment. Instead of training DCSP for segmenting affordances, we trained DCSP for segmenting objects. For object segmentation, DCSP performs very well and achieves 53.4% mean IoU for the object categories. Therefore localizing the affordance on the tool constitutes the main challenge. This is also evident from the qualitative comparison shown in Fig. 6.4. Qualitative results for complete images are shown in Fig. 6.5.

6.3.2 Number of Examples

Our proposed evaluation setup uses 6 random examples per tool class, but we also investigated the performance with an even smaller amount of examples, namely 1, 2 and 3 examples per tool class, and report the results in Table 6.2. Unlike in the previous section, we evaluate on complete test images, but still achieve a mean IoU of 41.9%. When using only one example tool per tool class, the performance drops to 35.4%.

6.3.3 Impact of Additional Training Data

To see if additional training data and transferring the labels from the examples to the additional training data is required at all, we trained our semantic segmentation network only on the images containing at least one of the 60 example tools. Pixels located inside the affordance bounding box of

method	contain	cut	display	engine	grasp	hit	pound	support	wrap grasp	mean
example tools only	0.563	0.000	0.501	0.206	0.226	0.553	0.005	0.016	0.388	0.273
proposed	0.564	0.180	0.723	0.329	0.288	0.596	0.295	0.323	0.469	0.419

Table 6.3: Comparison of training on the example images only vs. our approach, which uses additional training data by label transfer. We evaluate on full images from the IIT-AFF dataset (*Nguyen et al.*, 2017b) and report IoU.

method	contain	cut	display	engine	grasp	hit	pound	support	wrap grasp	mean
"bbox-warped"	0.550	0.156	0.730	0.313	0.278	0.640	0.188	0.218	0.443	0.391
proposed	0.564	0.180	0.723	0.329	0.288	0.596	0.295	0.323	0.469	0.419

Table 6.4: Comparison of warping the affordance labels from the example tool vs copying them. We evaluate on full images from the IIT-AFF dataset (*Nguyen et al.*, 2017b) and report IoU.

the example tools were set to this affordance class, pixels belonging to any tool bounding box which does not belong to an example tool were ignored during training, and the rest was set to background. Since the number of training images is tiny in this setting, we reduced the number of iterations from 6000 to 300 and the step length during training accordingly to avoid overfitting. As can be seen in Table 6.3, the performance drops to 27.3% and for the affordances cut, pound, support to almost 0. Our approach is especially beneficial for challenging small affordances.

6.3.4 Warping vs No Warping

While we simply resize and copy the affordance localization cues from the most similar example tool to the tool of interest, one could also warp the localization cues of the example tool onto the target tool using the transformation provided by the semantic alignment network. On the one hand, this approach has the advantage of potentially better aligning the shape of the tools and therefore better aligning the functional parts. On the other hand, the warping might be reasonable for only some parts of the object but fail for other, in particular small, parts. Therefore, the benefit of using the warping transformation or not depends on the affordance classes. The results reported in Table 6.4 show that warping improves the accuracy for the classes display, engine, and hit, but it decreases the accuracy for the other affordance classes. In average, using the estimated warping transformation for label transfer reduces the accuracy from 41.9% to 39.1%.

6.3.5 Bounding Box vs. Pixel-wise Annotation

Obtaining affordance region bounding boxes for example tools is far cheaper than annotating the functional regions pixel-wise. To investigate the potential gain from a pixel-wise annotation, we conducted two ablation experiments. In the first, we transfer the pixel-wise affordance annotations from example tools to unlabeled tools without using the estimated warping transformation and in the second we use estimated warping transformation for label transfer. We report the results in Table 6.5: Providing pixel-wise affordance annotations for example tools increases the accuracy with and without warping. In case of warping the accuracy increases from 39.1% to 40.1% and without warping the accuracy increases from 41.9% to 44.8%.

supervision	contain	cut	display	engine	grasp	hit	pound	support	wrap grasp	mean
"pixel-wise copy"	0.601	0.245	0.745	0.368	0.388	0.589	0.260	0.333	0.502	0.448
"pixel-wise warp"	0.592	0.190	0.748	0.363	0.354	0.616	0.0	0.278	0.466	0.401
proposed	0.564	0.180	0.723	0.329	0.288	0.596	0.295	0.323	0.469	0.419

Table 6.5: Comparison of using accurate pixel-wise affordance annotations of the example tools vs. bounding boxes around affordances. We report the results with and without using the estimated warping transformation for label transfer. We evaluate on full images from the IIT-AFF dataset (*Nguyen et al.*, 2017b) and report IoU.

matching measure	contain	cut	display	engine	grasp	hit	pound	support	wrap grasp	mean
ResNet-101 features	0.573	0.206	0.705	0.348	0.287	0.608	0.262	0.322	0.423	0.415
proposed	0.564	0.180	0.723	0.329	0.288	0.596	0.295	0.323	0.469	0.419

Table 6.6: Comparison of two matching strategies between query tools and example tools: The proposed strategy uses the loss of a semantic alignment network trained in an unsupervised manner, the ablation uses the features of ResNet-101 pretrained on ImageNet. We evaluate on full images from the IIT-AFF dataset (*Nguyen et al.*, 2017b) and report IoU.

6.3.6 ResNet Features vs. Alignment

To investigate the benefit of the unsupervisedly trained semantical alignment network, we train a semantic segmentation model using an approach identical to the proposed method except for the matching criterion between query tools and example tools. Since the alignment network was trained on Pascal VOC 2012, we take the Pascal VOC 2012 semantic segmentation Resnet-101 model from (*Chen et al.*, 2018b), and generate the features of the res5c layer for each query tool and each example tool. Note that we use the same CNN backbone as for the semantical alignment network and require the same amount of cross dataset generalisation. After that, we retrieve for each query tool the example tool with the most similar feature map and transfer the labels of tha latter to the former. Specifically, the cosine distance serves as a measure for the similarity of the vectorized feature maps of two images v, w:

$$d = 1 - \frac{\langle v, w \rangle}{\|v\|\|w\|} \tag{6.4}$$

As can be seen from 6.6, the ResNet-101 features perform slightly worse than the weak alignment network.

6.3.7 Oracle Experiment: Ground Truth Bounding Box for Each Affordance of Each Query Tool

In this ablation experiment we investigate what is achievable if the bounding boxes around affordances are not only given for the example tools but also for all query tools. All pixels inside an affordance bounding box are set to this affordance. In case of a pixel belonging to multiple affordance bounding box, it is assigned to the affordance with the smallest bounding box. All other pixels are set to background. After that, the semantic segmentation network is trained and used for inference as in our proposed method. We report the results in 6.7. This additional supervision improves the results to 52.6%, however at the cost of additional annotations of query tools, while our method does not require any additional annotation once object bounding boxes are given. For example it

matching measure	contain	cut	display	engine	grasp	hit	pound	support	wrap grasp	mean
gt-bbox for each affordance	0.686	0.217	0.747	0.521	0.389	0.722	0.382	0.466	0.606	0.526
proposed	0.564	0.180	0.723	0.329	0.288	0.596	0.295	0.323	0.469	0.419

Table 6.7: Results for if ground truth bounding boxes would be given for each affordance of each query tool vs our method. We evaluate on full images from the IIT-AFF dataset (*Nguyen et al.*, 2017b) and report IoU.

method	Bird	Cat	Cow	Dog	Horse	Person	Sheep	mean
(*Krause et al.*, 2015)	0.099	0.135	0.115	0.141	0.067	0.106	0.105	0.110
(*Meng et al.*, 2017)	0.111	0.113	0.124	0.142	0.075	0.128	0.106	0.114
proposed	0.148	0.174	0.115	0.180	0.120	0.108	0.201	0.149

Table 6.8: Evaluation on the Pascal Parts (*Chen et al.*, 2014). Our method outperforms state-of-the-art methods for weakly supervised semantic parts segmentation. As on IIT-AFF dataset (*Nguyen et al.*, 2017b), we use 6 example objects per object class.

could be applied to the affordances of objects in the COCO dataset (*Lin et al.*, 2014). Even if the bounding boxes are not given for a custom dataset, they can be generated using a weakly supervised object detection system, *e.g.* (*Zhang et al.*, 2018b).

6.3.8 Evaluation on the Pascal Parts dataset

We finally evaluate our approach on the Pascal Parts dataset (*Chen et al.*, 2014). It contains images from the Pascal VOC 2012 dataset, which belong to the categories bird, cat, cow, dog, horse, person, and sheep. For each category, 4 to 5 semantic body parts are annotated. The task of part segmentation differs from affordance segmentation since different object classes do not share the same part category, *e.g.*, leg of horse and leg of sheep are considered as two different part classes in Pascal Parts. This is in contrast to affordances, which are shared among different tool classes. Since our method can be applied to both tasks, we also evaluate our approach on this dataset by randomly sampling 6 example objects per object class. Our approach outperforms the current state-of-the-art by +3.5% as can be seen in Table 6.8.

6.4 Conclusion

In this work, we have shown that it is possible to train a CNN for affordance or object part segmentation from very few annotated examples. This has been achieved by exploiting a semantic alignment network to transfer annotations from a small set of annotated examples to images that are only annotated by the object class. In our experiments, we have shown that our approach achieves state-of-the-art accuracy on the IIT-AFF dataset (*Nguyen et al.*, 2017b) and the Pascal Parts dataset (*Chen et al.*, 2014). A comparison between the method proposed in this Chapter and the one from Chapter 5 must be taken with a grain of salt, since the former deals with a multilabel problem while the later deals with a singlelabel problem. If we nevertheless make it, we see that the accuracy is roughly the same. In terms of annotation cost, the method from this chapter is far cheaper if we can assume that object level bounding boxes are available for free. If they are not, the annotation cost for

the method from Chapter 5 is significantly lower. *E.g.* for an object with 2 affordances, the method from the previous chapter requires 2 points, while the method from this chapter would require to draw an accurate bounding box around the object of interest.

CHAPTER 7

Harvesting Information from Captions for Weakly Supervised Semantic Segmentation

Contents

7.1 Introduction

In the previous Chapters, we showed how to train a model for object part affordance segmentation from cheaper supervision cues. However, these approaches either require an annotation effort proportional to the number of training images or assume object level bounding boxes to be given. In this Chapter, we will switch to the domain of object class segmentation to see how far we can get without spatial hints inside the image.

A possible source of image level supervision are image tags and they have already been explored in numerous works. A popular approach in this realm (*Kolesnikov and Lampert*, 2016; *Zhou et al.*, 2016; *Ahn and Kwak*, 2018) consists of estimating for each image in the training set so-called class activation maps (CAMs) (*Zhou et al.*, 2016), which indicate where certain semantic categories occur in an image. In a second step, pixel-wise class labels are extracted from the CAMs and a neural network for image segmentation is trained. These approaches, however, still assume curated image

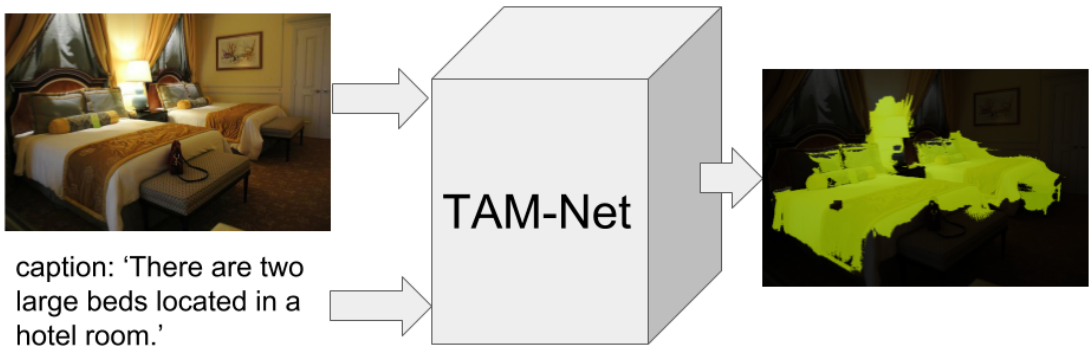


Figure 7.1: Given one or multiple captions per image in the training set, our network predicts text activation maps (TAMs) for each image which are then converted into class activation maps (CAMs) as illustrated in Figure 7.3. The text activation maps are more general than class activation maps since they localize compound concepts like "two large beds" as well as categories like "bed". The example shows the class activation map estimated for "bed" for this training example. Using the estimated CAMs of all training images, a standard convolutional neural network for semantic segmentation can then be trained.

tags, *i.e.* all classes of interest in the image must be tagged. This is not guaranteed for tags retrieved together with images from the Internet.

However, the cues we opt for are readily available textual descriptions. Such textual descriptions can be either from image captions or text surrounding or referring to an image in an article or blog. Such textual descriptions are more general than class tags and it is possible to reduce these textual descriptions to class tags by parsing the texts for the names of the categories of interest. However, textual descriptions are richer in the description of the image content than just class tags as illustrated in Figure 7.1. In case of class tags, the image would be only labeled by the relevant category "bed" and the only information we have is that the image contains at least one bed, but we do not know if the bed is large or small or if many instances are in the image. The caption "There are two large beds located in a hotel room" provides much more information. In particular, "two large beds" indicates that many pixels of the image should be labeled by the category "bed".

In this work, we therefore propose an approach that uses non curated image captions as weak supervision for training a convolutional neural network for semantic image segmentation. Our contribution focuses on the research question how class activation maps (CAMs) can be estimated from image captions and we show that these class activation maps are substantially more accurate than CAMs estimated from class tags. In order to estimate the class activation maps for the training images from captions, we learn a joint embedding of the visual representation of the image and the

textual representation of the caption as illustrated in Figure 7.3. Using such a joint embedding, our network predicts text activation maps (TAMs), which locate categories like "bed" as well compound concepts like "two large beds". For each class, the TAM of the class as well as the TAMs of the compounds which contain the class name are fused to generate the CAM for the class.

We provide a thorough ablation study that analyses the benefit of the additional textual context that is provided by the compound concepts. On the COCO dataset (*Lin et al.*, 2014), the proposed approach outperforms the current state-of-the-art in weakly supervised semantic segmentation.

7.2 Method

Generating class activation maps (*Zhou et al.*, 2016) on training images is a crucial intermediate step for the majority of current state-of-the-art weakly supervised image segmentation methods. In our work, we therefore focus on improving them. To this end, we harvest information from image captions instead of relying on class tags only. Our TAM-network is a multimodal architecture, which maps image pixels and text snippets into a common semantic space which allows us to calculate activation maps for class names as well as compound concepts that include the class name as illustrated in Figure 7.3. From TAMs of class names and class relevant compound concepts, we obtain class activation maps. From these CAMs, we directly estimate the pixel-wise class labels as described in Section 7.2.3.

We finally train the widely used VGG16-deeplab model (*Chen et al.*, 2018b) for semantic segmentation on the estimated labels which yields us the final segmentation model.

7.2.1 Parsing Captions

In our work, we distinguish three different types of text snippets: *Class names* contain the names of the classes of interest in the particular dataset as well as their plural form. *Compound concepts* are snippets of a sentence between beginning of sentence, prepositions, verbs and end of sentence. These snippets have to contain multiple words excluding articles. Essentially these are combinations of numbers, adjectives, adverbs and nouns like *two completely black dogs*. They are split into two categories: *Class related compound concepts* contain a class name inside them and *class unrelated compound concepts* do not. All of them are used during training of the TAM network, but the type of the snippet determines its weight in the loss. For CAM inference, we use class names and class related compound concepts. We use 300-dimensional word to vec (w2v) (*Mikolov et al.*, 2013) embeddings to convert text snippets to numerical vectors of equal length. The embedding of a single word is given by the word to vec dictionary. For text snippets containing multiple words, we take the arithmetical mean of the normalized embeddings of individual words. These embeddings are used as input to the textual path of the TAM-network.

We use the classes mentioned in the captions to determine what classes are present in a training image. If the class name is present in at least one of the image captions, the class is considered as present, otherwise not. Note that in contrast to curated image tags, the captions do not necessarily contain all classes that are present in an image.

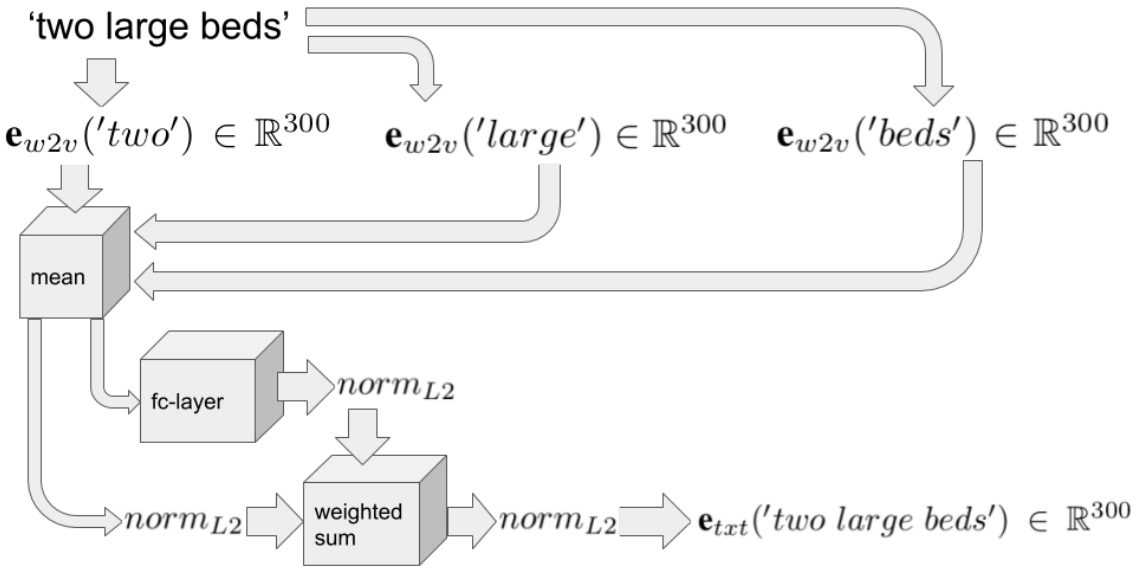


Figure 7.2: To obtain the textual embedding of an arbitrary snippet of text, we first encode each word with a Word2Vec model and average over words, which gives us the input embedding. Feeding it into a single fully connected layer with a residual connection yields the output embedding

7.2.2 Multi-Modal TAM Network

Our TAM network comprises a visual path and a textual path which map visual and textual information into a common 300-dimensional semantic embedding space.

Visual Path. Our visual embedding path maps an image I with X pixels to a pixel-wise visual embedding $\mathbf{E}_{vis}(I) \in \mathbb{R}^{X \times 300}$. It is a modification of the VGG16 architectures, but we will also report results for a ResNet38 architecture. In both cases, we change the output dimension of the last fully connected layer to the dimension of the common semantic embedding space.

Textual path. As shown in Figure 7.2, our textual path first obtains the word to vec embedding $\mathbf{e}_{w2v}(t) \in \mathbb{R}^{300}$ of a text snippet t by taking the average of the word to vec embeddings of the single words. Then $\mathbf{e}_{w2v}(t) \in \mathbb{R}^{300}$ is mapped to a vector $\mathbf{e}_{txt}(t) \in \mathbb{R}^{300}$ in the common semantic embedding space via:

$$\begin{aligned}\mathbf{e}_{txt}(t) =& norm_{L2}(norm_{L2}(\mathbf{e}_{w2v}(t)) \\ &+ w_{res} norm_{L2}(\mathbf{M}_{txt}\mathbf{e}_{w2v}(t)))\end{aligned} \tag{7.1}$$

where $norm_{L2}()$ denotes normalization by the L_2 norm, $\mathbf{M}_{txt} \in \mathbb{R}^{300 \times 300}$ is the weight matrix of a fully connected layer and $w_{res} \in \mathbb{R}$ is a hyperparameter.

We also investigated RNNs that take the word to vec embeddings of the individual words as input and output $\mathbf{e}_{txt}$, but they performed slightly worse than our approach. This is probably due to short length of our text snippets which contain mostly 2 or 3 words.

Textual Activation Map. Given an image I with X pixels and a text snippet t, we generate the textual activation map (TAM), which we denote by $\mathbf{S}(I, t) \in \mathbb{R}^X$, from the visual embedding

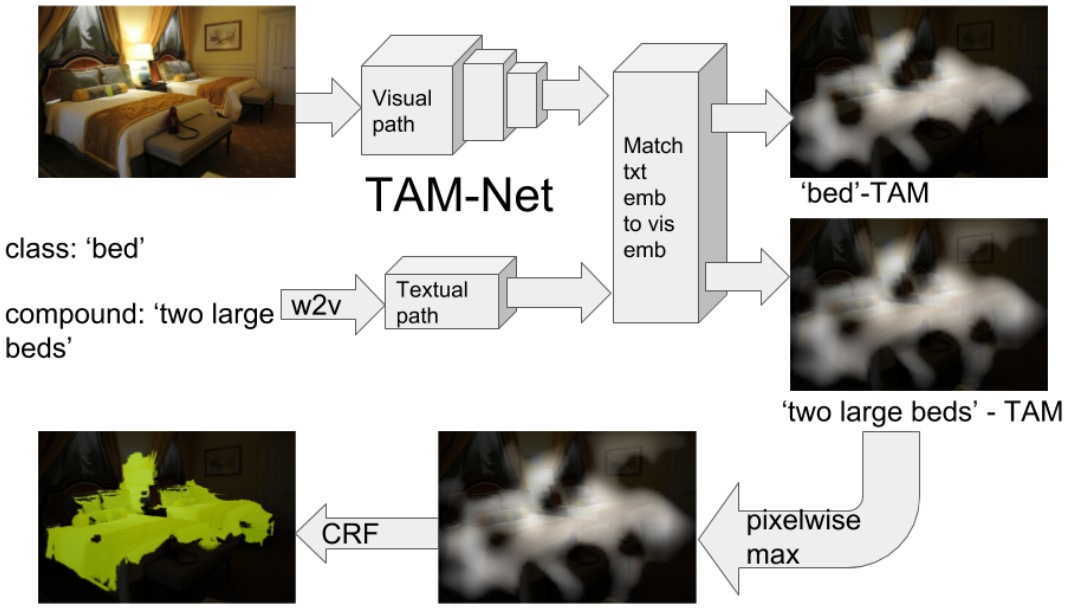


Figure 7.3: For each present class, we locate the class name as well as all compound concepts related to this class in the image (estimate their TAMs). We then normalize these TAMs and take for each class the pixel-wise maximum over them to arive at the CAM. Finally, we estimate pixel-wise class labels from these.

$\mathbf{E}_{vis}(I) \in \mathbb{R}^{X \times 300}$ and the textual embedding $\mathbf{e}_{txt}(t) \in \mathbb{R}^{300}$ by:

$$\mathbf{S}(I,t) = \mathbf{E}_{vis}(I)\mathbf{e}_{txt}(t). \tag{7.2}$$

To obtain the normalized TAM, we apply $relu$ to discard negative values and normalize it by

$$\mathbf{S}_{norm}(I,t) = \frac{\sqrt{relu(\mathbf{S}(I,t))}}{\max\limits_{x \in pixels} \sqrt{relu(S(I,t,x))}}. \tag{7.3}$$

7.2.3 Direct Estimation of pixel-wise Class Labels from TAMs

Since we aim to learn a model for semantic segmentation, we need to estimate the pixel-wise class labels for each training image I. To obtain these, we first calculate normalized class activation maps for all present classes as shown in Figure 7.3. To this end, for each class c mentioned in the captions of image I, we collect a set of text snippets $\Phi(c)$ which comprise the class name and all compound concepts related to it. Then we combine the normalized TAMs for all $t \in \Phi(c)$ into a normalized CAM $\mathcal{S}_{norm}(I,c) \in \mathbb{R}^X$ for class c by taking the pixel-wise maximum over the TAMs:

$$\mathcal{S}_{norm}(I,c,x) = \max_{t \in \Phi(c)} \{S_{norm}(I,t,x)\} \tag{7.4}$$

We obtain the background activation map $\mathbf{b}(I)$ as in (*Ahn and Kwak*, 2018) by

$$b(I,x) = (1 - \max_{c \in C(I)} \{\mathcal{S}_{norm}(I,c,x)\})^{\alpha} \tag{7.5}$$

where $C(I)$ are the classes present in image I. We keep $\alpha = 4$ which is the value suggested in (*Ahn and Kwak*, 2018). We then finally refine the normalized CAMs with a CRF (*Krahenbuhl and Koltun*, 2011) to estimate pixel-wise class labels.

7.2.4 Training of Embedding Architecture

We finally describe how we train our network. For training, we propose the following loss

$$L = \lambda_{cls} L_{cls} + \lambda_{cpt} L_{cpt} + \lambda_{ac} L_{ac}. \tag{7.6}$$

The class loss L_{cls} and concepts loss L_{cpt} ensure that the textual activation maps have high values for classes and compounds that are present in a training image and low values if they are not present. The auto consistency loss L_{ac} ensures that the TAMs are geometric consistent if an image is vertically flipped as illustrated in Figure 7.4.

Class Loss. For each class c, we apply average pooling over the pixels of the TAM for its class name t_c, *i.e.*, $S_{pool}(I, t_c) = \frac{1}{X} \sum_p \mathbf{S}(I, t_c)$. These values can be seen as logits for the probability of this class to be present in the image. We use average pooling as in (*Huang et al.*, 2018; *Chaudhry et al.*, 2017; *Ahn and Kwak*, 2018) since it typically leads to activation maps that cover the complete class and not only its most distinctive areas as it is the case for max pooling.

The class loss L_{cls} is then given by the multilabel binary cross entropy loss:

$$\begin{aligned} L_{cls} = &- \sum_{c \in C(I)} log\left(\frac{1}{1 + e^{-S_{pool}(I,t_c)}}\right) \\ &- \sum_{c \notin C(I)} log\left(\frac{e^{-S_{pool}(I,t_c)}}{1 + e^{-S_{pool}(I,t_c)}}\right). \end{aligned} \tag{7.7}$$

The loss is minimized if the TAMs for the present classes show a strong signal while the TAMs of the classes that are not present in the image are close to zero.

Concepts Loss. The concepts loss is calculated in the same way as the class loss, except that we use the TAMs $\mathbf{S}(I, t)$ of compound concepts instead of using TAMs of class names. As for the class loss, we apply average pooling $S_{pool}(I, t) = \frac{1}{X} \sum_p \mathbf{S}(I, t)$. While we use the multilabel binary cross entropy loss as for the class loss, we have to subsample the missing concepts since there are otherwise too many. The concepts loss is then given by

$$\begin{aligned} L_{cpt} = &- \sum_{t \in Comp(I)} log\left(\frac{1}{1 + e^{-S_{pool}(I,t)}}\right) \\ &- \sum_{t \in ContrComp(I)} log\left(\frac{e^{-S_{pool}(I,t)}}{1 + e^{-S_{pool}(I,t)}}\right) \end{aligned} \tag{7.8}$$

where $Comp(I)$ are the compound concepts present in image I and $ContrComp(I)$ are contrastive compound concepts that are randomly sampled from other images.

Auto-Consistency Loss. The purpose of the auto-consistency loss is to ensure that geometric transformations of the image do not alter the accordingly transformed TAMs as illustrated in Figure 7.4. We use the image I and the flipped image $I_{flipped}$ for training and obtain the corresponding TAMs $\mathbf{S}(I, t_c)$ and $\mathbf{S}(I_{flipped}, t_c)$ for all classes. Note that there is no TAM for the background class,

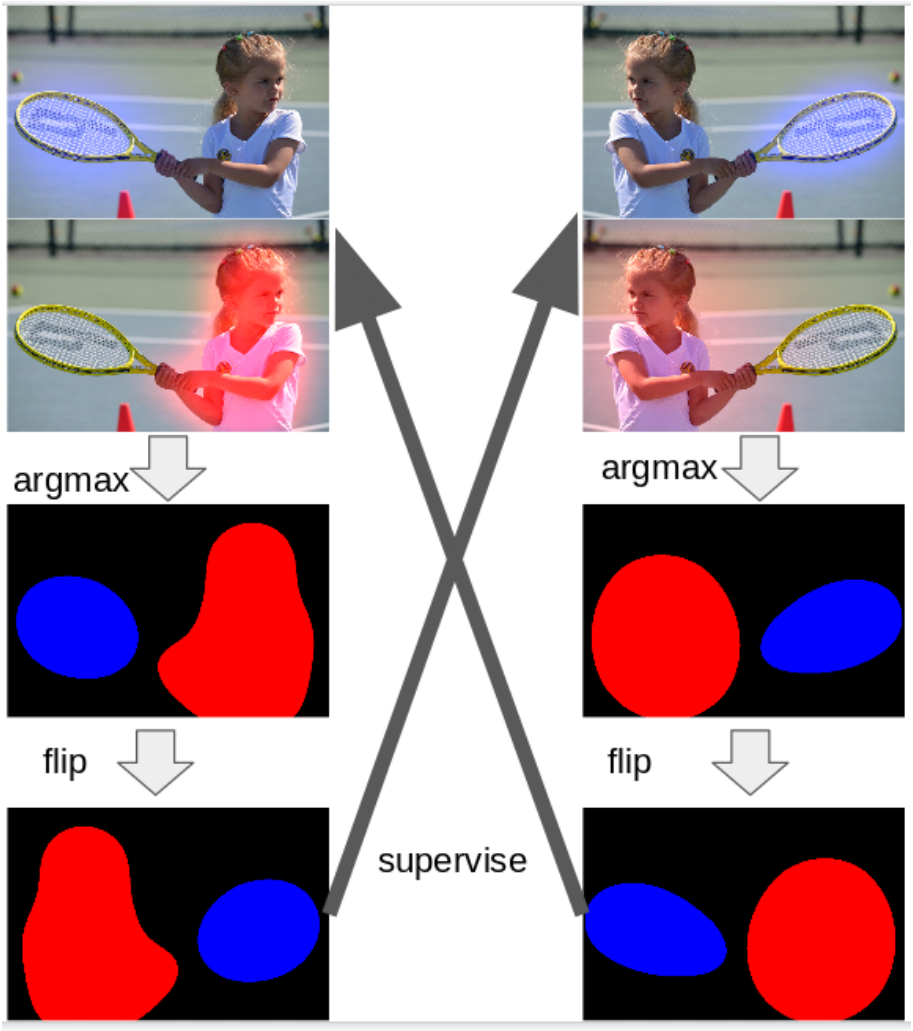


Figure 7.4: The auto consistency loss enforces invariance under geometric transformations like flipping. To this end, labels are estimated in an online manner from the TAMs of class names for the image and the flipped image. Then these are flipped and the estimated labels of the image are used to supervise the embedding of the flipped image and vice versa.

we therefore set $\mathbf{S}(I, t_{bg}) = 0$ for the background class. We convert them into pixel-wise class probabilities $\mathbf{P}$ and $\mathbf{P}_f$ by applying the softmax for each pixel x, *i.e.*, $\mathbf{P}(x, c) = \text{softmax}_{c'}(\mathbf{S}(I, t_{c'}, x))$.

We also compute a pixel-wise labeling for $\mathbf{S}(I, t_c)$ and $\mathbf{S}(I_{flipped}, t_c)$ as in Section 7.2.3 but without CRF. Instead, we simply use the class with the highest activation per pixel

$$\hat{c}(I, x) = \underset{c \in \{C(I) \cup bg\}}{argmax} \; S_{norm}(I, t_c, x) \tag{7.9}$$

where the background activation is estimated as in (7.5). We finally mirror the pixel-wise segmentations as shown in Figure 7.4. The auto-consistency loss L_{ac} is then given by

$$\begin{aligned} L_{ac}(I) = &- \frac{1}{2} \sum_{x \in pixel} log\left(\frac{1}{1 + e^{-P_{flipped}(x, \hat{c}(I,x))}} \right) \\ &- \frac{1}{2} \sum_{x \in pixel} log\left(\frac{1}{1 + e^{-P(x, \hat{c}_{flipped}(I,x))}} \right). \end{aligned} \tag{7.10}$$

7.3 Experiments

For our experiments, we use the COCO dataset (*Lin et al.*, 2014), since it provides several captions for each image and instance level segmentations for 80 object classes which we convert to class level segmentations. As train set we use the COCO train2014 split containing 83k images. To evaluate the final semantic segmentation models, we use the COCO val2014 split containing 40k images as our test set. Our evaluation metric is intersection over union averaged over 81 classes (80 object classes and background).

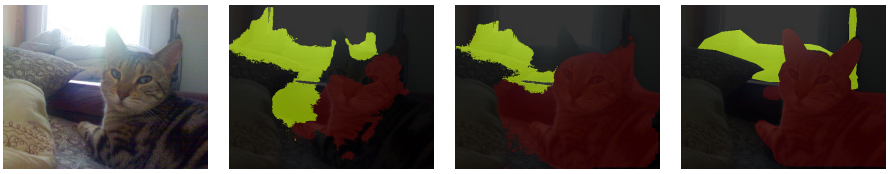
Implementation Details. The visual paths of our VGG16 and ResNet38 architecture is identical to the respective architectures from (*Ahn and Kwak*, 2018) up to the last layer. While we use mostly the VGG architecture for our experiments, we show some results using the ResNet at the end. We train the VGG architecture as well as the ResNet architecture for 15 epochs. For VGG, the learning rate is 0.1 for weights and 0.2 for biases, for ResNet it is 0.01 and 0.02, respectively. For VGG the batch size is 16, for ResNet it is 8. Weight decay equals 0.0005 for both architectures. The first two convolutional blocks are not finetuned at all and for the fc8 layer and the textual path, we scale up the learning rate by 10. During training, the learning rate decays to 0 according to the polynomial policy. The data augmentation techniques include random scaling, cropping and mirroring. We set $\lambda_{cls} = 1.0$, $\lambda_{cpt} = 0.3$ and $\lambda_{ac} = 0.001$ so that each loss term is roughly in the same order of magnitude.

For the concepts loss (7.8), we sample a maximum of 10 compound concepts mentioned in the captions of an image. To collect contrastive compound concepts which are absent in the image, we first sample 10 random images and extract the compound concepts from their captions. Then we randomly sample 50 of these concepts.

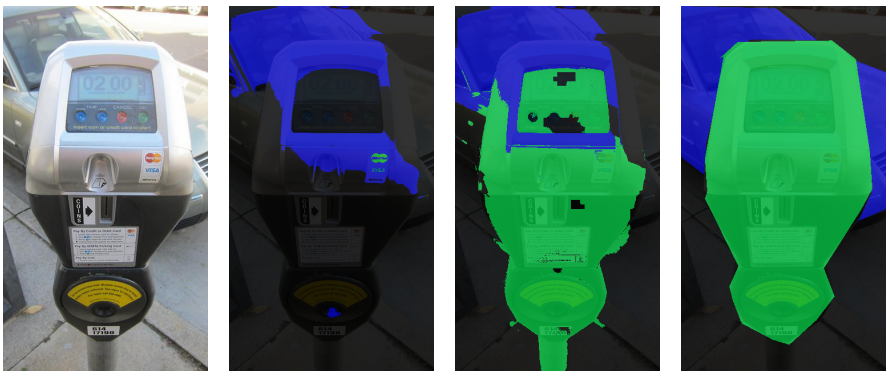
For semantic segmentation we use the baseline VGG16 deeplab model (*Chen et al.*, 2018b). We keep the hyperparameters but increase the number of iterations by a factor of 3 to account for the bigger size of COCO as compared to Pascal VOC 2012.

A **dog** laying on a **surf board** and riding a small wave

A **cat** laying on top of a **bed** next to a window.



A high-tech **parking meter** in front of a parked **car**.



A computer desk, with a **laptop** on a stand, a **keyboard**, and a **mouse**.

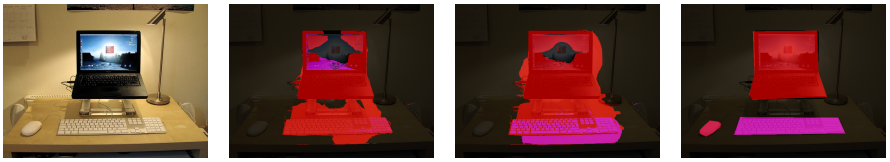


Figure 7.5: Examples of estimated pixel-wise class labels on the training set. From left to right: Image, baseline, proposed method, ground truth. Above each image we provide the caption and highlight the class names of the COCO classes.

Results on training set							
method	TAM-Net	L_{cls}	L_{cpt}	L_{ac}	precision	recall	IoU
baseline class tags from captions	VGG16				0.370	0.207	0.146
baseline ground-truth class tags	VGG16				0.337	0.230	0.158
prop. L_{cls}	VGG16	$\surd$			0.378	0.205	0.144
prop. L_{cpt}	VGG16		$\surd$		0.288	0.383	0.185
prop. L_{cls} + L_{cpt}	VGG16	$\surd$	$\surd$		0.382	0.316	0.199
prop. L_{cls} + L_{cpt} + L_{ac}	VGG16	$\surd$	$\surd$	$\surd$	0.383	0.329	0.203
prop. L_{cls} + L_{cpt} + L_{ac}	ResNet38	$\surd$	$\surd$	$\surd$	0.468	0.509	0.305
VisGround (*Engilberge et al.*, 2018)	ResNet151				0.375	0.432	0.231

Table 7.1: Results for estimated pixel labels on the training set.

7.3.1 Evaluation of System Components

For our ablation studies, we first evaluate the accuracy of the pixel labels that are estimated on the training images from the captions as described in Section 7.2.3.

Simply using a one-hot encoding of the classes retrieved from captions instead of the textual path gives an IoU of 14.6% as can be seen in Table 7.1. If we use only the class loss L_{cls}, the accuracy is similar to the baseline. Using the concepts loss L_{cpt}, however, improves mean IoU substantially to 18.5%. While the recall increases, the precision decreases. This is expected since the compound concepts provide mainly the textual context and are less class focused. Using both loss functions, alleviates this effect and improves the mean IoU to 19.9%.

Including auto consistency during training leads to further improvement from 19.9% to 20.3%. We show some qualitative results in Figure 7.5. If we use a ResNet38 architecture instead of a VGG16 architecture, our results improve to 30.5% IoU.

7.3.2 Comparison to Visual Grounding

We also compare our approach for generating CAMs with the state-of-the-art approach for weakly supervised visual grounding (*Engilberge et al.*, 2018). The authors use complete image captions from the COCO dataset as supervision for training. During inference, this model receives an image and a single text snippet referring to an object or to stuff in this image and returns an activation map for this snippet. To segment the image region the snippet refers to, the authors suggest to threshold the activation map with the average of the minimum and maximum value of this map. We can not use this method out of the shelf since locating multiple text snippets simultaneously is not intended for visual grounding and there is no policy to select the label of a pixel if it is assigned to multiple segments. To adapt the model to our setting and estimate pixel-wise class labels for the training images, we first generate the activation maps for the classes retrieved from the captions. Then, we subtract from each activation map the average of its minimum and maximum value. Finally, we set the activations for background to 0 and apply argmax over the classes. This yields 23.1% IoU as shown in Table 7.1. This is better than our results for the VGG16 architecture but far inferior to our ResNet38 architecture, although the visual grounding model uses a deeper ResNet151 architecture.

Impact of different types of concepts							
type of concepts	cls. rel. comp. cpt.	other comp. cpt.	no adj.	sgl. adj+nouns	precision	recall	IoU
cls. rel. only	√				0.404	0.306	0.199
all comp. conc.	√	√			0.382	0.316	0.199
all concepts	√	√		√	0.383	0.306	0.193
no adjectives	√	√	√		0.380	0.283	0.183

Table 7.2: Using compound concepts only for training leads to best results. Results are reported without auto-consistency loss.

Performance dependence on w_{res}			
size of w_{res}	precision	recall	IoU
0.0	0.395	0.287	0.190
0.2 (prop.)	0.382	0.316	0.199
0.4	0.277	0.494	0.201
0.6	0.255	0.523	0.194

Table 7.3: Small values of w_{res} allow the textual path to adjust the w2v embeddings to the needs of visual grounding, while large values lead to degeneration during training and inferior performance. Results are reported without auto-consistency loss.

7.3.3 Evaluating Concept Types

We compare the performance of our system when using different types of sentence snippets during training and report the results in Table 7.2. In our proposed method, all compound concepts are fed into the concept loss. Conceptually, adjectives provide additional cues during class activation map calculation. If we discard the adjectives, the accuracy decreases as shown in Table 7.2. Using only compounds containing COCO class names increases the precision but reduces the recall, the IoU remains unchanged. We therefore use all compound concepts from the captions in all other experiments. Feeding all nouns and adjectives additionally to the compounds into the loss decreases the performance. This shows that the compound concepts better encapsulate the context of the captions than single words.

7.3.4 Evaluating Textual Embedding

The hyperparameter w_{res} (7.1) controls the extent to which w2v embeddings (*Mikolov et al.*, 2013) are adjusted in the textual path. Intuitively high values add flexibility to the network and allow the textual path to adjust the w2v embeddings to the task at hand. Table 7.3 reports the results without the auto-consistency loss. As can be seen, increasing w_{res} from 0 improves the performance with a peak at 0.4. However, for higher values, the performance decreases again. This is because higher flexibility allows the network to find degenerate solutions: If w_{res} becomes too high, the textual path maps all class names and compounds appearing frequently to one vector v and all other class names and compounds to the negative vector $-v$. The visual path then maps everything to v.

Performance dependence on λ_{cpt}			
size of λ_{cpt}	precision	recall	IoU
0.1	0.362	0.301	0.186
0.3 (prop.)	0.382	0.316	0.199
0.5	0.396	0.319	0.204

Table 7.4: Performance grows when the impact of compound concepts is increased. Results are reported without auto-consistency loss.

7.3.5 Evaluating Concept Loss Weight

In Table 7.4, we also evaluate the impact of the parameter λ_{cpt} which weights the concept loss in (7.6). As in the previous experiment, we do not use the auto consistency loss. Even a small concept loss already improves the baseline approach giving 18.6%. By further increasing the weight of the concepts to 0.5, the performances raises to 20.4%.

7.3.6 Comparison to Ground-Truth Image Tags

The class tags we get from the captions are not perfect as can be seen from Table 7.5. Although for some classes the recall is very low, using parsed tags instead of clean tags does not harm the weakly supervised segmentation performance significantly. Training the baseline model with clean tags instead of retrieved tags improves the mean IoU from 14.6% to 15.8% as shown in Table 7.1. Surprisingly, the precision of CAMs obtained with clean tags is even smaller than with retrieved tags. It seems that COCO object classes not mentioned in the captions are more difficult to locate since captions only mention the most important objects which tend to be large. If an approach for weakly supervised image segmentation fails to locate an object that is expected to be present in an image, the precision decreases. It is remarkable that the gain from captions is substantially higher than the gain from ground-truth class tags, which shows that the captions provide more information than just tags.

7.3.7 Results on Test Set

To demonstrate the effect of improved CAMs on the final segmentation results, we train deeplab (*Chen et al.*, 2018b) for semantic segmentation on the estimated pixel-wise labels and evaluate its performance on the test set. We first evaluate the impact of the CRF and compare the results obtained by the baseline approach with ground truth class tags to our proposed approach using VGG16 and ResNet38.

The results reported in Table 7.6 indicate that the quality of the estimated labels is a strong predictor for the quality of the final segmentation. The performance gap of 4.5% on the train set between the baseline and the proposed method results in 4.9% on the test set. Applying a CRF on top of the test set prediction gives a slight improvement of 0.6%. We use a CRF in all remaining experiments. If we use a ResNet38 architecture for our TAM network, we obtain 28.5%.

We also compare our method to deep seeded region growing (DSRG) (*Huang et al.*, 2018), which is a state-of-the-art weakly supervised semantic segmentation method and it achieves 26.0% IoU on COCO. To train their model, the authors take CAMs and background cues from a strongly supervised

Class tag retrieval precision and recall		
class type	precision	recall
person and accessory	95.0%	33.0%
vehicles	89.5%	63.6%
outdoor	95.4%	56.0%
animal	87.1%	92.3%
sport	96.2%	64.2%
kitchenware	89.4%	20.3%
food	85.9%	68.7%
furniture	90.3%	44.0%
electronics	92.3%	48.3%
appliance	79.8%	46.0%
indoor	97.0%	43.7%

Table 7.5: Recall and precision of image class tags retrieved from image captions.

Results on the test set.		
method	TAM-Net	IoU test set
Baseline gt class tags no CRF	VGG16	0.161
Proposed no CRF	VGG16	0.210
Proposed	VGG16	0.216
Proposed	ResNet38	0.285

Table 7.6: Results of the final semantic segmentation model on the test set. The gain in accuracy of estimated pixel labels on the train set transfers well to the test set.

Comparison to DSRG(*Huang et al.*, 2018)	
method	IoU test set
DSRG(*Huang et al.*, 2018)	0.260
Proposed (VGG16)	0.216
Proposed (VGG16) + DSRG (*Huang et al.*, 2018)	0.269
Proposed (ResNet38)	**0.285**
Proposed (ResNet38)+DSRG (*Huang et al.*, 2018)	0.277

Table 7.7: DSRG(*Huang et al.*, 2018) is a saliency based approach for weakly supervised semantic segmentation. It can be combined with our approach.

Comparison to the state-of-the-art	
method	IoU test set
BFBP (*Saleh et al.*, 2016)	0.204
SEC (*Kolesnikov and Lampert*, 2016)	0.224
DSRG (*Huang et al.*, 2018)	0.260
VisGround (*Engilberge et al.*, 2018) adapted	0.275
Proposed	**0.285**
(*Wang et al.*, 2020)*	0.277
(*Li et al.*, 2020)*	0.263
(*Yao and Gong*, 2020)*	0.336

Table 7.8: Comparison of the final semantic segmentation model with the state-of-the-art on the test set. * indicate results published after completion and acceptance of our work.

saliency model as input. We can therefore combine this approach with our method by feeding our generated CAMs to DSRG. We report the results in Table 7.7. For VGG16, DSRG improves IoU from 21.6% to 26.9%. This is also a better result than the number reported by the authors demonstrating the benefit of our CAMs. For ResNet38, however, DSRG decreases the accuracy of our approach. A possible explanation is that DSRG reduces the information from CAMs by estimating high confidence regions first and expands them using the saliency maps. If the CAMs are inaccurate, DSRG substantially improves the results. However, if the CAMs become more accurate, DSRG discards too much information from the CAMs.

We finally compare our approach with other approaches for weakly supervised semantic segmentation. As can be seen from Table 7.8, our approach outperforms the other weakly supervised semantic segmentation models as reported by (*Huang et al.*, 2018). We also include our adapted version of the visual grounding approach of (*Engilberge et al.*, 2018). Interestingly, this approach also achieves a higher IoU than (*Huang et al.*, 2018), which shows the rich information that is present in image captions. Nevertheless, our approach achieves the higher IoU despite of using a ResNet38 instead of a deeper ResNet151. Thus, at moment of publication, our method was the state-of-the-art and has only been outperformed by (*Yao and Gong*, 2020) since then.

7.4 Conclusion

We presented an approach that uses image captions as supervision for weakly supervised semantic image segmentation. Inspired by weakly supervised approaches that estimate class activation maps from class tags and deduce localization cues from them, our approach estimates text activation maps for the class names as well as compound concepts and fuses them to obtain better class activation maps. We evaluated our approach on the COCO dataset and demonstrated that our approach outperforms the state-of-the-art for weakly supervised image segmentation at the moment of publication of this work. Although overall in our experiments image captions turned out to be better than image tags in terms of accuracy as well as cost, they must be used with caution. First, some object classes do not attract human attention and are often not mentioned in captions, see Table 7.5. And second, image tags do not help to distinguish coocurring classes like different object parts. Figure 6.4 demonstrates such failure cases for a method based on image tags. Image captions are prone to the same problem if they do not mention attributes of the coocurring classes that would allow to distinguish them.

CHAPTER 8

Discovering Latent Classes for Semi-Supervised Semantic Segmentation

8.1 Introduction

The methods we proposed in the first part of the thesis perform well in certain domains but still exhibit fundamental limitations. The sparse spatial cues used in Chapters 4, 5 and 6 convey the location and rough extent of the segments but will miss tiny details if these details do not correlate with some heuristic like color contrast. In Chapter 7 we used image captions as supervision cues. However, they only work for object classes typically attracting human attention. Curated image tags are in principle useless for constantly present classes, like the road in autonomous driving datasets. Therefore in general, accurate pixel wise labels must be present in at least a small number of images. This Chapter deals with a setup where for a small fraction of data, complete pixel-wise labels are given. The larger part of the dataset has no labels of any kind at all.

Before us, (*Hung et al.*, 2018) already addressed this setup. On labeled data, the authors train a discriminator network that distinguishes segmentation predictions and ground-truth annotations. On unlabeled data, they use the discriminator to obtain two kinds of supervision signals. First, they use the adversarial loss to enforce realism in the predictions. And second, they use the discriminator to locate regions of sufficient realism in the prediction. These regions are then annotated by the semantic class with the highest probability. The network for semantic segmentation is then trained on the labeled images and the estimated regions of the unlabeled images. Although the approach reported impressive results for semi-supervised segmentation, it does not leverage the entire information which is present in the unlabeled images since it discards large parts of the images.

In this work, we propose an approach for the semi-supervised semantic segmentation that does not discard any information. Our key observation is that the difficulty of the semantic segmentation task depends on the definition of the semantic classes. This means that the task can be simplified if some classes are grouped together or if the classes are defined in a different way, which is more consistent with the similarity of the instances in the feature space. We, therefore, do not focus on regions where the semantic classes can be detected with high confidence and instead propose to learn latent classes that can be reliably inferred for the entire unlabeled training image as illustrated in Figure 8.1.

Our network consists of two branches and is trained on labeled and unlabeled images jointly in the end-to-end fashion as illustrated in Figure 8.2. While the semantic branch learns to infer the given semantic classes, the latent branch learns to predict the most helpful latent classes for the semi-supervised learning by itself. The number of latent and semantic classes can differ and we use the conditional entropy to enforce the consistency between them. This means that we aim to minimize the variety of semantic classes that are assigned to a particular latent class. We also introduce a second

(a) Image (b) Latent Classes (c) Semantic Classes

Figure 8.1: Our network learns not only semantic but also latent classes that are easier to predict. The figure shows an example of latent and semantic class segmentation predictions for an image that is not part of the training data. As it can be seen, the learned latent classes are very intuitive, since the vehicles are grouped into one latent class and difficult-to-segment objects like pedestrians, bicycles, and signs are grouped into another latent class

loss that ensures that the inferred semantic classes on the unlabeled images are consistent with the inferred latent classes. Using this loss, we employ the latent classes as additional supervision for the semantic branch.

The idea to learn an easier auxiliary task as an intermediate step was exploited in the area of domain adaptation for semantic segmentation (*Zhang et al.*, 2017; *Dai et al.*, 2019; *Kurmi et al.*, 2019; *Sakaridis et al.*, 2019; *Lian et al.*, 2019). However, to our knowledge, we are the first to use simpler auxiliary tasks as an intermediate step in semi-supervised semantic segmentation.

The idea to discover clusters in data with latent classes to facilitate learning was already used in object detection (*Razavi et al.*, 2012; *Zhu et al.*, 2014a), joint object detection and pose estimation (*Li et al.*, 2016) and weakly-supervised video segmentation (*Richard et al.*, 2017). However, apart from addressing a different vision task, these approaches discover subcategories of classes while we look for suitable supercategories.

We demonstrate that our model achieves state-of-the-art results on PASCAL VOC 2012 (*Everingham et al.*, 2014) and Cityscapes (*Cordts et al.*, 2016). Moreover, our proposed branched architecture of the segmentation network results in the increased interpretability of the results. We show that the latent classes predicted by the latent branch correspond to supercategories of semantic classes as illustrated in Figure 8.1. Additionally, we show that the learned latent classes are superior to manually defined supercategories.

8.2 Method

An overview of our method is given in Figure 8.2. Our proposed model is a two-branch network. While the semantic branch serves to solve the final task, the purpose of the latent branch is to learn to group the semantic classes into latent classes in a data driven way as fine-grained as possible. While the fraction of annotated data is not sufficient to produce good results for the task of semantic segmentation, it is enough to learn the prediction of latent classes reasonably well, since this task is easier. Thus, the predictions of the latent branch can then serve as a supervision signal for the semantic branch on unlabeled data.

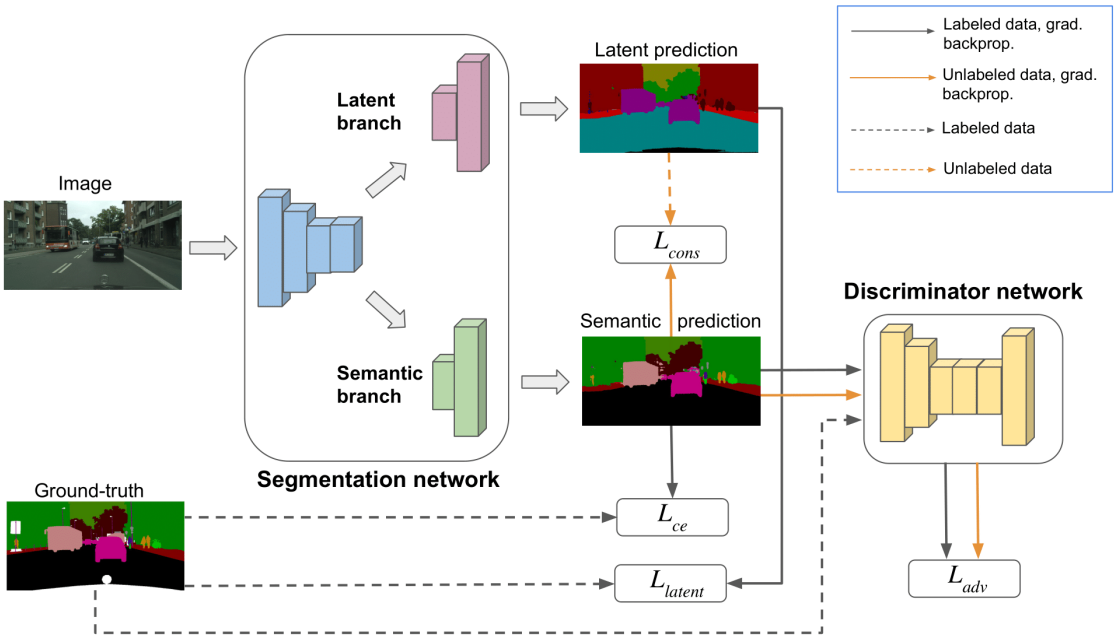


Figure 8.2: Overview of the proposed method. While the semantic branch infers pixel-wise class labels, the latent branch learns latent classes and infers the learned latent classes. The latent classes are learned only on the labeled images using the latent loss L_{latent} that ensures that the latent classes are as consistent as possible with the semantic classes. The semantic branch is trained on labeled images with the cross-entropy loss L_{ce} and on unlabeled images the predictions of the latent branch are used as supervision (L_{cons}). Additionally, the semantic branch receives adversarial feedback (L_{adv}) from a discriminator network distinguishing predicted and ground truth segmentations.

8.2.1 Semantic Branch

The task of the semantic branch S_c is to solve the final task of semantic segmentation, that is to predict the semantic classes for the input image. This branch is trained both on labeled and unlabeled data.

On labeled data, we optimize the semantic branch with respect to two loss terms. The first term is the cross-entropy loss:

$$L_{ce} = -\sum_{h,w,n}\sum_{c\in C} Y_n^{(h,w,c)} \log(S_c(X_n)^{(h,w,c)}) \tag{8.1}$$

where $X_n \in \mathbb{R}^{H\times W\times 3}$ is the image, $Y_n \in \mathbb{R}^{H\times W\times |C|}$ is the one-hot encoded ground truth for semantic classes, and S_c is the predicted probability of the semantic classes. To enforce realism in the semantic predictions, we additionally apply an adversarial loss:

$$L_{adv} = -\sum_{n,h,w} \log(D(S_c(X_n))^{(h,w)}) \tag{8.2}$$

Details of the discriminator network D are given in Section 8.2.4

On unlabeled data, the loss function for the semantic branch also consists of two terms. The first one is the adversarial term (8.2) and the second term is the consistency loss that is described in Section 8.2.3.

8.2.2 Latent Branch

In order to provide additional supervision for the semantic branch on the unlabeled data, we introduce a latent branch S_l that is trained only on the labeled data. The purpose of the latent branch is to learn latent classes that are easier to distinguish than the semantic classes and that can be better learned on a small set of labeled images. Figure 7.1 shows an example of latent classes where for instance semantic similar classes like vehicles are grouped together. One of the latent classes often corresponds to a stuff class that includes all difficult classes. This is desirable since having several latent classes that are easy to recognize and one latent class that contains the rest results in a simple segmentation task that can be learned from a small set of labeled images. However, we have to prevent a trivial solution where a single latent class contains all semantic classes. We therefore propose a loss that ensures that the latent classes $l \in \mathcal{L}$ have to provide as much information about semantic classes $c \in \mathcal{C}$ as possible.

To this end, we use the conditional entropy as loss:

$$L_{latent} = -\sum_{l \in \mathcal{L}} \sum_{c \in \mathcal{C}} P_b(c, l) \log(P_b(c|l)). \tag{8.3}$$

The loss is minimized if the variety of possible semantic classes for each latent class l is as low as possible. In the best case, there is a one-to-one mapping between the latent and semantic classes. The index b denotes that the probability is calculated batchwise. We first estimate the joint probability

$$P_b(c, l) = \frac{1}{NHW} \sum_{h,w,n} S_l(X_n)^{(h,w,l)} Y_n^{(h,w,c)} \tag{8.4}$$

where H and W are the image height and width, N is the number of images in the batch, S_l is the predicted probability of the latent classes, and $Y_n \in \mathbb{R}^{H \times W \times |\mathcal{C}|}$ is the one-hot encoded ground truth for the semantic classes. From this, we obtain

$$P_b(c|l) = \frac{P_b(c, l)}{\sum_c P_b(c, l)}. \tag{8.5}$$

Obtaining the conditional entropy from multiple batches is in principle desirable, but it requires the storage of feature maps from multiple batches. Therefore we compute it per batch.

8.2.3 Consistency Loss

While the latent branch is trained only on the labeled data, the purpose of the latent branch is to provide additional supervision for the semantic branch on the unlabeled data. Given that the latent branch solves a simpler task than the semantic branch, we can expect that the latent classes are more accurately predicted than the semantic classes. We therefore propose a loss that measures the consistency of the prediction of the semantic branch with the prediction of the latent branch. Since the

number of latent classes is less or equal than the number of semantic classes, we map the prediction of the semantic branch S_c to a probability distribution of latent classes $S_{\hat{l}_c}$:

$$S_{\hat{l}_c}(X_n)^{(h,w,l)} = \sum_{c \in \mathcal{C}} P(l|c) S_c(X_n)^{(h,w,c)}. \tag{8.6}$$

We estimate $P(l|c)$ from the predictions of the latent branch on the labeled data. We keep track of how often semantic and latent classes co-occur with an exponentially moving average:

$$M_{c,l}^{(i)} = (1-\alpha) M_{c,l}^{(i-1)} + \alpha \sum_{h,w,n} Y_n^{(h,w,c)} S_l(X_n)^{(h,w,l)} \tag{8.7}$$

where i denotes the number of the batch. The initialization is $M_{c,l}^0 = 0$. The parameter $0 < \alpha < 1$ controls how fast we update the average. We set α to the batch size divided by the number of images in the data set. Using the acquired co-occurence matrix M, $P(l|c)$ is estimated as:

$$P(l|c) = \frac{M_{c,l}}{\sum_{k \in \mathcal{L}} M_{c,k}}. \tag{8.8}$$

The consistency loss is then defined by the mean cross entropy between the latent variable maps predicted by the latent branch S_l and the ones constructed based on the prediction of the semantic branch $S_{\hat{l}_c}$:

$$L_{cons} = -\frac{1}{NHW} \sum_{n,h,w} \sum_{l \in \mathcal{L}} S_l(X_n)^{(h,w,l)} \log(S_{\hat{l}_c}(X_n)^{(h,w,l)}). \tag{8.9}$$

The minimization of this loss forces the semantic branch to predict classes which are assigned to highly probable latent classes.

8.2.4 Discriminator Network

Our discriminator network D is a fully-convolutional network *Shelhamer et al.* (2015) with 5 layers and Leaky-ReLu as nonlinearity. It takes label probability maps from the segmentation network or ground-truth maps as input and predicts spatial confidence maps. Each pixel represents the confidence of the discriminator about whether the corresponding pixel in a semantic label map was sampled from the ground-truth map or the segmentation prediction. We train the discriminator network with the help of the spatial cross-entropy loss using both labeled and unlabeled data:

$$L_D = -\sum_{h,w} (1-y_n) \log(1 - D(S_c(X_n))^{h,w}) + y_n \log(D(Y_n)^{h,w}) \tag{8.10}$$

where $y_n = 0$ if a sample is drawn from the segmentation network, and $y_n = 1$ if it is a ground-truth map. By minimizing such a loss, the discriminator learns to distinguish between the generated and ground-truth probability maps.

8.3 Experiments

8.3.1 Implementation Details

We conducted our experiments on three datasets for semantic segmentation: Pascal VOC 2012 *Everingham et al.* (2014), Cityscapes *Cordts et al.* (2016) and IIT Affordances *Nguyen et al.* (2017a). The Pascal VOC 2012 dataset contains images with objects from 20 foreground classes and one background class. There are 10528 training and 1449 validation images in total. The testing of the resulting model is carried out on the validation set. The Cityscapes dataset comprises images extracted from 50 driving videos. It contains 2975, 500 and 1525 images in the training, validation and test set, respectively, with annotated objects from 19 categories. We report the results of testing the resulting model on the validation set. The IIT Affordances dataset *Nguyen et al.* (2017a) contains images of 10 common human tools. It has 8835 images in total, where 50% are used for the training split, 20% for the validation split, and the rest 30% for the test split. Around 60% of the images in the dataset are from ImageNet, while the rest are taken from cluttered scenes, which implies a large variation of images within the dataset. As an evaluation metric, we use mean-intersection-over-union (mIoU)

For a fair comparison with Hung et al. *Hung et al.* (2018) and Mittal et al. *Mittal et al.* (2019), we choose the same backbone architecture and keep the same hyper-parameters where appropriate. For the segmentation network, we use a single scale ResNet-based DeepLab-v2 *Chen et al.* (2018b) architecture that is pre-trained on ImageNet *Russakovsky et al.* (2015) and MSCOCO *Lin et al.* (2014). We branch the proposed network at the last layer by applying Atrous Spatial Pyramid Pooling (ASPP) *Chen et al.* (2018b) two times for the semantic and latent branch. Finally, we use bilinear upsampling to make the predictions match the initial image size.

For the discriminator network, we use a fully convolutional network, which contains 5 convolutional layers with kernels of the sizes 4×4 and 64, 128, 256, 512 and 1 channels, applied with a stride equal to 2. Each convolutional layer, except for the last one, is followed by a Leaky-ReLU with the leakage coefficient equal to 0.2.

We train the segmentation network on labeled and unlabeled data jointly with $L = L_{labeled} + 0.1 \cdot L_{unlabeled}$ where the weight factor is the same as in *Hung et al.* (2018). The loss for the labeled and unlabeled data are given by

$$L_{labeled} = L_{ce} + L_{latent} + 0.01 \cdot L_{adv}, \tag{8.11}$$

$$L_{unlabeled} = L_{cons} + 0.01 \cdot L_{adv}. \tag{8.12}$$

The weight for the adversarial loss is also the same as in *Hung et al.* (2018). By default, we limit the number of latent classes to 20.

The optimization of the segmentation network is performed using SGD with a momentum equal to 0.9 and the learning rate decay of 10^{-4}. The learning rate, that is initially equal to $2.5 \cdot 10^{-4}$, is decreased with polynomial decay with the power of 0.9. For the discriminator, we employ the Adam optimizer *Kingma and Ba* (2015), where the initial learning rate is equal to 10^{-4} and that follows the same decay schedule as introduced for the segmentation network.

At each iteration, we alternately apply the described training scheme on the batch of the randomly sampled labeled and unlabeled data. To ensure the robustness of the evaluation procedure, we report results averaged over 5 random seeds that control the sampling procedure. We add the consistency

	Fraction of annotated images					
Method	1/50	1/20	1/8	1/4	1/2	Full
Hung et al. *Hung et al.* (2018)	55.6	64.6	69.5	72.1	73.8	74.9
Mittal et al. *Mittal et al.* (2019)	**63.3**	67.2	71.4	-	-	75.6
Proposed	59.6	68.2	71.3	**72.4**	**73.9**	75.0
Proposed + Classifier	61.8	**69.3**	**72.2**	-	-	75.3

Table 8.1: Comparison to the state-of-the-art on Pascal VOC 2012 using mIoU (%).

loss term only after 5000 iterations since the latent branch needs to learn some useful latent classes first.

On Pascal VOC 2012, during the training procedure, the images are cropped with crop size equal to 321×321 and undergo random scaling and horizontal mirroring. We train our model for 20k iterations with a batch size of 10 images. The testing of the resulting model is carried out on the validation set.

On the Cityscapes dataset, during training, we pre-process the images by performing cropping operations with crop size equal to 505×505 and additionally apply random scaling and horizontal mirroring. On the Cityscapes dataset, our model is trained for 40k iterations with batches of size 2. We report the results of testing the resulting model on the validation set.

On the IIT affordance dataset, the images are cropped with the crop size equal to 321×321 and undergo random scaling and horizontal mirroring. We train our model for 20k iterations with a batch size of 10 images on the training and validation images together. The testing of the resulting model is carried out on the test set.

8.3.2 Comparison with the State-of-the-Art

8.3.2.1 PASCAL VOC 2012.

On the PASCAL VOC 2012 dataset, we conducted our experiments on five fractions of annotated images, as shown in Table 8.1, where the rest of the images are used as unlabeled data. Since *Hung et al.* (2018) report the results only for the latest three fractions, we evaluate the performance of their method for the unreported fractions based on the publicly available code.

The improvement is especially pronounced, if we look at the sparsely labeled data fractions, such as $1/50$, $1/20$ and $1/8$. Our method performs on par with *Mittal et al.* (2019) and the leading method varies from data fraction to data fraction. However, our approach of learning latent variables is complementary to *Mittal et al.* (2019) and we can also add a classifier for refinement as in *Mittal et al.* (2019).

Figures 8.3 shows qualitative results on Pascal VOC 2012.

8.3.2.2 Cityscapes.

For the Cityscapes dataset, we follow the semi-supervised learning protocol that was proposed in *Hung et al.* (2018). This means that $1/8$, $1/4$ or $1/2$ of the training images are annotated and the

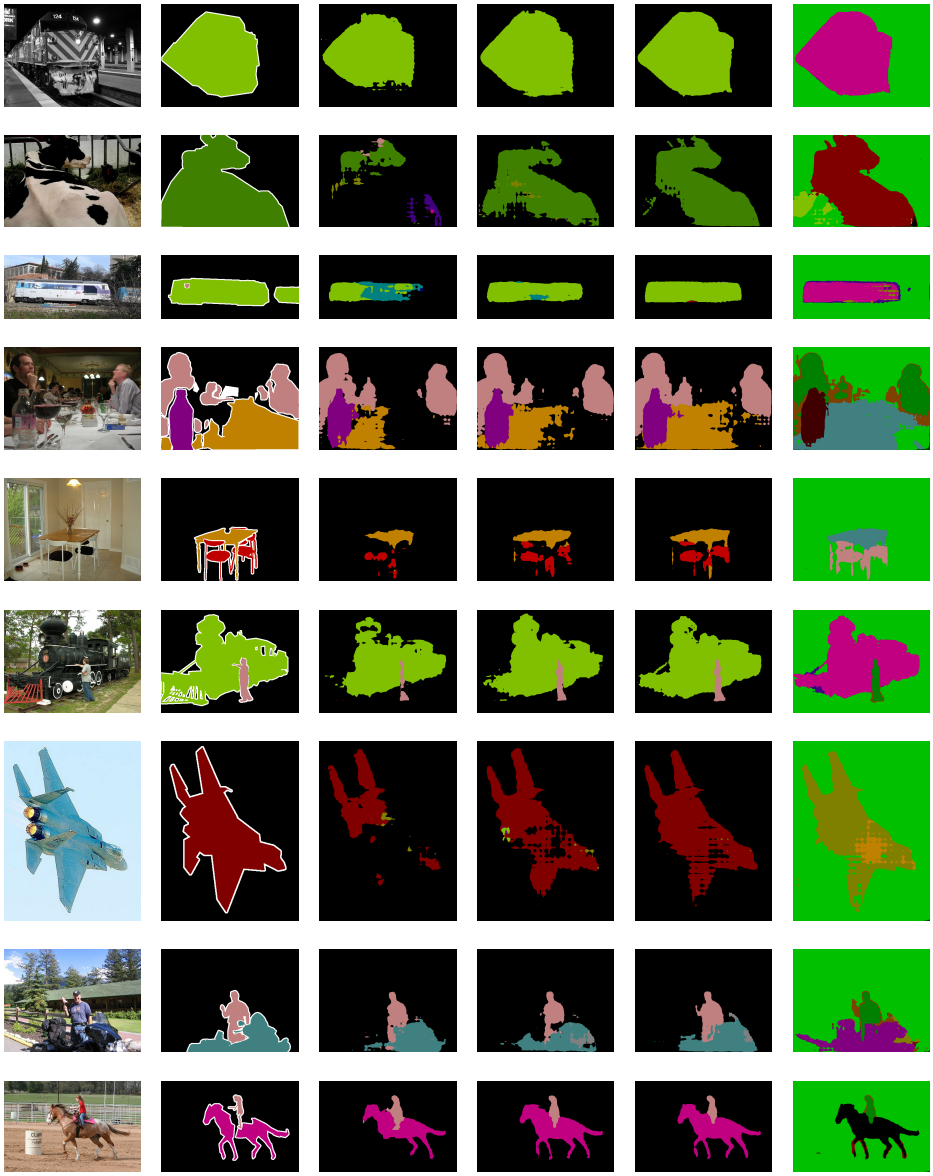


Figure 8.3: Qualitative examples from the Pascal VOC 2012 val set. From left to right: image, ground truth, L_{ce}, proposed without adversarial loss, proposed, latent classes.

Figure 8.4: Qualitative examples from the Cityscapes val set. From left to right: image, ground truth, proposed, latent classes.

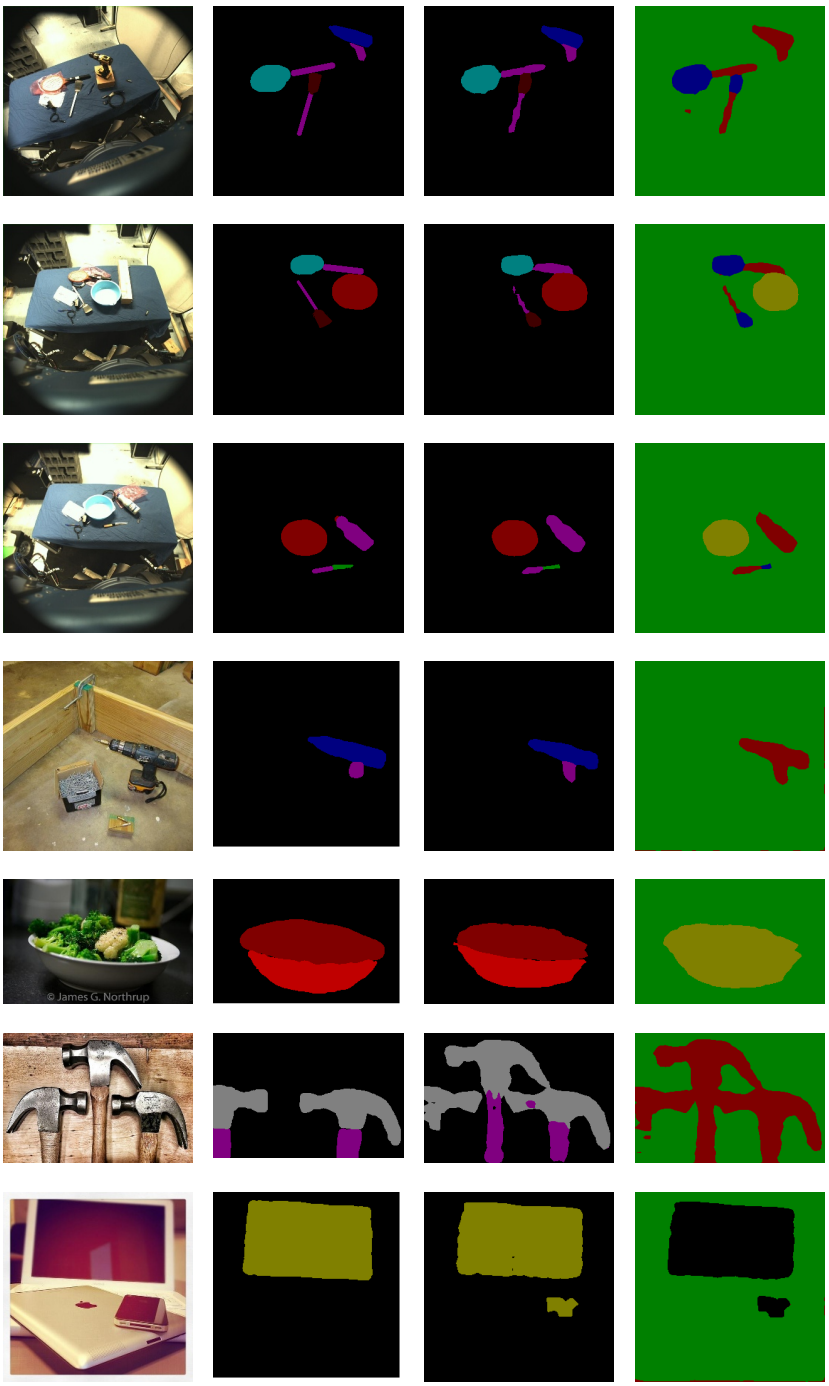


Figure 8.5: Qualitative examples from the IIT Affordances test set. From left to right: image, ground truth, proposed, latent classes.

		Fraction of annotated images			
Method	Pre-training	1/8	1/4	1/2	Full
Mittal et al. *Mittal et al.* (2019)		59.3	61.9	-	65.8
Proposed		61.0	63.1	-	64.9
Hung et al. *Hung et al.* (2018)	COCO	58.8	62.3	65.7	67.7
Proposed	COCO	**63.3**	**65.4**	**66.1**	66.3

Table 8.2: Comparison to the state-of-the-art on Cityscapes using mIoU (%).

IIT 2017 Affordances			
	Fraction of annotated images		
Method	1/50	1/20	1/8
	mIoU (%)		
Hung et al.*Hung et al.* (2018)	47.4	55.8	64.3
Proposed	**51.3**	**58.8**	**65.4**

Table 8.3: Comparison to Hung et.al on IIT Affordances. We used 7 latent classes for the proposed model

other images are used without any annotations. We report the results in Table 8.2. Since *Mittal et al.* (2019) does not pre-train the segmentation network on COCO, we evaluated our method also without COCO pre-training. We outperform both *Hung et al.* (2018) and *Mittal et al.* (2019) on all annotated data fractions.

Figure 8.4 shows qualitative results on Cityscapes.

8.3.3 IIT Affordances

We report the results in the Table 8.3. As for the other datasets, our approach outperforms *Hung et al.* (2018).

Figure 8.5 shows qualitative results on this dataset.

8.3.4 Ablation Experiments

In our ablation experiments, we evaluate the impact of each loss term. Then we examine the impact of the number of latent classes and show that they form meaningful supercategories of the semantic classes. Finally, we show that the learned latent classes outperform supercategories that are defined by humans.

Loss	mIoU (%)
L_{ce}	64.1
$L_{ce} + L_{latent}$	64.6
$L_{ce} + L_{latent} + L_{cons}$	67.3
$L_{ce} + L_{adv}^{labeled}$	68.7
$L_{ce} + L_{adv}$	69.4
$L_{ce} + L_{latent} + L_{cons} + L_{adv}$	71.3

Table 8.4: Impact of the loss terms. The evaluation is performed on Pascal VOC 2012 where 1/8 of the data is labeled. $L_{adv}^{labeled}$ denotes that the adversarial loss is only used for the labeled images.

8.3.4.1 Impact of the loss terms.

For analyzing the impact of the loss terms L_{ce} (8.1), L_{adv} (8.2), L_{latent} (8.3), and L_{cons} (8.9), we use the Pascal VOC 2012 dataset where 1/8 of the data is labeled. The results for different combinations of loss terms are reported in Table 8.4.

We start using only the entropy loss L_{ce} since this loss is always required. In this setting only the semantic branch is used and trained only on the labeled data. This setting achieves 64.1% mIoU. Adding the latent loss L_{latent} improves the performance by 0.5%. In this setting, the semantic and latent branch are used, but they are both only trained on the labeled data. Adding the consistency loss L_{cons} boosts the accuracy by 2.7%. This shows that the latent branch provides additional supervision for the semantic branch on the unlabeled data.

So far, we did not use the adversarial loss L_{adv}. When we add the adversarial loss only for the labeled data $L_{adv}^{labeled}$ to the entropy loss L_{ce}, the performance grows by 4.6%. In this setting, only the labeled data is used for training. If we use the adversarial loss also for the unlabeled data, the accuracy increases by 0.7%. This shows that the adversarial loss improves semi-supervised learning, but the gain is not as high compared to additionally using the latent branch to supervise the semantic branch on the unlabeled data. In this setting, all loss terms are used and the accuracy increases further by 1.9%. Compared to the entropy loss L_{ce}, the proposed loss terms increase the accuracy by 7.2%.

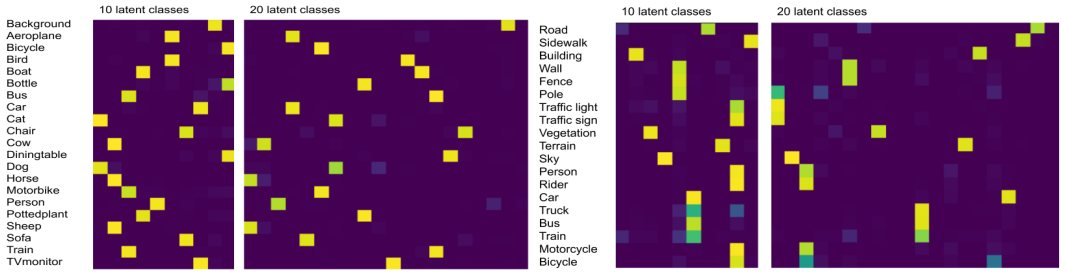
8.3.4.2 Impact of number of latent classes.

For our approach, we need to specify the maximum number of latent classes. While we used by default 20 in our previous experiments, we now evaluate it for 2, 4, 6, 10, and 20 latent classes on Pascal VOC 2012 with 1/8 of the data being labeled. The results are reported in Table 8.5. The performance grows monotonically with the number of latent classes reaching its peak for 20.

In the same table, we also report the number of effective latent classes. We consider a latent class l to be effectively used at threshold t, if $P(l|c) > t$ for at least one semantic class c. We report this number for $t = 0.1$ and $t = 0.9$. The number of effective latent classes differs only slightly for these two thresholds. This shows that a latent class typically either constitutes a supercategory of at least one semantic class or it is not used at all. We observe that until 10, all latent classes are used. If we allow up to 20 latent classes, only 14 latent classes are effectively used. In practice, we recommend to set the number of maximum latent classes to the number of semantic classes. The approach will then select as many latent classes as needed. Although we assume that the number of latent classes is

		Effective latent classes	
Max. latent classes	mIoU (%)	$t = 0.1$	$t = 0.9$
2	69.7	2	2
4	70.2	4	4
6	70.3	6	6
10	70.7	10	10
20	71.3	16	14
50	70.8	18	14

Table 8.5: Impact of the number of latent classes. The evaluation is performed on Pascal VOC 2012 where 1/8 of the data is labeled. A latent class l is considered effective, if there exists a semantic class c so that $P(l|c) > t$. The third column shows this number for $t = 0.1$ and the fourth for $t = 0.9$.



(a) $P(l|c)$ on Pascal VOC 2012 for 10 and 20 latent classes

(b) $P(l|c)$ on Cityscapes for 10 and 20 latent classes.

Figure 8.6: The distribution of latent classes for both datasets is pretty sparse, essentially the latent classes form supercategories of semantic classes that are similar in appearance. The grouping bicycle, bottle, and dining table for 10 latent classes seems to be unexpected, but due to the low number of latent classes the network is forced to group additional semantic classes. In this case, the network tends to group the most difficult classes of the dataset. In case of 20 latent classes, the merged classes are very intuitive, but not all latent classes are effectively used.

less or equal to the number of semantic classes, we also evaluated the approach for 50 latent classes. As expected, the accuracy drops but the approach remains stable. The number of effectively used latent classes also remains at 14. In practice, this setting should not be used since it violates the assumptions of the approach and can lead to unexpected behavior in some cases.

To see if a semantic class is typically mapped to a single latent class, we plot $P(l|c)$ for inference on Pascal VOC 2012 as well as on Cityscapes and show the results in Figure 8.6(a) and Figure 8.6(b), respectively. Indeed, the mapping from semantic classes to latent classes is very sparse. Typically, for each semantic class c, there is one dominant latent class l, i.e., $P(l|c) > 0.9$. If the number of latent classes increases to 20, some of the latent classes are not used. On Pascal VOC 2012, similar categories like cat and dog or cow, horse, and sheep are grouped. Some groupings are based on the common background like aeroplane and bird. The grouping bicycle, bottle, and dining table combines the most difficult classes of the dataset. However, we observed that there are small variations of the groupings for different runs when the number of latent classes is very small. On Cityscapes with

Mapping of semantic classes to supercategories	
Manually defined supercategory	VOC semantic classes
Background	Background
Aeroplane	Aeroplane
Bicycle	Bicycle
Bird	Bird
Boat	Boat
Person	Person
Ground vehicle with engine	Bus, car, motorbike, train
Mammal	Cat, cow, dog, horse, sheep
Furniture	Dinning table, sofa, chair
Miscellaneous	Bottle, tv monitor, potted plant

Table 8.6: Manual assignment of Pascal VOC 2012 classes to 10 supercategories that we use instead of learned latent classes in the ablation study.

20 latent classes, the semantic classes pole, traffic light, and traffic sign; person, rider, motorcycle, and bicycle; wall and fence; truck, bus, and train are grouped together. These groupings are very intuitive.

8.3.4.3 Comparison of learned latent classes with manually defined latent classes.

Since the latent classes typically learn supercategories of the semantic classes, the question arises if the same effect can be achieved with manually defined supercategories. In this experiment, the latent classes are replaced with 10 manually defined supercategories. Table 8.6 lists the manual assignment of the semantic classes to 10 supercategories.

In this setting, the latent branch is trained to predict these supercategories on the labeled data using the cross-entropy loss. For unlabeled data, everything remains the same as for the proposed method. We report the results in Table 8.7. The performance using the supercategories is only 69.0%, which is significantly below the proposed method for 10 latent variables.

Another approach would be to learn all semantic classes instead of the latent classes in the latent branch. In this case, both branches learn the same semantic classes. This gives 68.5%, which is also worse than the learned latent classes. If both branches predict the same semantic classes, we can also train them symmetrically. Being more specific, on labeled data they are both trained with the cross-entropy loss as well as the adversarial loss. On unlabeled data, we apply the adversarial loss to both of them and use the symmetric KL divergence as a consistency loss. This approach performs better, giving 69.1%, but it is still inferior to our proposed method. Overall, this shows the necessity to learn the latent classes in a data-driven way.

8.4 Conclusion

In this work, we addressed the task of semi-supervised semantic segmentation, where a small fraction of the dataset is labeled in a pixel-wise manner, while most images do not have any types of labeling. Our key contribution is a two-branch segmentation architecture, which uses latent classes

Method	Classes	mIoU (%)
Manual	10	69.0
Learned	10	70.7
Semantic classes	21	68.5
Semantic classes (KL)	21	69.1
Learned	20	71.3

Table 8.7: Comparison of learned latent classes with manually defined latent classes. The evaluation is performed on Pascal VOC 2012 where 1/8 of the data is labeled. In case of learned latent classes, the second column reports the maximum number of latent classes. In case of manually defined latent classes, the exact number of classes is reported.

learned in a data-driven way on labeled data to supervise the semantic segmentation branch on unlabeled data. We experimentally prove that the latent classes learned in this way have an interpretable meaning. Combined with an adversarial learning scheme, our method achieves new state-of-the-art results on Cityscapes, and performs on par with the state-of-the-art on the Pascal VOC 2012 dataset. All methods in previous Chapters implicitly assume certain properties of the data. The methods in Chapters 4, 5 and 6 rely on coarse localization cues from cheap annotations. These however are a poor approximation to faint structures like bicycles and will fail there. And image level supervision addressed in Chapter 7 does not help to locate omnipresent classes like *e.g.* roads in autonomous driving datasets. In the semi supervised setup, we do not make any assumptions about the data by design. Additionally, we can actively control the trade off between accuracy and cost by subsequently annotating a bigger part of the dataset. This is not possible in a weakly supervised setup, since the accuracy saturates below the fully supervised setup once you annotate all data with cheap supervision cues.

CHAPTER 9

Conclusion

Weakly supervised semantic segmentation was addressed in numerous works during the last 5 years. These works typically evaluated their approach on the Pascal VOC 2012 dataset (*Everingham et al.*, 2014). As supervision cues, the authors used keypoints, bounding boxes, scribbles or most popular, image tags. The current state-of-the-art works reach more than 60% mIoU using image tags only demonstrating enormous progress in this area.

However, Pascal VOC 2012 exhibits certain properties which make it particularly suitable for image tags as supervision cues.

- The classes of interest are objects often shown in an iconic setup.
- The classes are mutually exclusive.
- Typically, only a few classes appear in an image, there are no class pairs which almost always appear together or classes which are present in almost all images.
- It is sufficient to locate the object, but not the object parts inside the object.

This thesis examines several weakly and semi supervised semantic segmentation setups where these properties are not given. In our first three works we consider weakly supervised affordance segmentation. Affordances are functional attributes of object parts and the main difficulty is to locate the affordance within the object. Affordances need not be mutually exclusive since an object part can serve multiple needs. And finally, affordances appear in highly cluttered environments where the object of interest occupies a tiny part of the image.

Another problem with image tags is that they are usually assumed to be clean. This means they reduce the annotation cost but do not eliminate it completely. The question remains how well these methods would perform with noisy tags from the Internet. In our fourth work, we investigate image captions as a source of supervision. They contain noisy image tags as well as additional information like object attributes which provide additional supervision hints.

Weakly supervised semantic segmentation methods essentially compensate the lack of precise annotation with heuristics about data. Contrary, semi supervised semantic segmentation methods can in principle learn everything from data. In our last work we consider a general semi supervised setup and propose an algorithm which performs well on Cityscapes (*Cordts et al.*, 2016) a dataset for autonomous driving. It comprises stuff and object classes with many of them appearing essentially in every image.

9.1 Contributions

Our first contribution is the introduction of the CAD 120 affordance dataset. This dataset shows the relevant objects in a realistic setup and reflects two important properties of affordances. First, they

are in general not mutually exclusive, since an object part can be used for multiple purposes. This is reflected by a binary multi channel pixel-wise ground truth. And second, different object classes can exhibit the same affordance. In order to test if a method generalizes from the interior of a mug to the interior of a bowl, we propose two data splits. In the first split, the train and the test set contain the same object classes, while in the second split, none of the object classes from the test split appears in the train split.

As a second contribution, we are the first to propose two weakly supervised supervised segmentation methods for affordances using keypoints as supervision cues. The first method uses Grabcut (*Rother et al.*, 2004) to refine high confidence affordance regions. This method already achieves a significantly higher accuracy than out of the shelf weakly supervised semantic segmentation methods tailored to object classes. The second method is an improvement upon the first one in terms of performance, simplicity and training time. Generally, in a weakly supervised setup, the selection of hyper parameters by cross validation is impossible, since the ground truth is missing. One of the key features of our second method is hyper parameter selection by approximate cross validation.

Thirdly, we provide an algorithm to learn object part affordances from very few example tools. Assuming that bounding boxes for the tools are already given, our method works with a minimal annotation amount: It requires bounding boxes around object part affordances for very few instances (<10) per object class. As in the previous works, this approach significantly outperforms the contemporary state-of-the-art methods designed for weakly supervised semantic segmentation of object classes on the IIT-AFF affordance dataset (*Nguyen et al.*, 2017b). The reason for this comparison is the lack of weakly supervised affordance segmentation methods using the same supervision regime.

Our last contribution to weakly supervised semantic segmentation is the transition from image tags to image captions. Image captions are freely available on the Internet and provide additional information about the image content. The big difficulty with weakly supervised semantic segmentation is the lack of spatial hints. If one is able to localize attributes like color, objects having these attributes can be located by them. A challenge with utilizing captions is that they can contain arbitrary words. Therefore, we train a multi modal model which maps the image caption and the image itself to a common semantic embedding. The visual embeddings of regions referred by the text snippet have the highest correlation to the textual embedding: This allows us to use attributes of objects to locate them more precisely than from image tags only, which in turn leads to a new state-of-the-art results on the COCO dataset.

Our final contribution lies in the field of semi supervised semantic segmentation. We assume that a small fraction of training data has pixel-wise annotations and the remainder has no annotations of any type. To obtain a supervision signal on unlabeled data we learn supercategories on labeled data. These supercategories serve as a filter on unlabeled data suppressing semantic class predictions which do not belong to the supercategory. While supercategories can in principle be manually defined, for optimal performance, our algorithm discovers the latent supercategories itself. It chooses latent classes which reduce the uncertainty about the semantic class given the latent class. We prove that supervision from latent classes is complementary to other supervision signals like discriminator networks and significantly improves the segmentation accuracy. This performs on par with the current state-of-the-art method on the Pascal VOC 2012 dataset and outperforms it on the Cityscapes dataset.

9.2 Choosing the Right Method

This thesis examined 3 types of supervision levels: Image level supervision, sparse spacial clues inside the image and pixel wise annotation of a part of the data. The choice of the right method and consequently the annotation strategy depends on the data.

The cheapest cues are image captions as can be found on the Internet. The big advantage is that they scale well, *i.e.* do not require any annotation effort from the human. However, this type of supervision is not suitable for semantic classes which usually are not mentioned in image captions. Table 7.5 shows that while animals are typically mentioned by humans, kitchenware is of less interest. The former is mentioned in 92.3% of the cases while the latter in only 20.3%. In practice, one has to examine the data and see how often the relevant classes are mentioned in the captions. If the recall is low, this type of cues should not be used.

The more expensive alternative is to annotate the images with class tags or captions making sure that the semantic classes are mentioned if and only if they are present. Wrong correlations between image tags and image content are here a major problem. For example, the tag *blade of a knife* correlates with the grip of a knife in the image. The segmentation results shown in the second column of Figure 6.4 are consistent with the image tags, yet the spatial assignment of the object parts is wrong.

Given wrong correlations between image tags and image content, spatial clues are required to resolve them. In terms of cost, annotations of this type reside between image tags or captions and pixel wise annotations. It is generally impossible to estimate the reduction in annotation cost when moving from accurate polygons to key points or bounding boxes, since the number of vertices in a polygon depends on the object. For example, a door requires far less vertices than a bicycle. Generally the annotation cost will reduce roughly by one or two orders of magnitude. The method introduced in Chapter 6 is conceptually limited to object parts and works well for stiff objects. However, when the object is flexible, a far higher amount of example images is needed to propagate the part labels properly to the rest of the dataset. The methods from Chapter 4 and Chapter 5 are designed for non exclusive semantic labels and are conceptually not limited to affordances or object parts. The first method relies on two assumptions. The first is that color similarity is a strong clue for semantic similarity and the second is that the circle shape fits the shape of the object part well enough. The method in Chapter 5 also relies on the assumption that circles fit the shape of the semantic class well, but in a more subtle way. For classes with faint shape like a bicycle, approximate cross validation can in principle rule out too big circle sizes. However, a high number of circles is needed to cover a whole bicycle at least once. It is likely even higher than the number of polygon vertices needed for accurate pixel wise annotation. So this method will also probably fail for faint structures.

Finally the semi supervised approach from Chapter 8 requires the same amount of annotation effort per image as fully supervised ones. However, faint structures and wrong correlations do not constitute a problem for it.

In summary, image level supervision is suitable for data without wrong correlations. Sparse spatial cues work well if the geometry derived from the cues fits well the true shapes of the classes. If the data does not fit any of these scenarios, semi supervised learning is the only option.

In principle, supervision cues of different detail can be combined during the training of the neural network. If the semantic classes are mutually exclusive, one can extend the semi supervised approach from Chapter 8 to work with image labels. The most straight forward way would be to set the

activations of the semantic classes absent in the image to negative infinity before calculating S_c. For mutually non exclusive semantic classes, the method from Chapter 5 can be easily extended to work with fully annotated data as well. The only change would be to use the ground truth annotations on labeled data instead of the approximation obtained by binarization. Thus, in both cases, one can go with a one size fits all approach by investing half of the budget into pixel wise annotations and the other half into cheaper supervision cues.

9.3 Outlook

Semantic segmentation has many applications, *e.g.* in medical imaging, aerial surveillance or as a component in autonomous driving. Unfortunately, the annotation costs are extremely high when compared to *e.g.* visual question answering. Consequently, the community looked for approaches to learn semantic segmentation from less data. It achieved remarkable success on Pascal VOC 2012 (*Everingham et al.*, 2014) the dataset it mainly focuses on: The state-of-the-art methods learning from image tags only perform better than the fully supervised methods 4 years ago. However, the works in this thesis showed that the methods tailored to Pascal VOC 2012 do not necessary perform well on other datasets. Several research directions treated in this thesis need a further investigation:

Non-exclusive labels In Chapter 4 and Chapter 5, we deal with non-exclusive pixel-wise labels. In our case, the functional attributes of object parts are annotated and a particular pixel can have none, one or multiple attributes. Mutually non exclusive labels are not limited to affordances but are omnipresent. Objects exhibit different types of attributes like color, material, several physical attributes or affordances. Furthermore, as shown for affordances, a pixel can have multiple labels even within an attribute type. For this reason, weakly or semi supervised semantic segmentation of attributes will have to be designed for mutually non exclusive classes. Our works on the CAD 120 affordance dataset are a first step in this direction.

Limitations of learning from image labels. On Pascal VOC 2012, using the areas with high class activation as initial localisation cues turned out to be a very successful tactic. However, the fundamental weakness of this approach are wrong correlations between image tags and image content. For example, the class activation maps for object parts need not coincide with them but can highlight more salient object regions. Due to data biases, this problem is not constrained to object parts: Trains and railways, tennis balls and tennis fields, skis and skiers are such false friends occurring on COCO. It remains an open question if such complications can only be resolved by sparse spatial cues like keypoints or scribbles or if there exist more clever methods involving knowledge mining and logic. For example, in case of skis and skiers, one could suppress the activation for ski on the skier with an activation map for human. The stuff classes in autonomous driving datasets suffer especially heavily from this problem: Since the road and the sky are present in almost all images, image tags almost never provide a discriminatory signal. In any case, this is a fruitful research direction.

Webly supervision The ultimate achievement would be to entirely get rid of the human annotator. The Internet is a storage of human knowledge available for free. In Chapter 7 we simulated learning from image captions on the Internet by learning from image captions on COCO which are by design not explicitly related to the semantic classes of this dataset. Although we achieved promising results, reaching 28% mIoU is still far from satisfactory solving the task. A fundamental problem behind webly supervision is the biased noise in the annotations. While *e.g.* animals in safari images are almost always correctly mentioned, the recall for small objects like spoons in cluttered scenes can

be very low. This limits the applicability of image tag based approaches or their caption based generalizations. A solution might be additional image classifiers which estimate from the scene how likely image tags are missing and filtering out images with non reliable webly annotations.

Latent classes Discovering latent classes and subsequently using them for supervision on unlabeled data is not limited to semi supervised semantic segmentation of images. Any task where dense features have to be classified into dense labels are susceptible to this approach. While this is beyond the scope of this thesis, in principle, this method should be evaluated for tasks like semi supervised 3D segmentation or semantic completion as well as video segmentation or action anticipation in videos. Depending on the task, latent classes could be augmented with domain specific information like long term temporal models for action segmentation or anticipation.

Additionally, there exist setups not tackled at all in weakly and semi supervised semantic segmentation. Probably one of the biggest obstacles are missing datasets and benchmarks. These setups are:

Semantic hierarchy There is no semantic hierarchy on Pascal VOC 2012 or COCO. However, in the real world, humans can distinguish tens of thousands of semantic classes organized in hierarchies. To model this feature, datasets with semantic hierarchies as well as pixel wise semantic labels are required. From a technical point of view, the structure of supercategories and subclasses itself is a supervision signal, since it requires the consistency of pixel-wise semantic labels.

Semantic embedding The ultimate type of class labels would be semantic embeddings for each pixel as a ground truth. These embeddings should contain the noun, its supercategory and attributes. The big advantage of such an encoding is that it reflects semantic similarity. Current evaluation metrics penalize the confusion of a cat and a bicycle equally heavy to the confusion of cats and tigers, although the second one would be more understandable from a human point of view. If the labels were semantic embeddings instead of categorical labels, the second confusion would be less severe. The big question is of course how to assign semantic embeddings to the pixels. A first solution might be the concatenation of Word2Vec vectors of the class, its super classes and attributes.

Bibliography

Ahn, Jiwoon and Kwak, Suha. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4981–4990, 2018. (Cited on pages 24, 61, 65, 66 and 68.)

Ahn, Jiwoon; Cho, Sunghyun, and Kwak, Suha. Weakly supervised learning of instance segmentation with inter-pixel relations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2209–2218, 2019. (Cited on page 25.)

Akata, Zeynep; Reed, Scott; Walter, Daniel; Lee, Honglak, and Schiele, Bernt. Evaluation of output embeddings for fine-grained image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2927–2936, 2015. (Cited on page 19.)

Akbari, Hassan; Karaman, Svebor; Bhargava, Surabhi; Chen, Brian; Vondrick, Carl, and Chang, Shih-Fu. Multi-level multimodal common semantic space for image-phrase grounding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12476–12486, 2019. (Cited on page 25.)

Akizuki, Shuichi and Aoki, Yoshimitsu. Tactile Logging for Understanding Plausible Tool Use Based on Human Demonstration. In *British Machine Vision Conference*, 2018. (Cited on page 21.)

Badrinarayanan, Vijay; Kendall, Alex, and Cipolla, Roberto. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017. (Cited on page 22.)

Bearman, Amy; Russakovsky, Olga; Ferrari, Vittorio, and Fei-Fei, Li. What's the point: Semantic segmentation with point supervision. In *European Conference on Computer Vision*, pages 549–565, 2016. (Cited on pages 24, 28, 33, 34, 35, 36, 37, 44, 46 and 48.)

Bouritsas, Giorgos; Koutras, Petros; Zlatintsi, Athanasia, and Maragos, Petros. Multimodal visual concept learning with weakly supervised techniques. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4914–4923, 2018. (Cited on page 25.)

Briq, Rania; Moeller, Michael, and Gall, Juergen. Convolutional simplex projection network for weakly supervised semantic segmentation. In *British Machine Vision Conference*, 2018. (Cited on page 24.)

Caesar, Holger; Uijlings, Jasper, and Ferrari, Vittorio. COCO-Stuff: Thing and stuff classes in context. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1209–1218, 2018. (Cited on page 4.)

Castellini, Claudio; Tommasi, Tatiana; Noceti, Nicoletta; Odone, Francesca, and Caputo, Barbara. Using object affordances to improve object recognition. *Autonomous Mental Development*, 3(3): 207–215, 2011. (Cited on page 20.)

Chang, Wei-Lun; Wang, Hui-Po; Peng, Wen-Hsiao, and Chiu, Wei-Chen. All about structure: Adapting structural information across domains for boosting semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1900–1909, 2019. (Cited on page 23.)

Chao, Yu-Wei; Wang, Zhan; Mihalcea, Rada, and Deng, Jia. Mining semantic affordances of visual object categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4259–4267, 2015. (Cited on page 19.)

Chaudhry, Arslan; Dokania, Puneet K, and Torr, Philip H. S. Discovering Class-Specific Pixels for Weakly-Supervised Semantic Segmentation. In *British Machine Vision Conference*, 2017. (Cited on pages vi, ix, 24, 54, 56 and 66.)

Chen, Kan; Gao, Jiyang, and Nevatia, Ram. Knowledge aided consistency for weakly supervised phrase grounding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4042–4050, 2018a. (Cited on page 25.)

Chen, Liang-Chieh; Papandreou, George; Kokkinos, Iasonas; Murphy, Kevin, and Yuille, Alan L. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *International Conference on Learning Representations*, 2015. (Cited on pages 21, 30, 40, 43, 53 and 54.)

Chen, Liang-Chieh; Papandreou, George; Schroff, Florian, and Adam, Hartwig. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017. (Cited on page 21.)

Chen, Liang-Chieh; Papandreou, George; Kokkinos, Iasonas; Murphy, Kevin, and Yuille, Alan L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4): 834–848, 2018b. (Cited on pages 21, 30, 34, 40, 43, 58, 63, 68, 72 and 82.)

Chen, Liang-Chieh; Zhu, Yukun; Papandreou, George; Schroff, Florian, and Adam, Hartwig. Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv:1802.02611*, 2018c. (Cited on page 22.)

Chen, Minghao; Xue, Hongyang, and Cai, Deng. Domain adaptation for semantic segmentation with maximum squares loss. In *International Conference on Computer Vision*, pages 2090–2099, 2019a. (Cited on page 23.)

Chen, Wuyang; Jiang, Ziyu; Wang, Zhangyang; Cui, Kexin, and Qian, Xiaoning. Collaborative global-local networks for memory-efficient segmentation of ultra-high resolution images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8924–8933, 2019b. (Cited on page 23.)

Chen, Xianjie; Mottaghi, Roozbeh; Liu, Xiaobai; Fidler, Sanja; Urtasun, Raquel, and Yuille, Alan. Detect what you can: Detecting and representing objects using holistic models and body parts. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1979–1986, 2014. (Cited on pages x, 50 and 59.)

Cheng, Ming-Ming; Mitra, Niloy J.; Huang, Xiaolei; Torr, Philip H. S., and Hu, Shi-Min. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2015. (Cited on page 24.)

Choi, Jaehoon; Kim, Taekyung, and Kim, Changick. Self-ensembling with GAN-based data augmentation for domain adaptation in semantic segmentation. In *International Conference on Computer Vision*, pages 6829–6839, 2019. (Cited on page 23.)

Chu, Fu-Jen; Xu, Ruinian; Seguin, Landan, and Vela, Patricio A. Toward affordance detection and ranking on novel objects for real-world robotic manipulation. *IEEE Robotics and Automation Letters*, 4(4):4070–4077, 2019. (Cited on page 21.)

Cordts, Marius; Omran, Mohamed; Ramos, Sebastian; Rehfeld, Timo; Enzweiler, Markus; Benenson, Rodrigo; Franke, Uwe; Roth, Stefan, and Schiele, Bernt. The Cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. (Cited on pages 7, 78, 82 and 93.)

Dai, Dengxin; Sakaridis, Christos; Hecker, Simon, and Van Gool, Luc. Curriculum model adaptation with synthetic and real data for semantic foggy scene understanding. *International Journal on Computer Vision*, 128(5):1182–1204, 2019. (Cited on page 78.)

Datta, Samyak; Sikka, Karan; Roy, Anirban; Ahuja, Karuna; Parikh, Devi, and Divakaran, Ajay. Align2ground: Weakly supervised phrase grounding guided by image-caption alignment. In *International Conference on Computer Vision*, pages 2601–2610, 2019. (Cited on page 25.)

Dempster, Arthur P.; Laird, Nan M., and Rubin, Donald B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 1977. (Cited on page 15.)

Deng, Jia; Krause, Jonathan; Berg, Alexander C, and Fei-Fei, Li. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3450–3457, 2012. (Cited on page 19.)

Desai, Chaitanya and Ramanan, Deva. Predicting functional regions on objects. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 968–975, 2013. (Cited on page 20.)

Ding, Henghui; Jiang, Xudong; Liu, Ai Qun; Thalmann, Nadia Magnenat, and Wang, Gang. Boundary-aware feature propagation for scene segmentation. In *International Conference on Computer Vision*, pages 6818–6828, 2019a. (Cited on page 23.)

Ding, Henghui; Jiang, Xudong; Shuai, Bing; Liu, Ai Qun, and Wang, Gang. Semantic correlation promoted shape-variant context for segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8885–8894, 2019b. (Cited on pages 22 and 23.)

Do, Thanh-Toan; Nguyen, Anh; Reid, Ian; Caldwell, Darwin G, and Tsagarakis, Nikos G. AffordanceNet: An end-to-end deep learning approach for object affordance detection. In *IEEE International Conference on Robotics and Automation*, pages 1–5, 2018. (Cited on page 21.)

Du, Liang; Tan, Jingang; Yang, Hongye; Feng, Jianfeng; Xue, Xiangyang; Zheng, Qibao; Ye, Xiaoqing, and Zhang, Xiaolin. SSF-DAN: Separated semantic feature based domain adaptation network for semantic segmentation. In *International Conference on Computer Vision*, pages 982–991, 2019. (Cited on page 23.)

Durand, Thibaut; Mordan, Taylor; Thome, Nicolas, and Cord, Matthieu. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5957–5966, 2017. (Cited on page 25.)

Engilberge, Martin; Chevallier, Louis; Pérez, Patrick, and Cord, Matthieu. Finding beans in burgers: Deep semantic-visual embedding with localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3984–3993, 2018. (Cited on pages 25, 70 and 74.)

Everingham, Mark; Eslami, S. M.Ali; Van Gool, Luc.; Williams, Christopher K. I.; Winn, John, and Zisserman, Andrew. The Pascal Visual Object Classes Challenge: A retrospective. *International Journal on Computer Vision*, 111(1):98–136, 2014. (Cited on pages 4, 5, 7, 51, 78, 82, 93 and 96.)

Fan, Ruochen; Hou, Qibin; Cheng, Ming-Ming; Yu, Gang; Martin, Ralph R., and Hu, Shi-Min. Associating inter-image salient instances for weakly supervised semantic segmentation. In *European Conference on Computer Vision*, pages 371–388, 2018. (Cited on page 24.)

Fang, Kuan; Wu, Te-Lin; Yang, Daniel; Savarese, Silvio, and Lim, Joseph J. Demo2Vec: Reasoning object affordances from online videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2139–2147, 2018. (Cited on page 21.)

Farhadi, Alireza; Endres, Ian; Hoiem, Derek, and Forsyth, David. Describing objects by their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785, 2009. (Cited on page 19.)

Ferrari, Vittorio and Zisserman, Andrew. Learning visual attributes. In *Advances in Neural Information Processing Systems*, pages 433–440, 2007. (Cited on page 19.)

Fowler, Sam; Kim, Hansung, and Hilton, Adrian. Human-centric scene understanding from single view 360 video. In *International Conference on 3D Vision*, 2018. (Cited on page 20.)

Fu, Jun; Liu, Jing; Tian, Haijie; Li, Yong; Bao, Yongjun; Fang, Zhiwei, and Lu, Hanqing. Dual attention network for scene segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019. (Cited on page 23.)

Ge, Weifeng; Yang, Sibei, and Yu, Yizhou. Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1277–1286, 2018. (Cited on page 24.)

Grabner, Helmut; Gall, Juergen, and Van Gool, Luc. What makes a chair a chair? In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1529–1536, 2011. (Cited on page 20.)

Ham, Bumsub; Cho, Minsu; Schmid, Cordelia, and Ponce, Jean. Proposal flow: Semantic correspondences from object proposals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(7):1711–1725, 2017. (Cited on pages 51 and 54.)

He, Junjun; Deng, Zhongying, and Qiao, Yu. Dynamic multi-scale filters for semantic segmentation. In *International Conference on Computer Vision*, pages 3561–3571, 2019a. (Cited on page 23.)

He, Junjun; Deng, Zhongying; Zhou, Lei; Wang, Yali, and Qiao, Yu. Adaptive pyramid context network for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7519–7528, 2019b. (Cited on page 22.)

He, Kaiming; Gkioxari, Georgia; Dollár, Piotr, and Girshick, Ross. Mask R-CNN. In *International Conference on Computer Vision*, pages 2980–2988, 2017. (Cited on page 21.)

He, Tong; Shen, Chunhua; Tian, Zhi; Gong, Dong; Sun, Changming, and Yan, Youliang. Knowledge adaptation for efficient semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 578–587, 2019c. (Cited on page 23.)

Hermans, Tucker; Rehg, James M, and Bobick, Aaron. Affordance prediction via learned object attributes. In *IEEE International Conference on Robotics and Automation Workshops*, 2011. (Cited on pages 20 and 27.)

Hong, Seunghoon; Yeo, Donghun; Kwak, Suha; Lee, Honglak, and Han, Bohyung. Weakly supervised semantic segmentation using web-crawled videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2224–2232, 2017. (Cited on page 24.)

Hou, Qibin; Cheng, Ming-Ming; Hu, Xiao-Wei; Borji, Ali; Tu, Zhuowen, and Torr, Philip H. S. Deeply supervised salient object detection with short connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):815–828, 2018a. (Cited on page 24.)

Hou, Qibin; Massiceti, Daniela; Dokania, Puneet Kumar; Wei, Yunchao; Cheng, Ming-Ming, and Torr, Philip H. S. Bottom-up top-down cues for weakly-supervised semantic segmentation. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 263–277, 2018b. (Cited on page 24.)

Huang, Zilong; Wang, Xinggang; Wang, Jiasi; Liu, Wenyu, and Wang, Jingdong. Weakly-supervised semantic segmentation network with deep seeded region growing. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7014–7023, 2018. (Cited on pages x, 24, 66, 72 and 74.)

Huang, Zilong; Wang, Xinggang; Huang, Lichao; Huang, Chang; Wei, Yunchao, and Liu, Wenyu. CCNet: Criss-cross attention for semantic segmentation. In *International Conference on Computer Vision*, pages 603–612, 2019. (Cited on page 23.)

Hung, Wei-Chih; Tsai, Yi-Hsuan; Liou, Yan-Ting; Lin, Yen-Yu, and Yang, Ming-Hsuan. Adversarial learning for semi-supervised semantic segmentation. In *British Machine Vision Conference*, 2018. (Cited on pages 25, 77, 82, 83 and 87.)

Hwang, Jyh-Jing; Ke, Tsung-Wei; Shi, Jianbo, and Yu, Stella X. Adversarial structure matching loss for image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. (Cited on page 22.)

Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015. (Cited on page 12.)

Jiang, Huaizu; Wang, Jingdong; Yuan, Zejian; Wu, Yang; Zheng, Nanning, and Li, Shipeng. Salient object detection: A discriminative regional feature integration approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2083–2090, 2013a. (Cited on page 24.)

Jiang, Yun; Koppula, Hema, and Saxena, Ashutosh. Hallucinated humans as the hidden context for labeling 3d scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2993–3000, 2013b. (Cited on page 20.)

Jin, Bin; Segovia, Maria V. Ortiz, and Süsstrunk, Sabine. Webly supervised semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1705–1714, 2017. (Cited on page 24.)

Kalluri, Tarun; Varma, Girish; Chandraker, Manmohan, and Jawahar, C.V. Universal semi-supervised semantic segmentation. In *International Conference on Computer Vision*, pages 5258–5269, 2019. (Cited on page 25.)

Katz, Dov; Venkatraman, Arun; Kazemi, Moslem; Bagnell, J Andrew, and Stentz, Anthony. Perceiving, learning, and exploiting object affordances for autonomous pile manipulation. *Autonomous Robots*, 37(4):369–382, 2014. (Cited on pages 20 and 27.)

Ke, Tsung-Wei; Hwang, Jyh-Jing; Liu, Ziwei, and Yu, Stella X. Adaptive affinity fields for semantic segmentation. In *European Conference on Computer Vision*, pages 605–621, 2018. (Cited on page 22.)

Khan, Fahad Shahbaz; Anwer, Rao Muhammad; van de Weijer, Joost; Bagdanov, Andrew D; Vanrell, Maria, and Lopez, Antonio M. Color attributes for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3306–3313, 2012. (Cited on page 19.)

Khoreva, A.; Benenson, R.; Hosang, J.; Hein, M., and Schiele, B. Simple does it: Weakly supervised instance and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1665–1674, 2017. (Cited on pages 24 and 25.)

Kim, Dong In and Sukhatme, Gaurav. Semantic labeling of 3d point clouds with object affordance for robot manipulation. In *IEEE International Conference on Robotics and Automation*, pages 5578–5584, 2014. (Cited on pages 20 and 27.)

Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. (Cited on pages 12 and 82.)

Kjellström, Hedvig; Romero, Javier, and Kragić, Danica. Visual object-action recognition: Inferring object affordances from human demonstration. *Computer Vision and Image Understanding*, 115 (1):81–90, 2011. (Cited on pages 20 and 21.)

Kolesnikov, Alexander and Lampert, Christoph H. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European Conference on Computer Vision*, pages 695–711, 2016. (Cited on pages 24, 28, 33, 34, 35, 36, 37, 44, 46, 48, 61 and 74.)

Koppula, Hema S. and Saxena, Ashutosh. Physically grounded spatio-temporal object affordances. In *European Conference on Computer Vision*, pages 831–847. 2014. (Cited on pages 20 and 29.)

Koppula, Hema S. and Saxena, Ashutosh. Anticipating human activities using object affordances for reactive robotic response. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38 (1):14–29, 2016. (Cited on page 20.)

Krahenbuhl, Philipp and Koltun, Vladlen. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, pages 109–117, 2011. (Cited on pages 13, 22 and 66.)

Krause, Jonathan; Jin, Hailin; Yang, Jianchao, and Fei-Fei, Li. Fine-grained recognition without part annotations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5546–5555, 2015. (Cited on pages 25 and 59.)

Krizhevsky, Alex; Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. (Cited on page 13.)

Kumra, Sulabh and Kanan, Christopher. Robotic grasp detection using deep convolutional neural networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 769–776, 2017. (Cited on page 20.)

Kurmi, Vinod Kumar; Bajaj, Vipul; Venkatesh, K. S., and Namboodiri, Vinay P. Curriculum based dropout discriminator for domain adaptation. In *British Machine Vision Conference*, 2019. (Cited on page 78.)

Lampert, Christoph H; Nickisch, Hannes, and Harmeling, Stefan. Learning to detect unseen object classes by between-class attribute transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958, 2009. (Cited on page 19.)

Larsson, Mans; Stenborg, Erik; Hammarstrand, Lars; Pollefeys, Marc; Sattler, Torsten, and Kahl, Fredrik. A cross-season correspondence dataset for robust semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. (Cited on page 23.)

Lee, Jungbeom; Kim, Eunji; Lee, Sungmin; Lee, Jangho, and Yoon, Sungroh. FickleNet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5267–5276, 2019. (Cited on page 24.)

Lenz, Ian; Lee, Honglak, and Saxena, Ashutosh. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015. (Cited on page 20.)

Li, Hanchao; Xiong, Pengfei; Fan, Haoqiang, and Sun, Jian. DFANet: Deep feature aggregation for real-time semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9522–9531, 2019a. (Cited on page 23.)

Li, Hanxi; He, Xuming; Barnes, Nick, and Mingwen, Wang. Learning hough transform with latent structures for joint object detection and pose estimation. pages 116–129, 2016. ISBN 978-3-319-27673-1. doi: 10.1007/978-3-319-27674-8_11. (Cited on page 78.)

Li, Kunpeng; Wu, Ziyan; Peng, Kuan-Chuan; Ernst, Jan, and Fu, Yun. Tell me where to look: Guided attention inference network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9215–9223, 2018a. (Cited on page 24.)

Li, Qizhu; Arnab, Anurag, and Torr, Philip H. S. Weakly- and semi-supervised panoptic segmentation. In *European Conference on Computer Vision*, pages 106–124, 2018b. (Cited on page 24.)

Li, Xia; Zhong, Zhisheng; Wu, Jianlong; Yang, Yibo; Lin, Zhouchen, and Liu, Hong. Expectation-maximization attention networks for semantic segmentation. In *International Conference on Computer Vision*, pages 9166–9175, 2019b. (Cited on page 23.)

Li, Xueting; Liu, Sifei; Kim, Kihwan; Wang, Xiaolong; Yang, Ming-Hsuan, and Kautz, Jan. Putting humans in a scene: Learning affordance in 3d indoor environments. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12368–12376, 2019c. (Cited on page 21.)

Li, Yang; Liu, Yang; Liu, Guojun, and Guo, Maozu. Weakly supervised semantic segmentation by iterative superpixel-CRF refinement with initial clues guiding. *Neurocomputing*, 391:25–41, 2020. (Cited on page 74.)

Li, Yunsheng; Yuan, Lu, and Vasconcelos, Nuno. Bidirectional learning for domain adaptation of semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6936–6945, 2019d. (Cited on page 23.)

Lian, Qing; Lv, Fengmao; Duan, Lixin, and Gong, Boqing. Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In *International Conference on Computer Vision*, pages 6757–6766, 2019. (Cited on pages 23 and 78.)

Lin, Di; Dai, Jifeng; Jia, Jiaya; He, Kaiming, and Sun, Jian. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3159–3167, 2016a. (Cited on page 24.)

Lin, Guosheng; Shen, Chunhua; Van Den Hengel, Anton, and Reid, Ian. Efficient piecewise training of deep structured models for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3194–3203, 2016b. (Cited on page 22.)

Lin, Guosheng; Milan, Anton; Shen, Chunhua, and Reid, Ian. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5168–5177, 2017. (Cited on pages 21 and 22.)

Lin, Tsung-Yi; Maire, Michael; Belongie, Serge; Hays, James; Perona, Pietro; Ramanan, Deva; Dollár, Piotr, and Zitnick, C Lawrence. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. 2014. (Cited on pages 6, 59, 63, 68 and 82.)

Liu, Chenxi; Chen, Liang-Chieh; Schroff, Florian; Adam, Hartwig; Hua, Wei; Yuille, Alan L., and Fei-Fei, Li. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 82–92, 2019a. (Cited on page 23.)

Liu, Yifan; Chen, Ke; Liu, Chris; Qin, Zengchang; Luo, Zhenbo, and Wang, Jingdong. Structured knowledge distillation for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2604–2613, 2019b. (Cited on page 23.)

Liu, Ziwei; Li, Xiaoxiao; Luo, Ping; Loy, Chen Change, and Tang, Xiaoou. Semantic image segmentation via deep parsing network. In *International Conference on Computer Vision*, pages 1377–1385, 2015. (Cited on page 22.)

Lüddecke, Timo; Kulvicius, Tomas, and Wörgötter, Florentin. Context-based affordance segmentation from 2d images for robot actions. *Robotics and Autonomous Systems*, 119:92–107, 2019. (Cited on page 21.)

Luo, Yawei; Liu, Ping; Guan, Tao; Yu, Junqing, and Yang, Yi. Significance-aware information bottleneck for domain adaptive semantic segmentation. In *International Conference on Computer Vision*, pages 6777–6786, 2019. (Cited on page 23.)

Marin, Dmitrii; He, Zijian; Vajda, Peter; Chatterjee, Priyam; Tsai, Sam; Yang, Fei, and Boykov, Yuri. Efficient segmentation: Learning downsampling near semantic boundaries. In *International Conference on Computer Vision*, pages 2131–2141, 2019a. (Cited on page 23.)

Marin, Dmitrii; Tang, Meng; Ayed, Ismail Ben, and Boykov, Yuri. Beyond gradient descent for regularized segmentation losses. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019b. (Cited on page 23.)

Meng, Fanman; Li, Hongliang; Wu, Qingbo; Luo, Bing, and Ngan, King Ngi. Weakly supervised part proposal segmentation from multiple images. *IEEE Transactions on Image Processing*, 26: 4019–4031, 2017. (Cited on pages 25 and 59.)

Mikolov, Tomas; Sutskever, Ilya; Chen, Kai; Corrado, Greg S, and Dean, Jeff. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013. (Cited on pages 9, 17, 63 and 71.)

Mittal, Sudhanshu; Tatarchenko, Maxim, and Brox, Thomas. Semi-supervised semantic segmentation with high- and low-level consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1, 2019. (Cited on pages 25, 82, 83 and 87.)

Mou, Lichao; Hua, Yuansheng, and Zhu, Xiao Xiang. A relation-augmented fully convolutional network for semantic segmentation in aerial scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12416–12425, 2019. (Cited on page 23.)

Myers, Austin; Teo, Ching L; Fermüller, Cornelia, and Aloimonos, Yiannis. Affordance detection of tool parts from geometric features. In *IEEE International Conference on Robotics and Automation*, pages 1374–1381, 2015. (Cited on pages v, 20, 21, 27, 28, 31, 33, 34, 37, 39 and 43.)

Nekrasov, Vladimir; Chen, Hao; Shen, Chunhua, and Reid, Ian. Fast neural architecture search of compact semantic segmentation models via auxiliary cells. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9126–9135, 2019. (Cited on page 23.)

Nguyen, A.; Kanoulas, D.; Caldwell, D. G., and Tsagarakis, N. Object-based affordances detection with convolutional neural networks and dense conditional random fields. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017a. (Cited on page 82.)

Nguyen, Anh; Kanoulas, Dimitrios; Caldwell, Darwin G, and Tsagarakis, Nikos G. Detecting object affordances with convolutional neural networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2765–2770, 2016. (Cited on page 21.)

Nguyen, Anh; Kanoulas, Dimitrios; Caldwell, Darwin G, and Tsagarakis, Nikos G. Object-based affordances detection with convolutional neural networks and dense conditional random fields. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5908–5915, 2017b. (Cited on pages vi, ix, x, 6, 21, 50, 54, 55, 56, 57, 58, 59 and 94.)

Noh, Hyeonwoo; Hong, Seunghoon, and Han, Bohyung. Learning deconvolution network for semantic segmentation. In *International Conference on Computer Vision*, pages 1520–1528, 2015. (Cited on page 23.)

Oh, Seong Joon; Benenson, Rodrigo; Khoreva, Anna; Akata, Zeynep; Fritz, Mario, and Schiele, Bernt. Exploiting saliency for object segmentation from image level labels. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5038–5047, 2017. (Cited on page 24.)

Orsic, Marin; Kreso, Ivan; Bevandic, Petra, and Segvic, Sinisa. In defense of pre-trained ImageNet architectures for real-time semantic segmentation of road-driving images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12607–12616, 2019. (Cited on page 23.)

Pang, Yanwei; Li, Yazhao; Shen, Jianbing, and Shao, Ling. Towards bridging semantic gap to improve semantic segmentation. In *International Conference on Computer Vision*, pages 4229–4238, 2019. (Cited on page 23.)

Papandreou, George; Chen, Liang-Chieh; Murphy, Kevin P., and Yuille, Alan L. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *International Conference on Computer Vision*, pages 1742–1750, 2015. (Cited on pages 23, 28, 33, 34, 35, 36, 37, 44, 46 and 48.)

Parikh, Devi and Grauman, Kristen. Relative attributes. In *International Conference on Computer Vision*, pages 503–510, 2011. (Cited on page 19.)

Paszke, Adam; Chaurasia, Abhishek; Kim, Sangpil, and Culurciello, Eugenio. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv1606.02147*, 2016. (Cited on page 23.)

Pathak, Deepak; Krähenbühl, Philipp, and Darrell, Trevor. Constrained convolutional neural networks for weakly supervised segmentation. In *International Conference on Computer Vision*, pages 1796–1804, 2015. (Cited on page 24.)

Pham, Trung; Do, Thanh-Toan; Sünderhauf, Niko, and Reid, Ian. Scenecut: Joint geometric and object segmentation for indoor scenes. In *IEEE International Conference on Robotics and Automation*, pages 1–9, 2018. (Cited on page 21.)

Pinheiro, Pedro H. O. and Collobert, Ronan. From image-level to pixel-level labeling with convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1713–1721, 2015. (Cited on page 24.)

Pohlen, Tobias; Hermans, Alexander; Mathias, Markus, and Leibe, Bastian. Full-resolution residual networks for semantic segmentation in street scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4151–4160, 2017. (Cited on page 22.)

Pont-Tuset, Jordi; Arbeláez, Pablo; Barron, Jonathan T.; Marques, Ferran, and Malik, Jitendra. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(1):128–140, 2017. (Cited on page 24.)

Qi, Xiaojuan; Liu, Zhengzhe; Shi, Jianping; Zhao, Hengshuang, and Jia, Jiaya. Augmented feedback in semantic segmentation under image level supervision. In *European Conference on Computer Vision*, pages 90–105, 2016. (Cited on page 24.)

Razavi, Nima; Gall, Juergen; Kohli, Pushmeet, and Van Gool, Luc. Latent hough transform for object detection. In *European Conference on Computer Vision*, pages 312–325, 2012. (Cited on page 78.)

Richard, Alexander; Kuehne, Hildegard, and Gall, Jürgen. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1273–1282, 2017. (Cited on page 78.)

Rocco, Ignacio; Arandjelović, Relja, and Sivic, Josef. End-to-end weakly-supervised semantic alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6917–6925, 2018. (Cited on pages 51 and 54.)

Ronneberger, Olaf; Fischer, Philipp, and Brox, Thomas. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. (Cited on page 22.)

Rother, Carsten; Kolmogorov, Vladimir, and Blake, Andrew. GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Transaction on Graphics*, 23(3):309–314, 2004. (Cited on pages 9, 14, 24, 32 and 94.)

Roy, Anirban and Todorovic, Sinisa. A multi-scale CNN for affordance segmentation in RGB images. In *European Conference on Computer Vision*, pages 186–201, 2016. (Cited on pages 21, 27 and 28.)

Roy, Anirban and Todorovic, Sinisa. Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7282–7291, 2017. (Cited on page 24.)

Rumelhart, David E.; Hinton, Geoffrey E., and Williams, Ronald J. Learning representations by back-propagating errors. *Nature*, 323:533–, 1986. (Cited on page 12.)

Russakovsky, Olga; Deng, Jia; Su, Hao; Krause, Jonathan; Satheesh, Sanjeev; Ma, Sean; Huang, Zhiheng; Karpathy, Andrej; Khosla, Aditya; Bernstein, Michael; Berg, Alexander C., and Fei-Fei, Li. ImageNet large scale visual recognition challenge. *International Journal on Computer Vision*, 115(3):211–252, 2015. (Cited on pages 21, 54 and 82.)

Sakaridis, Christos; Dai, Dengxin, and Gool, Luc Van. Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In *International Conference on Computer Vision*, pages 7373–7382, 2019. (Cited on page 78.)

Saleh, Fatemehsadat; Akbarian, Mohammad Sadegh Ali; Salzmann, Mathieu; Petersson, Lars; Gould, Stephen, and Alvarez, Jose M. Built-in foreground/background prior for weakly-supervised semantic segmentation. In *European Conference on Computer Vision*, pages 413–432, 2016. (Cited on page 74.)

Sawatzky, Johann; Srikantha, Abhilash, and Gall, Juergen. Weakly supervised affordance detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2017. (Cited on page 5.)

Schoeler, Markus and Wörgötter, Florentin. Bootstrapping the semantics of tools: Affordance analysis of real world objects on a per-part basis. *IEEE Transactions on Cognitive and Developmental Systems*, pages 84–98, 2016. (Cited on page 21.)

Shannon, Claude Elwood. A mathematical theory of communication. In *The Bell Systems technical Journal*, pages 379–423, 1948. (Cited on page 16.)

Shelhamer, Evan; Long, Jonathan, and Darrell, Trevor. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. (Cited on pages 21 and 81.)

Shen, Yunhang; Ji, Rongrong; Wang, Yan; Wu, Yongjian, and Cao, Liujuan. Cyclic guidance for weakly supervised joint detection and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 697–707, 2019. (Cited on page 25.)

Shetty, Rakshith; Schiele, Bernt, and Fritz, Mario. Not using the car to see the sidewalk – quantifying and controlling the effects of context in classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8218–8226, 2019. (Cited on page 23.)

Shimoda, Wataru and Yanai, Keiji. Distinct class-specific saliency maps for weakly supervised semantic segmentation. In *European Conference on Computer Vision*, pages 218–234, 2016. (Cited on page 24.)

Shimoda, Wataru and Yanai, Keiji. Self-supervised difference detection for weakly-supervised semantic segmentation. In *International Conference on Computer Vision*, pages 5207–5216, 2019. (Cited on page 25.)

Siam, Mennatullah; Jiang, Chen; Lu, Steven; Petrich, Laura; Gamal, Mahmoud; Elhoseiny, Mohamed, and Jagersand, Martin. Video object segmentation using teacher-student adaptation in a human robot interaction (HRI) setting. In *IEEE International Conference on Robotics and Automation*, pages 50–56, 2019. (Cited on page 20.)

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. (Cited on page 33.)

Song, Chunfeng; Huang, Yan; Ouyang, Wanli, and Wang, Liang. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3136–3145, 2019. (Cited on page 24.)

Song, Hyun Oh; Fritz, Mario; Goehring, Daniel, and Darrell, Trevor. Learning to detect visual grasp affordance. *IEEE Transactions on Automation Science and Engineering*, 13(2):798–809, 2016. (Cited on page 20.)

Sun, Ruoqi; Zhu, Xinge; Wu, Chongruo; Huang, Chen; Shi, Jianping, and Ma, Lizhuang. Not all areas are equal: Transfer learning for semantic segmentation via hierarchical region selection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4360–4369, 2019. (Cited on page 23.)

Tang, Meng; Djelouah, Abdelaziz; Perazzi, Federico; Boykov, Yuri, and Schroers, Christopher. Normalized cut loss for weakly-supervised CNN segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1818–1827, 2018a. (Cited on page 24.)

Tang, Meng; Perazzi, Federico; Djelouah, Abdelaziz; Ben Ayed, Ismail; Schroers, Christopher, and Boykov, Yuri. On regularized losses for weakly-supervised CNN segmentation. In *European Conference on Computer Vision*, pages 524–540, 2018b. (Cited on page 24.)

Tian, Zhi; He, Tong; Shen, Chunhua, and Yan, Youliang. Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3126–3135, 2019. (Cited on page 23.)

Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N; Kaiser, Łukasz, and Polosukhin, Illia. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. (Cited on page 23.)

Vemulapalli, Raviteja; Tuzel, Oncel; Liu, Ming-Yu, and Chellapa, Rama. Gaussian conditional random field network for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3224–3233, 2016. (Cited on page 22.)

Vezhnevets, Alexander and Buhmann, Joachim M. Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3249–3256, 2010. (Cited on page 23.)

Vezhnevets, Alexander; Ferrari, Vittorio, and Buhmann, Joachim M. Weakly supervised semantic segmentation with a multi-image model. In *International Conference on Computer Vision*, pages 643–650, 2011. (Cited on page 23.)

Vezhnevets, Alexander; Ferrari, Vittorio, and Buhmann, Joachim M. Weakly supervised structured output learning for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 845–852, 2012. (Cited on page 23.)

Vu, Tuan-Hung; Jain, Himalaya; Bucher, Maxime; Cord, Matthieu, and Perez, Patrick. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019. (Cited on page 23.)

Wang, Panqu; Chen, Pengfei; Yuan, Ye; Liu, Ding; Huang, Zehua; Hou, Xiaodi, and Cottrell, Garrison. Understanding convolution for semantic segmentation. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1451–1460, 2018a. (Cited on pages 22 and 23.)

Wang, Wei; Yu, Kaicheng; Hugonot, Joachim; Fua, Pascal, and Salzmann, Mathieu. Recurrent U-Net for resource-constrained segmentation. In *International Conference on Computer Vision*, pages 2142–2151, 2019. (Cited on page 23.)

Wang, Xiang; You, Shaodi; Li, Xi, and Ma, Huimin. Weakly-supervised semantic segmentation by iteratively mining common object features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1354–1362, 2018b. (Cited on page 24.)

Wang, Xiang; Liu, Sifei; Ma, Huimin, and Yang, Ming-Hsuan. Weakly-supervised semantic segmentation by iterative affinity learning. *International Journal on Computer Vision*, 128:1736–1746, 2020. (Cited on page 74.)

Wei, Yunchao; Feng, Jiashi; Liang, Xiaodan; Cheng, Ming-Ming; Zhao, Yao, and Yan, Shuicheng. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6488–6496, 2017a. (Cited on page 24.)

Wei, Yunchao; Liang, Xiaodan; Chen, Yunpeng; Shen, Xiaohui; Cheng, Ming-Ming; Feng, Jiashi; Zhao, Yao, and Yan, Shuicheng. STC: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11): 2314–2320, 2017b. (Cited on page 24.)

Wei, Yunchao; Xiao, Huaxin; Shi, Honghui; Jie, Zequn; Feng, Jiashi, and Huang, Thomas S. Revisiting dilated convolution: A simple approach for weakly- and semi-supervised semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7268–7277, 2018. (Cited on page 24.)

Wei, Zhen; Zhang, Jingyi; Liu, Li; Zhu, Fan; Shen, Fumin; Zhou, Yi; Liu, Si; Sun, Yao, and Shao, Ling. Building detail-sensitive semantic segmentation networks with polynomial pooling. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7115–7123, 2019. (Cited on page 23.)

Wu, Zuxuan; Wang, Xin; Gonzalez, Joseph E.; Goldstein, Tom, and Davis, Larry S. ACE: Adapting to changing environments for semantic segmentation. In *International Conference on Computer Vision*, pages 2121–2130, 2019. (Cited on page 23.)

Xiao, Fanyi; Sigal, Leonid, and Lee, Yong Jae. Weakly-supervised visual grounding of phrases with linguistic structures. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5253–5262, 2017. (Cited on page 25.)

Xu, Jia; Schwing, Alexander, and Urtasun, Raquel. Tell me what you see and I will show you where it is. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3190–3197, 2014. (Cited on page 23.)

Yao, Qi and Gong, Xiaojin. Saliency guided self-attention network for weakly and semi-supervised semantic segmentation. *IEEE Access*, 2020. (Cited on page 74.)

Yu, Changqian; Wang, Jingbo; Peng, Chao; Gao, Changxin; Yu, Gang, and Sang, Nong. Learning a discriminative feature network for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1857–1866, 2018. (Cited on page 23.)

Yu, Fisher and Koltun, Vladlen. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations*, 2016. (Cited on page 21.)

Zeng, Yu; Zhuge, Yunzhi; Lu, Huchuan, and Zhang, Lihe. Joint learning of saliency detection and weakly supervised semantic segmentation. In *International Conference on Computer Vision*, pages 7222–7232, 2019. (Cited on page 24.)

Zhang, Fan; Chen, Yanqin; Li, Zhihang; Hong, Zhibin; Liu, Jingtuo; Ma, Feifei; Han, Junyu, and Ding, Errui. ACFNet: Attentional class feature network for semantic segmentation. In *International Conference on Computer Vision*, pages 6797–6806, 2019a. (Cited on page 23.)

Zhang, Hang; Dana, Kristin; Shi, Jianping; Zhang, Zhongyue; Wang, Xiaogang; Tyagi, Ambrish, and Agrawal, Amit. Context encoding for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018a. (Cited on page 23.)

Zhang, Jianming; Lin, Zhe; Brandt, Jonathan; Shen, Xiaohui, and Sclaroff, Stan. Top-down neural attention by excitation backprop. In *European Conference on Computer Vision*, pages 543–559, 2016. (Cited on page 24.)

Zhang, Wei; Zeng, Sheng; Wang, Dequan, and Xue, Xiangyang. Weakly supervised semantic segmentation for social images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2718–2726, 2015. (Cited on page 23.)

Zhang, Yang; David, Philip, and Gong, Boqing. Curriculum domain adaptation for semantic segmentation of urban scenes. In *International Conference on Computer Vision*, pages 2039–2049, 2017. (Cited on page 78.)

Zhang, Yiheng; Qiu, Zhaofan; Liu, Jingen; Yao, Ting; Liu, Dong, and Mei, Tao. Customizable architecture search for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11641–11650, 2019b. (Cited on page 23.)

Zhang, Yongqiang; Bai, Yancheng; Ding, Mingli; Li, Yongqiang, and Ghanem, Bernard. W2F: A weakly-supervised to fully-supervised framework for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 928–936, 2018b. (Cited on page 59.)

Zhao, Fang; Li, Jianshu; Zhao, Jian, and Feng, Jiashi. Weakly supervised phrase localization with multi-scale anchored transformer network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5696–5705, 2018a. (Cited on page 25.)

Zhao, Hengshuang; Shi, Jianping; Qi, Xiaojuan; Wang, Xiaogang, and Jia, Jiaya. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6230–6239, 2017. (Cited on page 22.)

Zhao, Hengshuang; Qi, Xiaojuan; Shen, Xiaoyong; Shi, Jianping, and Jia, Jiaya. Icnet for real-time semantic segmentation on high-resolution images. In *European Conference on Computer Vision*, pages 405–420, 2018b. (Cited on page 23.)

Zhao, Hengshuang; Zhang, Yi; Liu, Shu; Shi, Jianping; Change Loy, Chen; Lin, Dahua, and Jia, Jiaya. PSANet: Point-wise spatial attention network for scene parsing. In *European Conference on Computer Vision*, pages 267–283, 2018c. (Cited on page 23.)

Zheng, Shuai; Jayasumana, Sadeep; Romera-Paredes, Bernardino; Vineet, Vibhav; Su, Zhizhong; Du, Dalong; Huang, Chang, and Torr, Philip H. S. Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision*, pages 1529–1537, 2015. (Cited on page 22.)

Zhou, Bolei; Khosla, Aditya; Lapedriza, Agata; Oliva, Aude, and Torralba, Antonio. Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016. (Cited on pages 24, 61 and 63.)

Zhou, Yizhou; Sun, Xiaoyan; Zha, Zheng-Jun, and Zeng, Wenjun. Context-reinforced semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4046–4055, 2019. (Cited on page 22.)

Zhu, Xiangxin; Anguelov, Dragomir, and Ramanan, Deva. Capturing long-tail distributions of object subcategories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 915–922, 2014a. (Cited on page 78.)

Zhu, Yi; Zhou, Yanzhao; Xu, Huijuan; Ye, Qixiang; Doermann, David, and Jiao, Jianbin. Learning instance activation maps for weakly supervised instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3116–3125, 2019a. (Cited on page 25.)

Zhu, Yixin; Zhao, Yibiao, and Chun Zhu, Song. Understanding tools: Task-oriented object modeling, learning and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2855–2864, 2015. (Cited on page 20.)

Zhu, Yuke; Fathi, Alireza, and Fei-Fei, Li. Reasoning about object affordances in a knowledge base representation. In *European Conference on Computer Vision*, pages 408–424. 2014b. (Cited on page 19.)

Zhu, Zhen; Xu, Mengde; Bai, Song; Huang, Tengteng, and Bai, Xiang. Asymmetric non-local neural networks for semantic segmentation. In *International Conference on Computer Vision*, pages 593–602, 2019b. (Cited on page 23.)

Zhuang, Bohan; Shen, Chunhua; Tan, Mingkui; Liu, Lingqiao, and Reid, Ian. Structured binary neural networks for accurate image classification and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–422, 2019. (Cited on page 23.)