What's the Point: Semantic Segmentation with Point Supervision

Amy Bearman[1](✉), Olga Russakovsky[2], Vittorio Ferrari[3], and Li Fei-Fei[1]

[1] Stanford University, Stanford, USA
{abearman,feifeili}@cs.stanford.edu
[2] Carnegie Mellon University, Pittsburgh, USA
olgarus@cmu.edu
[3] University of Edinburgh, Edinburgh, Scotland, UK
vittorio.ferrari@ed.ac.uk

Abstract. The semantic image segmentation task presents a trade-off between test time accuracy and training time annotation cost. Detailed per-pixel annotations enable training accurate models but are very time-consuming to obtain; image-level class labels are an order of magnitude cheaper but result in less accurate models. We take a natural step from image-level annotation towards stronger supervision: we ask annotators to *point* to an object if one exists. We incorporate this point supervision along with a novel objectness potential in the training loss function of a CNN model. Experimental results on the PASCAL VOC 2012 benchmark reveal that the combined effect of point-level supervision and objectness potential yields an improvement of 12.9 % mIOU over image-level supervision. Further, we demonstrate that models trained with point-level supervision are more accurate than models trained with image-level, squiggle-level or full supervision given a fixed annotation budget.

Keywords: Semantic segmentation · Weak supervision · Data annotation

1 Introduction

At the forefront of visual recognition is the question of how to effectively teach computers new concepts. Algorithms trained from carefully annotated data enjoy better performance than their weakly supervised counterparts (e.g., [1] vs. [2,3] vs. [4,5] vs. [6]), yet obtaining such data is very time-consuming [5,7].

It is particularly difficult to collect training data for semantic segmentation, i.e., the task of assigning a class label to every pixel in the image. Strongly supervised methods require a training set of images with per-pixel annotations [3, 8–12] (Fig. 1). Providing an accurate outline of a single object takes between 54 s [13] and 79 s [5]. A typical indoor scene contains 23 objects [14], raising the annotation time to tens of minutes per image. Methods have been developed to reduce the annotation time through effective interfaces [5,15–19], e.g., through

B. Leibe et al. (Eds.): ECCV 2016, Part VII, LNCS 9911, pp. 549–565, 2016.
DOI: 10.1007/978-3-319-46478-7_34

requesting human feedback only as necessary [13]. Nevertheless, accurate per-pixel annotations remain costly and scarce.

To alleviate the need for large-scale detailed annotations, weakly supervised semantic segmentation techniques have been developed. The most common setting is where only image-level labels for the presence or absence of classes are provided during training [4,20–25], but other forms of weak supervision have been explored as well, such as bounding box annotations [4], eye tracks [26], free-form squiggles [17,18], or noisy web tags [27]. These methods require significantly less annotation effort during training, but are not able to segment new images nearly as accurately as fully supervised techniques.

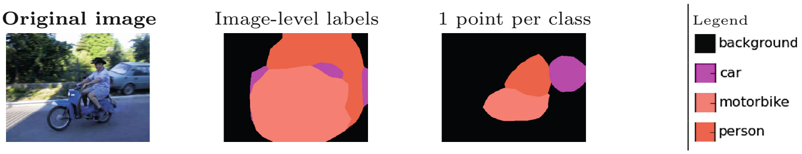

Fig. 1. Semantic segmentation models trained with our point-level supervision are much more accurate than models trained with image-level supervision (and even more accurate than models trained with full pixel-level supervision given the same annotation budget). The second two columns show test time results.

In this work, we take a natural step towards stronger supervision for semantic segmentation at negligible additional time, compared to image-level labels. The most natural way for humans to refer to an object is by pointing: "That cat over there" *(point)* or "What is that over there?" *(point)*. Psychology research has indicated that humans point to objects in a consistent and predictable way [3,28]. The fields of robotics [10,29] and human-computer interaction [9] have long used pointing as the effective means of communication. However, point annotation is largely unexplored in semantic segmentation.

Our **primary contribution** is a novel supervision regime for semantic segmentation based on humans pointing to objects. We extend a state-of-the-art convolutional neural network (CNN) framework for semantic segmentation [5,23] to incorporate point supervision in its training loss function. With just one annotated point per object class, we considerably improve semantic segmentation accuracy. We ran an extensive human study to collect these points on the PASCAL VOC 2012 dataset and evaluate the annotation times. We also make the user interface and the annotations available to the community.[1]

One lingering concern with supervision at the point level is that it is difficult to infer the full extent of the object. Our **secondary contribution** is incorporating an generic objectness prior [30] directly in the loss to guide the training of a CNN. This prior helps separate objects (e.g., car, sheep, bird) from background (e.g., grass, sky, water), by providing a probability that a pixel belongs to an

[1] Please refer to the project page: http://vision.stanford.edu/whats_the_point.

object. Such priors have been used in segmentation literature for proposing a set of candidate segments [31], selecting image regions to segment [32], as unary potentials in a conditional random field model [20], or during inference [25]. However, to the best of our knowledge, we are the first to employ this directly in the loss to guide the training of a CNN.

The combined effect of our contributions is a substantial increase of 12.9 % mean intersection over union (mIOU) on the PASCAL VOC 2012 dataset [33] compared to training with image-level labels (Fig. 1). Further, we demonstrate that models trained with point-level supervision outperform models trained with image-level, squiggle-level, and full supervision by 2.7–20.8 % mIOU given a fixed annotation budget.

2 Related Work

Types of Supervision for Semantic Segmentation. To reduce the up-front annotation time for semantic segmentation, recent works have focused on training models in a weakly- or semi-supervised setting. Many forms of supervision have been explored, such as eye tracks [26], free-form squiggles [17,18], noisy web tags [27], size constraints on objects [6] or heterogeneous annotations [34]. Common settings are image-level labels [4,23,25] and bounding boxes [4,35]. [14,36,37] use co-segmentation methods trained from image-level labels to automatically infer the segmentations. [6,23,25] train CNNs supervised only with image-level labels by extending the Multiple-Instance Learning (MIL) framework for semantic segmentation. [4,35] use an EM procedure, which alternates between estimating pixel labels from bounding box annotations and optimizing the parameters of a CNN.

There is a trade-off between annotation time and accuracy: models trained with higher levels of supervision are more accurate than weakly supervised models, but they require costly human-annotated datasets. We propose an intermediate form of supervision, using points, which adds negligible additional annotation time to image-level labels, yet achieves better accuracy. [19] also uses point supervision, but it trains a patch-level CNN classifier to serve as a unary potential in a CRF, whereas we use point supervision directly during CNN training.

CNNs for Segmentation. Recent successes in semantic segmentation have been driven by methods that train CNNs originally built for image classification to assign semantic labels to each pixel in an image [5,11,32,38]. One extension of the fully convolutional network (FCN) architecture developed by [5] is to train a multi-layer deconvolution network end-to-end [39]. More inventive forms of post-processing have also been developed, such as combining the responses at the final layer of the network with a fully-connected CRF [38]. We develop our approach on top of the basic framework common to many of these methods.

Interactive Segmentation. Some semantic segmentation methods are interactive, in that they collect additional annotations at test time to refine the segmentation. These annotations can be collected as points [2] or free-form squiggles [15]. These methods require additional user input at test time; in contrast, we only collect user points once and only use them at training time.

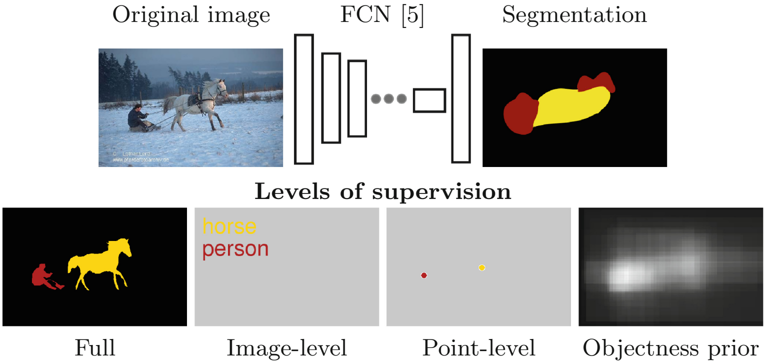

Fig. 2. (*Top*): Overview of our semantic segmentation training framework. (*Bottom*): Different levels of training supervision. For full supervision, the class of every pixel is provided. For image-level supervision, the class labels are known but their locations are not. We introduce point-level supervision, where each class is only associated with one or a few pixels, corresponding to humans pointing to objects of that class. We include an objectness prior in our training loss function to accurately infer the object extent.

3 Semantic Segmentation Method

We describe here our approach to using point-level supervision (Fig. 2) for training semantic segmentation models. In Sect. 4, we will demonstrate that this level of supervision is cheap and efficient to obtain. In our setting (in contrast to [2]), supervised points are only provided on training images. The learned model is then used to segment test images with no additional human input.

Current state-of-the-art semantic segmentation methods [4,5,23,25,38], both supervised and unsupervised, employ a unified CNN framework. These networks take as input an image of size $W \times H$ and output a $W \times H \times N$ score map where N is the set of classes the CNN was trained to recognize (Fig. 2). At test time, the score map is converted to per-pixel predictions of size $W \times H$ by either simply taking the maximally scoring class at each pixel [5,23] or employing more complicated post-processing [4,25,38].

Training models with different levels of supervision requires defining appropriate loss functions in each scenario. We begin by presenting two of the most commonly used in the literature. We then extend them to incorporate (1) our proposed point supervision and (2) a novel objectness prior.

Full Supervision. When the class label is available for every pixel during training, the CNN is commonly trained by optimizing the sum of per-pixel cross-entropy terms [5,38]. Let $\mathcal{I}$ be the set of pixels in the image. Let s_{ic} be the CNN score for pixel i and class c. Let $S_{ic} = \exp(s_{ic}) / \sum_{k=1}^{N} \exp(s_{ik})$ be the softmax probability of class c at pixel i. Given a ground truth map G indicating that pixel i belongs to class G_i, the loss on a single training image is:

$$\mathcal{L}_{pix}(S, G) = -\sum_{i \in \mathcal{I}} \log(S_{iG_i}) \tag{1}$$

The loss is simply zero for pixels where the ground truth label is not defined (e.g., in the case of pixels defined as "difficult" on the boundary of objects in PASCAL VOC [33]).

Image-Level Supervision. In this case, the only information available during training are the sets $L \subseteq \{1, \ldots, N\}$ of classes present in the image and $L' \subseteq \{1, \ldots, N\}$ of classes not present in the image. The CNN model can be trained with a different cross-entropy loss:

$$\mathcal{L}_{img}(S, L, L') = -\frac{1}{|L|} \sum_{c \in L} \log(S_{t_c c}) - \frac{1}{|L'|} \sum_{c \in L'} \log(1 - S_{t_c c}) \tag{2}$$
$$\text{with } t_c = \underset{i \in \mathcal{I}}{\text{argmax}}\ S_{ic}$$

The first part of Eq. (2), corresponding to $c \in L$, is used in [23]. It encourages each class in L to have a high probability on at least one pixel in the image. The second part has been added in [6], corresponding to the fact that no pixels should have high probability for classes that are not present in the image.

Point-Level Supervision. We study the intermediate case where the object classes are known for a small set of supervised pixels $\mathcal{I}_s$, whereas other pixels are just known to belong to some class in L. We generalize Eqs. (1) and (2) to:

$$\mathcal{L}_{point}(S, G, L, L') = \mathcal{L}_{img}(S, L, L') - \sum_{i \in \mathcal{I}_s} \alpha_i \log(S_{iG_i}) \tag{3}$$

Here, α_i determines the relative importance of each supervised pixel. We experiment with several formulations for α_i. (1), for each class we ask the user to either determine that the class is not present in the image or to point to one object instance. In this case, $|\mathcal{I}_s| = |L|$ and α_i is uniform for every point; (2), we ask multiple annotators to do the same task as (1), and we set α_i to be the confidence of the accuracy of the annotator that provided the point; (3), we ask the annotator(s) to point to every *instance* of the classes in the image, and α_i corresponds to the *order* of the points: the first point is more likely to correspond to the largest object instance and thus deserves a higher weight α_i.

Objectness Prior. One issue with training models with very few or no supervised pixels is correctly inferring the spatial extent of the objects. In general, weakly supervised methods are prone to local minima: focusing on only a small part of the target object, or predicting all pixels as belonging to the background class [23]. To alleviate this problem, we introduce an additional term in our training objective based on an objectness prior (Fig. 2). Objectness provides a probability for whether each pixel belongs to *any* object class [30] (e.g., bird, car, sheep), as opposed to background (e.g., sky, water, grass). These probabilities have been used in the weakly supervised semantic segmentation literature before as unary potentials in graphical models [20] or during inference following a CNN segmentation [25]. To the best of our knowledge, we are the first to incorporate them directly into CNN training.

Let P_i be the probability that pixel i belongs to an object. Let $\mathcal{O}$ be the classes corresponding to objects, with the other classes corresponding to backgrounds. In PASCAL VOC, $\mathcal{O}$ is the 20 object classes, and there is a single generic background class. We define a new loss:

$$\mathcal{L}_{obj}(S,P) = -\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} P_i \log\left(\sum_{c\in\mathcal{O}} S_{ic}\right) + (1-P_i)\log\left(1-\sum_{c\in\mathcal{O}} S_{ic}\right) \tag{4}$$

At pixels with high P_i values, this objective encourages placing probability mass on object classes. Alternatively, when P_i is low, it prefers mass on the background class. Note that $\mathcal{L}_{obj}$ requires no human supervision (beyond pre-training the generic objectness detector), and thus can be combined with any loss above.

4 Crowdsourcing Annotation Data

In this section, we describe our method for collecting annotations for the different levels of supervision. The annotation time required for point-level and squiggle-level supervision was measured directly during data collection. For other types of supervision, we rely on the annotation times reported in the literature.

Image-Level Supervision (20.0 sec/img). Collecting image-level labels takes 1 seconds per class [26]. Thus, annotating an image with 20 object classes in PASCAL VOC is expected to take 20 seconds per image.

Full Supervision (239.7 sec/img). There are 1.5 object classes per image on average in PASCAL VOC 2012 [33]. It takes 1 s to annotate every object that is not present (to obtain an image-level "no" label), for 18.5 s of labeling time. Additionally, there are 2.8 object instances on average per image that need to be segmented [33]. The authors of the COCO dataset report 22 worker hours for 1,000 segmentations [16]. This implies a mean labeling time of 79 seconds per object segmentation, adding 2.8 × 79 s of labeling in our case. Thus, the total expected annotation time is 239.7 seconds per image.

4.1 Point-Level Supervision (22.1 sec/img)

We used Amazon Mechanical Turk (AMT) to annotate point-level supervision on 20 PASCAL VOC object classes over 12,031 images: all training and validation images of the PASCAL VOC 2012 segmentation task [33] plus the additional images of [40]. Figure 3 (left) shows the annotation inferface and Fig. 3 (center) shows some collected data. We use two different point-level supervision tasks.

For each image, we obtain either (1) one annotated point per object class, on the first instance of the class the annotator sees (1*Point*), and (2) one annotated point per object instance (*AllPoints*). We make these collected annotations and the annotation system publicly available.

Annotation Time. There are 1.5 classes on average per image in PASCAL VOC 2012. It takes workers a median of 2.4 s to click on the first instance of an object. Thus, the labeling 1*Point* takes $18.5 \times 1 + 1.5 \times 2.4 = \mathbf{22.1}$ seconds per image. It takes workers a median of 0.9 s to click on every additional instance of an object class. There are 2.8 instances on average per image, so labeling *AllPoints* takes $18.5 \times 1 + 1.5 \times 2.4 + (2.8 - 1.5) \times 0.9 = \mathbf{23.3}$ seconds per image. Point supervision is only 1.1–1.2× more time-consuming than obtaining image-level labels, and more than 10× cheaper than full supervision.

Quality Control. Quality control for point annotation was done by planting 10 evaluation images in a 50-image task and ensuring that at least 8 are labeled correctly. We consider a point correct if it falls inside a tight bounding box around the object. For the *AllPoints* task, the number of annotated clicks must be at least the number of known object instances.

Error Rates. Simply determining the presence or absence of an object class in an image was fairly easy, and workers incorrectly labeled an object class as absent only 1.0 % of the time. On the 1*Point* task, 7.2 % of points were on a pixel with a different class label (according to the PASCAL ground truth), and an additional 0.8 % were on an unclassified "difficult" pixel. For comparison, [41] reports much higher 25 % average error rates when drawing bounding boxes. Our collected data is high-quality, confirming that pointing to objects comes naturally to humans [3,9].

Annotators had more difficulty with the *AllPoints* class: 7.9 % of ground truth instances were left unannotated, 14.8 % of the clicks were on the wrong object class, and 1.6 % on "difficult" pixels. This task caused some confusion among workers due to blurry or very small instances; for example, many of these instances are not annotated in the ground truth but were clicked by workers, accounting for the high false positive rate.

4.2 Squiggle-Level Supervision (34.9 sec/img)

[17,18] have experimented with training with free-form squiggles, where a subset of pixels are labeled. While [17] simulates squiggles by randomly labeling superpixels from the ground truth, we follow [18] in collecting squiggle annotations (and annotation times) from humans for 20 object classes on all PASCAL VOC 2012 trainval images. This allows us to properly compare this supervision setting to human points. We extend the user interface shown in Fig. 3 (left) by asking annotators to draw one squiggle on one instance of the target class. Figure 3 (right) shows some collected data.

Annotation Time. As before, it takes 18.5 s to annotate the classes not present in the image. For every class that is present, it takes 10.9 s to draw a free-form squiggle on the target class. Therefore, the labeling time of 1*Squiggle* is

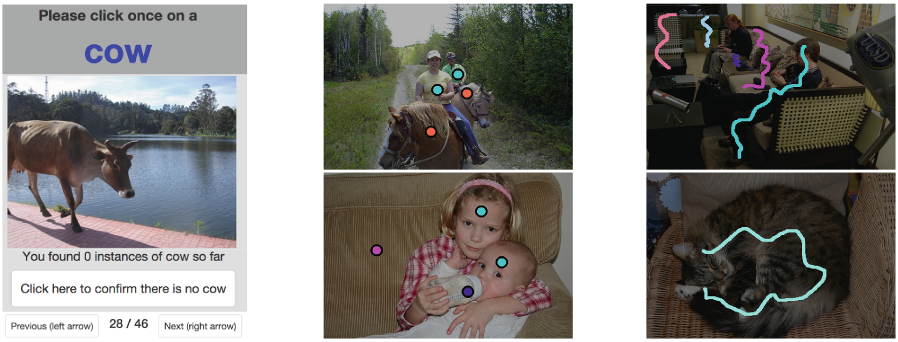
Please click once on a
cow
You found 0 instances of cow so far
Click here to confirm there is no cow
Previous (left arrow)
28 / 46
Next (right arrow)

Fig. 3. *Left.* AMT annotation UI for point-level supervision. *Center.* Example points collected. *Right.* Example squiggles collected. Colors correspond to different classes. (Color figure online)

$18.5 + 1.5 \times 10.9 = \mathbf{34.9}$ seconds per image. This is 1.6× more time-consuming than obtaining 1*Point* point-level supervision and 1.7× more than image-level labels.

Error Rates. We used similar quality control to point-level superivision. Only 6.3 % of the annotated pixels were on the wrong object class, and an additional 1.4 % were on pixels marked as "difficult" in PASCAL VOC [33].

In Sect. 5 we compare the accuracy of the models trained with different levels of supervision.

5 Experiments

We empirically demonstrate the efficiency of our point-level and objectness prior. We compare these forms of supervision against image-level labels, squiggle-level, and fully supervised data. We conclude that point-level supervision makes a much more efficient use of annotator time, and produces much more effective models under a fixed time budget.

5.1 Setup

Dataset. We train and evaluate on the PASCAL VOC 2012 segmentation dataset [33] augmented with extra annotations from [40]. There are 10,582 training images, 1,449 validation images and 1,456 test images. We report the mean intersection over union (mIOU), averaged over 21 classes.

CNN Architecture. We use the state-of-the-art fully convolutional network model [5]. Briefly, the architecture is based on the VGG 16-layer net [8], with all fully connected layers converted to convolutional layers. The last classifier layer is discarded and replaced with a 1×1 convolution layer with channel dimension $N = 21$ equal to the number of object classes. The final modification is the addition of a deconvolution layer to bilinearly upsample the output to pixel-level dense predictions.

CNN Training. We train following a procedure similar to [5]. We use stochastic gradient descent with a fixed learning rate of 10^{-5}, doubling the learning rate for biases, and with a minibatch of 20 images, momentum of 0.9 and weight decay 0.0005. The network is initialized with weights pre-trained for a 1000-way classification task of the ILSVRC 2012 dataset [5,7,8].[2] In the fully supervised case we zero-initialize the classifier weights [5], and for all the weakly supervised cases we follow [23] to initialize them with weights learned by the original VGG network for classes common to both PASCAL and ILSVRC. We backpropagate through all layers to fine-tune the network, and train for 50,000 iterations. We build directly on the publicly available implementation of [5,42].[3]

Objectness prior. We calculate the per-pixel objectness prior by assigning each pixel the average objectness score of all windows containing it. These scores are obtained by using the pre-trained model from the released code of [30]. The model is trained on 50 images with 291 object instances randomly sampled from a variety of different datasets (e.g., INRIA Person, Caltech 101) that do not overlap with PASCAL VOC 2007–2012 [30]. For fairness of comparison, we include the annotation cost of training the objectness prior. We estimate the 291 bounding boxes took 10.2 s each on average to obtain [41], adding up to a total of 49.5 min of annotation. Amortized across the 10,582 PASCAL training images, using the objectness prior thus costs **0.28 s** of extra annotation per image.

5.2 Synergy Between Point-Level Supervision and Objectness Prior

We first establish the baselines of our model and show the benefits of both point-level supervision and objectness prior. Table 1 (top) summarizes our findings and Table 2 (top) shows the per-class accuracy breakdown.

Baseline. We train a baseline segmentation model from image-level labels with no additional information. We base our model on [23], which trains a similar fully convolutional network and obtains 25.1 % mIOU on the PASCAL VOC 2011 validation set. We notice that the *absence* of a class label in an image is also an important supervisor signal, along with the presence of a class label, as in [6]. We incorporate this insight into our loss function $\mathcal{L}_{img}$ in Eq. 2, and see a substantial 5.4 % improvement in mIOU from the baseline, when evaluated on the PASCAL VOC 2011 validation set.

Effect of Point-Level Supervision. We now run a key experiment to investigate how having just one annotated point per class per image improves semantic

[2] Standard in the literature [1,4,5,23,25,38]. We do not consider the cost of collecting those annotations; including them would not change our overall conclusions.

[3] [5] introduces additional refinement by decreasing the stride of the output layers from 32 pixels to 8 pixels, which improves their results from 59.7 % to 62.7 % mIOU on the PASCAL VOC 2011 validation set. We use the original model with stride of 32 for simplicity.

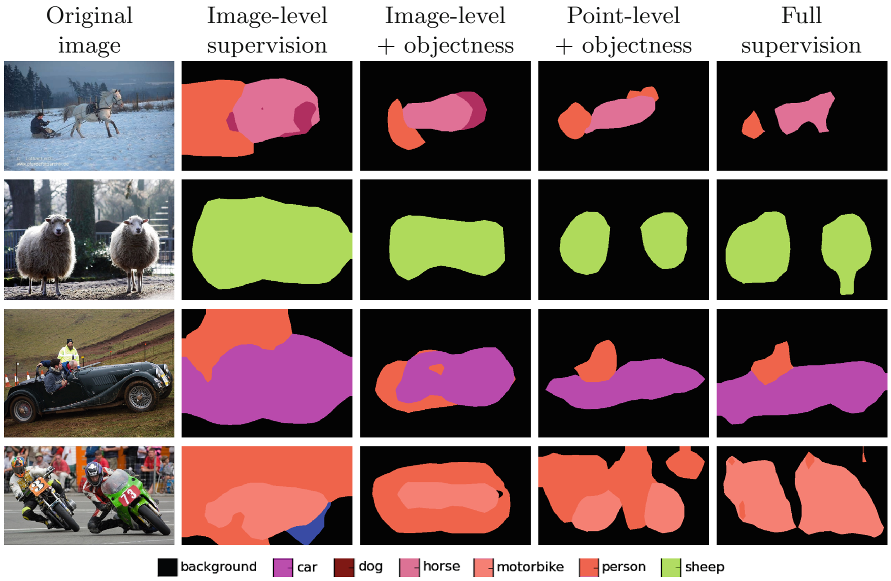

Fig. 4. Qualitative results on the PASCAL VOC 2012 validation set. The model trained with image-level labels usually predicts the correct classes and their general locations, but it over-extends the segmentations. The objectness prior improves the accuracy of the image-level model by helping infer the object extent. Point supervision aids in separating distinct objects (row 2) and classes (row 4) and helps correctly localize the objects (rows 3 and 4). Best viewed in color. (Color figure online)

segmentation accuracy. We use loss $\mathcal{L}_{point}$ of Eq. (3). On average there are only 1.5 supervised pixels per image (as many as classes per image). All other pixels are unsupervised and not considered in the loss. We set $\alpha = 1/n$ where n is the number of supervised pixels on a particular training image. On the PASCAL VOC 2012 validation set, the accuracy of a model trained using $\mathcal{L}_{img}$ is 29.8% mIOU. Adding our point supervision improves accuracy by 5.3% to 35.1% mIOU (row 3 in Table 1).

Effect of Objectness Prior. One issue with training models with very few or no supervised pixels is the difficulty of inferring the full extent of the object. With image-level labels, the model tends to learn that objects occupy a much greater area than they actually do (second column of Fig. 4). We introduce the objectness prior in the loss using Eq. (4) to aid the model in correctly predicting the extent of objects (third column on Fig. 4). This improves segmentation accuracy: when supervised only with image-level labels, the *Img* model obtained 29.8 % mIOU, and the *Img* + *Obj* model improves to 32.2 % mIOU.

Effect of Combining Point-Level Supervision and Objectness. The effect of the objectness prior is even more apparent when used together with point-level supervision. When supervised with 1*Point*, the *Img* model achieves 35.1 % mIOU, and the *Img* + *Obj* model improves to 42.7 % mIOU (rows 3 and 4 in

Table 1. Results on the PASCAL VOC 2012 validation set, including both annotation time (second column) and accuracy of the model (last column). Top, middle and bottom correspond to Sects. 5.2, 5.3 and 5.4 respectively.

Supervision	Time (s)	Model	mIOU (%)
Image-level labels	20.0	*Img*	29.8
Image-level labels	20.3	*Img* + *Obj*	32.2
1*Point*	22.1	*Img*	35.1
1*Point*	22.4	*Img* + *Obj*	42.7
AllPoints	23.6	*Img* + *Obj*	42.7
AllPoints (weighted)	23.5	*Img* + *Obj*	43.4
1*Point* (3 annotators)	29.6	*Img* + *Obj*	43.8
1*Point* (random annotators)	22.4	*Img* + *Obj*	42.8 – 43.8
1*Point* (random points)	240	*Img* + *Obj*	46.1
Full supervision	239.7	*Img*	58.3
Hybrid approach	24.5	*Img* + *Obj*	53.1
1 squiggle per class	35.2	*Img* + *Obj*	49.1

Table 1). Conversely, when starting from the *Img* + *Obj* image-level model, the effect of a single point of supervision is stronger. Adding just one point per class improves accuracy by 10.5 % from 32.2 % to 42.7 %.

Conclusions. We make two conclusions. First, the objectness prior is very effective for training models with none or very few supervised pixels – and this comes with no additional human supervision cost on the target dataset. Thus, for the rest of the experiments in this paper, whenever not all pixels are labeled (i.e., all but full supervision) we always use *Img* + *Obj* together. Second, our two contributions operate in synergetic ways. The combined effect of both point-level supervision and objectness prior is a +13 % improvement (from 29.8 % to 42.7 % mIOU).

5.3 Point-Level Supervision Variations

Our goal in this section is to build a deeper understanding of the properties of point-level supervision that make it an advantageous form of supervision. Table 1 summarizes our findings and Table 2 shows the per-class accuracy breakdown.

Multiple Instances. Using points on all instances (*AllPoints*) instead of just one point per class (1*Point*) remains at 42.7 % mIOU: the benefit from extra supervision is offset by the confusion introduced by some difficult instances that are annotated. We introduce a weighting factor $\alpha_i = 1/2^r$ in Eq. (3) where r is the ranked order of the point (so the first instance of a class gets weight 1, the second instance gets weight 1/2, etc.). This *AllPoints* (weighted) method improves results by a modest 0.7 % to 43.4 % mIOU.

Table 2. Per-class segmentation accuracy (%) on the PASCAL VOC 2012 validation set. (Top) Models trained with image-level, point supervision and (optionally) an objectness prior described in Sect. 5.2. (Bottom) Models supervised with variations of point-level supervision described in Sect. 5.3.

Model	Bg	Aer	Bic	Bir	Boa	Bot	Bus	Car	Cat	Cha	Cow	Din	Dog	Hor	Mot	Per	Pot	She	Sof	Tra	Tv	Avg
Img	60	25	15	23	21	20	48	36	47	9	34	21	37	32	37	18	24	34	21	40	24	30
$Img + Obj$	**79**	42	20	**39**	33	17	34	39	45	10	35	13	42	34	33	23	19	40	15	38	28	32
$Img + 1Point$	56	25	16	22	20	31	53	34	**53**	8	**41**	**42**	43	**40**	42	46	24	38	**29**	46	30	35
$Img + 1Point + Obj$	78	**49**	**23**	37	**37**	**37**	**57**	**50**	51	**14**	40	41	**50**	38	**51**	**47**	**31**	**48**	28	**49**	**45**	**43**
$AllPoints$	79	49	21	**40**	38	38	50	45	53	17	**43**	40	47	**44**	51	51	22	47	**29**	52	44	43
$AllPoints$ (weighted)	77	48	**23**	38	36	38	57	52	52	13	42	41	50	43	52	46	31	49	28	50	44	43
$1Point$ (3 annot.)	79	**50**	23	39	37	39	60	50	54	15	41	**42**	49	42	52	50	29	49	29	49	44	44
$1Point$ (random)	**80**	49	23	39	**41**	**46**	**60**	**61**	**56**	**18**	38	41	**54**	42	**55**	**57**	**32**	**51**	26	**55**	**45**	**46**

Patches. The segmentation model effectively enforces spatial label smoothness, so increasing the area of supervised pixels by a radius of 2, 5 and 25 pixels around a point has little effect, with 43.0–43.1 % mIOU (not shown in Table 1).

Multiple Annotators. We also collected $1Point$ data from 3 different annotators and used all points during training. This achieved a modest improvement of 1.1 % from 42.7 % to 43.8 %, which does not seem worth the additional annotation cost (29.3 versus 22.1 seconds per image).

Random Annotators. Using the data from multiple annotators, we also ran experiments to estimate the effect of human variance on the accuracy of the model. For each experiment, we randomly selected a different independent annotator to label each image. Three runs achieved 42.8, 43.4, and 43.8 mIOU respectively, as compared to our original result of 42.7 mIOU. This suggests that the variation in the location of the annotators' points does not significantly affect our results. This also further confirms that humans are predictable and consistent in pointing to objects [3,28].

Random Points. An interesting experiment is supervising with one point per class, but randomly sampled on the target object class using per-pixel supervised ground truth annotations (instead of asking humans to click on the object). This improved results over the human points by 3.4 %, from 42.7 % to 46.1 %. This is due to the fact that humans are predictable and consistent in pointing [3,28], which reduces the variety in point-level supervision across instances.

5.4 Incorporating Stronger Supervision

Hybrid Approach with Points and Full Supervision. A fully supervised segmentation model achieves 58.3 % mIOU at a cost of 239.7 seconds per image; recall that a point-level supervised model achieves 42.7 % at a cost of 22.4 seconds per image. We explore the idea of combining the benefits of the high accuracy of full supervision with the low cost of point-level supervision. We train a hybrid segmentation model with a combination of a small number of fully-supervised images (100 images in this experiment), and a large number of point-supervised images (the remaining 10,482 images in PASCAL VOC 2012). This model achieves 53.1 % mIOU, a significant 10.4 % increase in accuracy over the 1*Point* model, falling only 5.2 % behind full supervision. This suggests that the first few fully-supervised images are very important for learning the extent of objects, but afterwards, point-level supervision is quite effective at providing the location of object classes. Importantly, this hybrid model maintains a low annotation time, at an average of only 24.5 seconds per image: $(100 \times 239.7 + 10482 \times 22.4)/(100 + 10482) = 24.5\,\mathrm{s}$, which is 9.8× cheaper than full supervision. We will further explore the tradeoffs between annotation cost and accuracy in Sect. 5.5.

Squiggles. Free-form squiggles are a natural extension of points towards stronger supervision. Squiggle-level supervision annotates a larger number of pixels: we collect an average of 502.7 supervised pixels per image with squiggles, vs. 1.5 with 1*Point*. Like points, squiggles provide a nice tradeoff between accuracy and annotation cost. The squiggle-supervised model achieves 16.9 % higher mIOU than image-level labels and 6.4 % higher mIOU than 1*Point*, at only 1.6–1.7× the cost. However, squiggle-level supervision falls short of the hybrid approach on both annotation time and accuracy: squiggle-level takes a longer 35.2 s compared to 24.5 s for hybrid, and squiggle-level achieves only 49.1 % mIOU compared to the better 53.1 % mIOU with hybrid. This suggests that hybrid supervision combining large-scale point-level annotations with full annotation on a handful of images is a better annotation strategy than squiggle-level annotation.

5.5 Segmentation Accuracy on a Budget

Fixed Budget. Given a fixed annotation time budget, what is the right strategy to obtain the best semantic segmentation model possible? We investigate the problem by fixing the total annotation time to be the $10,582 \times (20.3) = 60\,\mathrm{h}$ that it would take to annotate all the $10,582$ training times with image-level labels. For each supervision method, we then compute the number of images N that it is possible to label in that amount of time, randomly sample N images from the training set, use them to train a segmentation model, and measure the resulting accuracy on the validation set. Table 3 reports both the number of images N and the resulting accuracy of fully supervised (22.1 % mIOU), image-level supervised (29.8 % mIOU), squiggle-level supervised (40.2 % mIOU) and point-level supervised (42.9 % mIOU) model. **Point-level supervision outperforms the other types of supervision on a fixed budget**, providing an optimal tradeoff between annotation time and resulting segmentation accuracy.

Comparisons to Others. For the rest of this section, we use a model trained on all 12,031 training+validation images and evaluate on the PASCAL VOC 2012 *test* set (as opposed to the validation set above) to allow for fair comparison to prior work. Point-level supervision ($Img + 1Point + Obj$) obtains 43.6 %

Table 3. Accuracy of models on the PASCAL VOC 2012 validation set given a fixed budget (and number of images annotated within that budget). Point-level supervision provides the best trade-off between annotation time and accuracy. Details in Sect. 5.5.

Supervision	mIOU (%)
Full (883 imgs)	22.1
Image-level (10,582 imgs)	29.8
Squiggle-level (6,064 imgs)	40.2
Point-level (9,576 imgs)	**42.9**

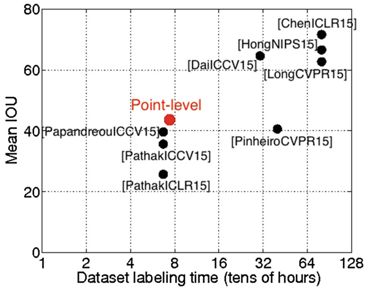

Fig. 5. Results without resource constraints on the PASCAL VOC 2012 *test* set. The x-axis is log-scale.

mIOU on the test set. Figure 5 shows the tradeoffs between annotation time and accuracy of different methods, discussed below.

Unlimited Budget (Strongly Supervised). We compare both the annotation time and accuracy of our point-supervised 1*Point* model with published techniques with much larger annotation budgets, as a reference for what might be achieved by our method if given more resources. Long *et al.* [5] reports 62.2 % mIOU, Hong *et al.* [34] reports 66.6 % mIOU, and Chen *et al.* [38] reports 71.6 % mIOU, but in the fully supervised setting that requires about 800 h of annotation, an order of magnitude more time-consuming than point supervision. Future exploration will reveal whether point-level supervision would outperform a fully supervised algorithm given 800 annotation hours of data.

Small Budget (Weakly Supervised). We also compare to weakly supervised published results. Pathak ICLR *et al.* [23] achieves 25.7 % mIOU, Pathak ICCV *et al.* [6] achieves 35.6 % mIOU, and Papandreou *et al.* [4] achieves 39.6 % mIOU with only image-level labels requiring approximately 67 h of annotation on the 12,301 images (Sect. 4). Pinheiro et al. [25] achieves 40.6 % mIOU but with 400 h of annotations.[4] We improve in accuracy upon all of these methods and achieve 43.6 % with point-level supervision requiring about 79 annotation hours. Note that our baseline model is a significantly simplified version of [4,23]. Incorporating additional features of their methods is likely to further increase our accuracy at no additional cost.

Size Constraint. Finally, we compare against the recent work of [6] which trains with image-level labels but incorporates an additional bit of supervision in the form of object size constraints. They achieve 43.3 % mIOU (omitting the

[4] [25] trains with only image-level annotations but adds 700,000 additional positive ImageNet images and 60,000 background images. We choose not to count the 700,000 freely available images but the additional 60,000 background images they annotated would take an additional $60,000 \times 20$ classes $\times 1\,\text{s} = 333\,\text{h}$. The total annotation time is thus $333 + 67 = 400\,\text{h}$.

CRF post-processing), on par with 43.6 % using point-level supervision. This size constraint should be fast to obtain although annotation times are not reported. These two simple bits of supervision (point-level and size) are complementary and may be used together effectively in the future.

6 Conclusions

We propose a new time-efficient supervision approach for semantic image segmentation based on humans pointing to objects. We show that this method enables training more accurate segmentation models than other popular forms of supervision when given the same annotation time budget. In addition, we introduce an objectness prior directly in the loss function of our CNN to help infer the extent of the object. We demonstrated the effectiveness of our approach by evaluating on the PASCAL VOC 2012 dataset. We hope that future large-scale semantic segmentation efforts will consider using the point-level supervision we have proposed, building upon our released dataset and annotation interfaces.

Acknowledgments. We would like to thank Evan Shelhamer for helping us set up the baseline model of [5], as well as all the other Caffe developers. We also thank Lamberto Ballan, Michelle Greene, Anca Dragan, and Jon Krause.

V. Ferrari was supported by the ERC Starting Grant VisCul. L. Fei-Fei was supported by an ONR-MURI grant. GPUs were graciously donated by NVIDIA.

References

1. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR (2014)
2. Wang, T., Han, B., Collomosse, J.: TouchCut: fast image and video segmentation using single-touch interaction. Comput. Vis. Image Underst. **120**, 14–30 (2014)
3. Clark, H.H.: Coordinating with each other in a material world. Discourse Stud. **7**(4–5), 507–525 (2005)
4. Papandreou, G., Chen, L.C., Murphy, K., Yuille, A.L.: Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In: ICCV (2015)
5. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
6. Pathak, D., Krähenbühl, P., Darrell, T.: Constrained convolutional neural networks for weakly supervised segmentation. In: ICCV (2015)
7. Russakovsky, O., Deng, J., et al.: ImageNet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)
8. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
9. Merrill, D., Maes, P.: Augmenting looking, pointing and reaching gestures to enhance the searching and browsing of physical objects. In: LaMarca, A., Langheinrich, M., Truong, K.N. (eds.) Pervasive 2007. LNCS, vol. 4480, pp. 1–18. Springer, Heidelberg (2007). doi:10.1007/978-3-540-72037-9_1

10. Hild, M., Hashimoto, M., Yoshida, K.: Object recognition via recognition of finger pointing actions. In: Image Analysis and Processing, pp. 88–93 (2003)
11. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. TPAMI **35**(8), 1915–1929 (2013)
12. Gould, S.: Multiclass pixel labeling with non-local matching constraints. In: CVPR (2012)
13. Jain, S.D., Grauman, K.: Predicting sufficient annotation strength for interactive foreground segmentation. In: ICCV, December 2013
14. Guillaumin, M., Kuettel, D., Ferrari, V.: ImageNet auto-annotation with segmentation propagation. IJCV **110**(3), 328–348 (2014)
15. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: interactive foreground extraction using iterated graph cuts. In: ACM SIGGRAPH (2004)
16. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollr, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: ECCV (2014)
17. Xu, J., Schwing, A.G., Urtasun, R.: Learning to segment under various forms of weak supervision. In: CVPR (2015)
18. Lin, D., Dai, J., Jia, J., He, K., Sun, J.: ScribbleSup: scribble-supervised convolutional networks for semantic segmentation. In: CVPR (2016)
19. Bell, S., Upchurch, P., Snavely, N., Bala, K.: Material recognition in the wild with the materials in context database. In: CVPR (2015)
20. Vezhnevets, A., Ferrari, V., Buhmann, J.: Weakly supervised semantic segmentation with a multi-image model. In: ICCV (2011)
21. Vezhnevets, A., Ferrari, V., Buhmann, J.: Weakly supervised structured output learning for semantic segmentation. In: CVPR (2012)
22. Song, H.O., Girshick, R., Jegelka, S., Mairal, J., Harchaoui, Z., Darrell, T.: On learning to localize objects with minimal supervision. In: ICML (2014)
23. Pathak, D., Shelhamer, E., Long, J., Darrell, T.: Fully convolutional multi-class multiple instance learning. In: ICLR (2015)
24. Xu, J., Schwing, A.G., Urtasun, R.: Tell me what you see and i will show you where it is. In: CVPR (2014)
25. Pinheiro, P.O., Collobert, R.: From image-level to pixel-level labeling with convolutional networks. In: CVPR (2015)
26. Papadopoulos, D.P., Clarke, A.D.F., Keller, F., Ferrari, V.: Training object class detectors from eye tracking data. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 361–376. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10602-1_24
27. Ahmed, E., Cohen, S., Price, B.: Semantic object selection. In: CVPR (2014)
28. Firestone, C., Scholl, B.J.: Please tap the shape, anywhere you like: shape skeletons in human vision revealed by an exceedingly simple measure. Psychol. Sci. **25**(2), 377–386 (2014)
29. Sauppé, A., Mutlu, B.: Robot deictics: how gesture and context shape referential communication. In: Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction (2014)
30. Alexe, B., Deselares, T., Ferrari, V.: Measuring the objectness of image windows. PAMI **34**(11), 2189–2202 (2012)
31. Carreira, J., Sminchisescu, C.: Constrained parametric min-cuts for automatic object segmentation. In: CVPR (2010)
32. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8695, pp. 297–312. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10584-0_20

33. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
34. Hong, S., Noh, H., Han, B.: Decoupled deep neural network for semi-supervised semantic segmentation. In: NIPS (2015)
35. Dai, J., He, K., Sun, J.: Boxsup: exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: ICCV (2015)
36. Chai, Y., Lempitsky, V., Zisserman, A.: BiCoS: a bi-level co-segmentation method for image classification. In: CVPR (2011)
37. Joulin, A., Bach, F., Ponce, J.: Discriminative clustering for image cosegmentation. In: CVPR (2010)
38. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. In: ICLR (2015)
39. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: ICCV (2015)
40. Hariharan, B., Arbelaez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: ICCV (2011)
41. Russakovsky, O., Li, L.J., Fei-Fei, L.: Best of both worlds: human-machine collaboration for object annotation. In: CVPR (2015)
42. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia. ACM (2014)