Block Annotation: Better Image Annotation with Sub-Image Decomposition

Hubert Lin
Cornell University

Paul Upchurch
Cornell University

Kavita Bala
Cornell University

Abstract

Image datasets with high-quality pixel-level annotations are valuable for semantic segmentation: labelling every pixel in an image ensures that rare classes and small objects are annotated. However, full-image annotations are expensive, with experts spending up to 90 minutes per image. We propose block sub-image annotation as a replacement for full-image annotation. Despite the attention cost of frequent task switching, we find that block annotations can be crowdsourced at higher quality compared to full-image annotation with equal monetary cost using existing annotation tools developed for full-image annotation. Surprisingly, we find that 50% pixels annotated with blocks allows semantic segmentation to achieve equivalent performance to 100% pixels annotated. Furthermore, as little as 12% of pixels annotated allows performance as high as 98% of the performance with dense annotation. In weakly-supervised settings, block annotation outperforms existing methods by 3-4% (absolute) given equivalent annotation time. To recover the necessary global structure for applications such as characterizing spatial context and affordance relationships, we propose an effective method to inpaint block-annotated images with high-quality labels without additional human effort. As such, fewer annotations can also be used for these applications compared to full-image annotation.

1. Introduction

Recent large-scale computer vision datasets place a heavy emphasis on high-quality fully dense annotations (in which over 90% of the pixels are labelled) for hundreds of thousands of images. Dense annotations are valuable for both semantic segmentation and applications beyond segmentation such as characterizing spatial context and affordance relationships [11, 23]. The long-tail distribution of classes means it is difficult to gather annotations for rare classes, especially if these classes are difficult to segment. Annotating every pixel in an image ensures that pixels corresponding to rare classes or small objects are labelled. Dense annotations also capture pixels that form the boundary between classes. For applications such as understanding spatial context between classes or affordance relationships, dense annotations are required for principled conclusions to be drawn. In the past, polygon annotation tools have enabled partially dense annotations (in which small semantic regions are densely annotated) to be crowdsourced at scale with public crowd workers. These tools paved the way for the cost-effective creation of large-scale partially dense datasets such as [8, 37]. Despite the success of these annotation tools, fully dense datasets have relied extensively on expensive expert annotators [60, 14, 41, 64, 42] and private crowdworkers [11].

We propose annotation of small blocks of pixels as a stand-in replacement for full-image annotation (figure 1). We find that these annotations can be effectively gathered by crowdworkers, and that annotation of a sparse number of blocks per image can train a high performance segmentation network. We further show these sparsely annotated images can be extended automatically to full-image annotations.

We show block annotation has:

- **Wide applicability.** (Section 3) Block annotations can be effectively crowdsourced at higher quality compared to full annotation. It is easy to implement and works with existing advances in image annotation.
- **Cost-efficient Design.** (Section 3) Block annotation reflects a cost-efficient design paradigm (while current research focuses on reducing annotation time). This is reminiscent of gamification and citizen science where enjoyable tasks lead to low-cost high-engagement work.
- **Complex Region Annotation.** (Section 3) Block annotation shifts focus from categorical regions to spatial regions. When annotating categorical regions, workers segment simple objects before complex objects. With spatial regions, informative complex regions are forced to be annotated.
- **Weakly-Supervised Performance.** (Section 4) Block annotation is competitive in weakly-supervised settings, outperforming existing methods by 3-4% (absolute) given equivalent annotation time.
- **Scalable Performance.** (Section 4) Full-supervision performance is achieved by annotating 50% of blocks per image. Thus, blocks can be annotated until desired performance is achieved, in contrast to methods such as scribbles.
- **Scalable Structure.** (Section 5) Block-annotated images can be effectively inpainted with high quality labels without additional human effort.

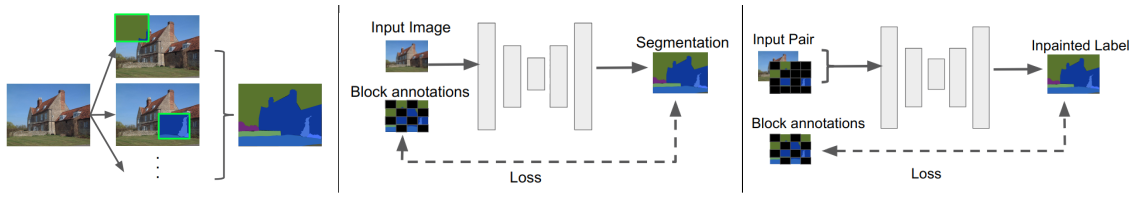


Figure 1: (a) Sub-image block annotations are more effective to gather than full-image annotations (b) Training on sparse block annotations enables semantic segmentation performance equivalent to full-image annotations (c) Block labels can be inpainted with high-quality labels.

2. Related Work

In this section we review recent works on pixel-level annotation in three areas: human annotation, and human-machine annotation, and dense segmentation with weak supervision.

Human Annotation. Manual labeling of every pixel is impractical for large-scale datasets. A successful method is to have crowdsource workers segment polygonal regions to click on boundaries. Employing crowdsource workers offers its own set of challenges with quality control and task design [56, 8, 55]. Although large-scale public crowdsourcing can be successful [37] recent benchmark datasets have resorted to in-house expert annotators [14, 43]. Annotation time can be reduced through improvements such as autopan, zoom [8] and shared polygon boundaries [60]. Polygon segmentation can be augmented by painted labels on superpixel groups [11] and Bezier curves [60]. Pixel-level labels for images can also be obtained by (1) constructing a 3D scene from an image collection, (2) grouping and labeling 3D shapes and (3) propagating shape labels to image pixels [39]. In our work, we investigate sub-image polygon annotation, which can be further combined with other methods (sec. 3.)

Human-Machine Annotation. Complex boundaries are time-consuming to trace manually. In these cases the cost of pixel-level annotation can be reduced by automating a portion of the task. Matting and object selection [50, 33, 34, 6, 58, 57, 10, 30, 59] generate tight boundaries from loosely annotated boundaries or few inside/outside clicks and scribbles. [44, 38] introduced a predictive method which automatically infers a foreground mask from 4 boundary clicks, and was extended to full-image segmentation in [2]. The number of boundary clicks was further reduced to as few as one by [1]. Predictive methods require an additional human verification step since the machine mutates the original human annotation. The additional step can be avoided with an online method. However, online methods (e.g., [1, 30, 2]) have higher requirements since the algorithm must be translated into the web browser setting and the worker's machine must be powerful enough to run the algorithm[1]. Alternatively, automatic proposals can be generated for humans to manipulate: [5] generates segments, [4] generates a set of matting layers, [61] generates superpixel labels, and [47] generates boundary fragments. In our work, we show that human-annotated blocks can be extended automatically into dense annotations (sec. 5), and we discuss how other human-machine methods can be used with blocks (sec. 3.6).

[1]Offloading online methods onto a cloud service offers a different landscape of higher costs (upfront development and ongoing operation costs).

Weakly-Supervised Dense Segmentation. There are alternatives to training with high-quality densely annotated images which substitute quantity for label quality and/or richness. Previous works have used low-quality pixel-level annotations [65], bounding boxes [45, 28, 49], point-clicks [7], scribbles [7, 36], image-level class labels [45, 53, 3], image-level text descriptions [24] and unlabeled related web videos [24] to train semantic segmentation networks. Combining weak annotations with small amounts of high-quality dense annotation is another strategy for reducing cost [9, 26]. [52] proposes a two-stage approach where image-level class labels are automatically converted into pixel-level masks which are used to train a semantic segmentation network. We find a small number of sub-image block annotations is a competitive form of weak supervision (sec. 4.3).

3. Block Annotation

Sub-image block annotation is composed of three stages: (1) Given an image I, select a small spatial region I'; (2) Annotate I' with pixel-level labels; (3) Repeat (with different I') until I is sufficiently annotated. In this paper, we explore the case where I' is rectangular, and focus on the use of existing pixel-level annotation tools.

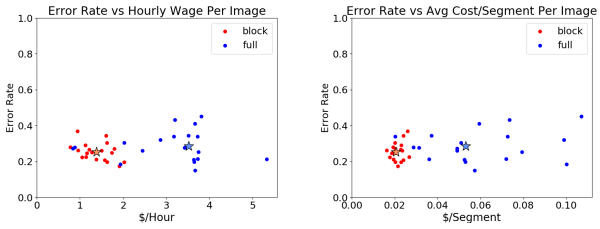
Can block annotations be gathered as effectively as full-image annotations with existing tools? In section 3.1, we show our annotation interface. In section 3.2, we explore the quality of block annotation versus full-image annotation. In section 3.3, we examine block annotation for a real-world dataset. In section 3.4, we discuss the cost of block annotation and show worker feedback. In section 3.5, we discuss how blocks for annotation can be selected in practice. Finally, in section 3.6 we discuss the compatibility of block annotation with existing annotation methods.

3.1. Annotation Interface

Our block annotation interface is given in figure 2 and implemented with existing tools [8]. For full image annotation, the highlighted block covers the entire image. Studies are deployed on Amazon Mechanical Turk.

(a) Highlighted block. (b) Finished block annotation.

Figure 2: **Block Annotation UI**. Annotators are given one highlighted block to annotate with the remainder of the image as context.



(a) (b)

Figure 3: **Annotation error rate** for block and full annotation. Each point represents one image. The same set of images are both block annotated and full-image annotated. The stars represent the centroid (median). Cost/time include estimated cost/time to assign labels for each segment [8]. **Lower-left is better**. With block annotation, workers (a) choose to work for lower wages and (b) segment more regions for less pay per region. The overall quality is higher for block annotation.

.

3.2. Quality of Block Annotation

We explore the quality of block annotations compared to full-image annotations on a synthetic dataset. How does the quality and cost compare between block and full annotations? We find that *the average quality for block-annotated images is higher while the total monetary cost is about the same*. The average quality of block annotations is consistently higher including for small regions (e.g. fig 4). The overall block annotation error is 12% lower than full annotation. For regions smaller than 0.5% of the image, the block annotation error is 6% lower. In figure 3, the cost and quality of block versus full image annotation is shown. Remarkably, we find that *workers are willing to work on block annotation tasks for a significantly lower hourly wage. This indicates that block annotation is more intrinsically palatable for crowdworkers*, in line with [27] which shows task design can influence quality of work. Moreover, workers are more likely to over-segment objects with respect to ground truth (e.g. individual cushions on a couch, handles on cabinets) with block annotation tasks. Note that block boundaries may also divide semantic regions. Table 1 contains additional statistics. Despite similar costs to annotate an image in blocks or in full, we show in section 4 that competitive performance is achieved with less than half of the blocks annotated per image.

Study Details. For these experiments, we chose to use a synthetic dataset. While human annotations may contain mistakes, synthetic datasets are generated with known ground truth labels with which annotation error can be computed. The CGIntrinsics dataset [35] contains physically-based renderings of indoor scenes from the SUNCG dataset [54, 63]. We use the more realistic CGIntrinsics renderings and the known semantic labels from SUNCG. The labels are categorized according to the NYU40[20] semantic categories. Due to the nature of indoor scenes, the depth and field of view of each image is smaller than outdoor datasets. The reduced complexity means that crowdworkers are able to produce good full-image annotations for this dataset.

We select MTurk workers who are skilled at both full-image annotation and block annotation in a pilot study (a standard quality control practice [8]). The final pool consists of 10 workers. Image difficulty is estimated by counting the maximum number of ground truth segments in a fixed-size sliding window. Windows, mirrors, and void regions are masked out in the images so that workers do not expend effort on visible content for which ground truth labels do not exist (such as objects seen through a window or mirror). We manually cull images that include transparent glass tables which are not visible in the renderings, or doorways through which visible content can be seen but no ground truth labels exist. After filtering, twenty of the one hundred most difficult images are selected. We choose a block size so that an average of 3.5 segments are in each block. This results in 16 blocks per image. For each task, a highlighted rectangle outlines the block to be annotated. We find that workers will annotate up to the inner edge of the highlighted boundary. Therefore, we ensure the edges of the rectangle do not overlap with the region to be annotated.

Workers are paid $0.06[2] per block annotation task and $0.96 per full-image task. Bonuses up to 1.5 times the base pay are awarded to attempt to raise the effective hourly wage for difficult tasks to $4 / hr. Our results show that workers

	Block	Full
Error	0.253	0.286
Error (small regions)	0.636	0.677
$ / hr	$1.40 / hr	$3.12 / hr
Total cost	$2.00	$2.05
Total cost (median)	$1.99	$2.23
# segments	95.68	38.95
$ / segment	$0.0215	$0.0595

Table 1: **Block vs Full Annotation**. Average statistics per image.

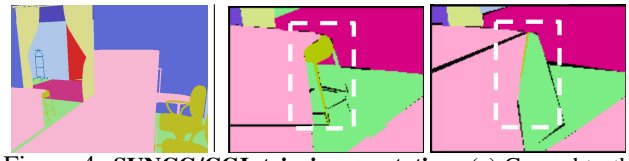


Figure 4: **SUNCG/CGIntrinsics annotation**. (a) Ground truth. (b) Block annotation (zoomed-in) (c) Full annotation (zoomed-in). White dotted box highlights an example where block annotation qualitatively outperforms full annotation. More in supplemental.

[2]$ refers to USD in throughout this paper.

	Block (Crowd)	Full (Expert [14, 42])
$ / Task	$0.13	-
Time / Task	2 min	1.5 hr

Table 2: **Real-world cost of annotation.** Cost evaluated on Cityscapes. Each block is annotated by MTurk workers. Full-image is annotated by experts in [14]. Note: [14] annotates instance segments. See table 1 for crowd-to-crowd comparison.

are willing to work on block annotation tasks beyond the time threshold for bonuses, effectively producing work for an hourly wage significantly lower than the intended $4 / hr. On the other hand, workers do not often exhibit this behavior with full annotation tasks. Different workers may work on different blocks belonging to the same image. We use two forms of quality control: (1) annotations must contain a number of segments greater than 25% of the known number of ground-truth segments for that task and (2) annotations cannot be submitted until at least 10 seconds / 3 minutes (block / full) have passed. All submissions satsifying these conditions are accepted during the user study. For an overview of QA methods, please refer to [48]. Labels are assigned by majority ground-truth voting, with cost estimated from [8].

To evaluate the quality of annotations in an image with K classes, we measure the class-balanced error rate (class-balanced Jaccard distance):

$$error\,rate = \frac{1}{K}\sum_{c=1}^{K}\frac{(FP_c + FN_c)}{(TP_c + FP_c + FN_c)} \qquad (1)$$
$$= 1 - mIOU$$

3.3. Viability of Real-World Block Annotation

How does block annotation fare with a real-world non-synthetic dataset? To study the viability of block annotating real-world datasets with scalable crowdsourcing, we ask crowdworkers to annotate blocks from images in Cityscapes [14]. We choose Cityscapes for the annotation complexity of its scenes – 1.5 hours of expert annotation effort is required per image. In contrast, other datasets such as [41, 11] require less than 20 minutes of annotation effort per image. We expect crowd work to be worse than expert work, so it is a surprisingly positive result that the quality of the crowdsourced segments are visually comparable to the expert Cityscapes segments (figure 5). Some crowdsourced segments are very high quality. We find that 47% of blocks have more crowd segments than expert segments (20% have fewer segments, and 33% have the same # of segments). A summary of the cost is given in table 2 which compares public crowdworkers to trained experts. It is feasible block annotation time will decrease with expert training. Given 100 uniformly sized blocks per image, we ask an expert to create equal-quality block and full annotations; we find one block is 1.56% of the effort of a full image.

Study Details. We searched for workers who produce high-quality work in a pilot study and found a set of 7 workers. These workers were found within a hundred pilot HITs

Figure 5: **Crowdsourced vs expert segments**. Crowdsourced block-annotated segments are compared to expert Cityscapes segments. Crowdsourced segments are colored for easier comparison. Top-left is a high-quality example. See supplemental for more.

(for a total cost of $4). We approved all of their submissions during the user study. We do not restrict workers from annotating outside of the block, and we do not force workers to densely annotate the block. We do not include the use of sentinels or tutorials as in [8].

Thirteen randomly selected validation images from Cityscapes are annotated by crowdworkers. Each image is divided into 100 uniformly shaped blocks. A total of 650 (50 per image) are annotated in random order. Workers are paid $0.06 per task. Workers are automatically awarded bonuses so that the effective hourly wage at least $5 for each block, with bonuses capped at $0.24 to prevent abuse. For one block, the total base payout is $0.06 with an average of $0.0636 in bonuses over 93 seconds of active work. On average, each annotated block contains 3.5 segments. Assigning class labels will cost an additional $0.01 and 26 seconds [8]. To be consistent with Cityscapes, we instruct workers to not segment windows, powerlines, or small regions of sky between leaves. However, workers will occasionally choose to do so and submit higher quality segments than required.

3.4. Annotation Cost and Worker Feedback

Our costs (tables 1, 2) are aligned with existing large-scale studies. Large-scale datasets [8, 37] show that cultivating good workers produces high quality data at low cost. Table 2 of [21] reports a median wage of $1.77/hr to $2.11/hr; the median MTurk wage in India is $1.43/hr [22]. For "image transcription", the median wage is $1.13/hr over 150K tasks.

Workers gave overwhelmingly positive feedback for block annotation (table 3), and we found that some workers would reserve hundreds of block annotation tasks at once. Only 3 out of the 57 workers who successfully completed at least one pilot or user study task requested higher pay. In contrast, our pilot studies showed that *workers are unwilling to accept full-image annotation tasks if the payment is reduced to match the wage of block annotation.* We conjecture that task enjoyment leads to long term high-quality output (c.f. [27]).

	"Nice" "Good" "Great"	"Fun" "Happy"	"Easy"	"Okay"	Release More HITs	Increase Pay
#	8	5	4	2	2	3

Table 3: **Block annotation worker feedback**. Free-form responses are aggregated over SUNCG and Cityscapes experiments, and collected at most once per worker. All 24 sentiments across all 19 worker responses are summarized.

3.5. Block Selection

Our experiments show that workers are comfortable annotating between 3 to 6 segments per block. Therefore, block size can be selected by picking a size such that the average number of segments per block falls in this range. For a novel dataset, this can be done fully labelling several samples and producing an estimate from the fully labelled samples. Without priors on spatial distribution of rare classes or difficult samples within an image, a checkerboard or pseudo-checkerboard pattern of blocks focuses attention (across different tasks) uniformly across the image. Far apart pixels within an image are less correlated than neighboring pixels. Therefore, it is good to sample blocks that are spread out to encourage pixel diversity within images.

3.6. Compatibility with Existing Annotation Methods

Block annotation is compatible with many annotation tools and innovations besides polygon boundary annotation.

Point-clicks and Scribbles. Annotations such as point clicks or scribbles are faster to acquire than polygons, which leads to a larger and more varied dataset at the same cost. Combining this with blocks will further increase annotation variety due to the diversity that come from annotating a few blocks in many images over annotating fewer number of images fully. Additionally, [7, 9] show that the most cost-effective method for semantic segmentation is a combination of densely annotated images and a large number of point clicks. The densely annotated images can be replaced by polygon block annotations since they also contain class boundary supervision for the segmentation network.

Superpixels. Superpixel annotations enable workers to mark a group of visually-related pixels at once [11]. This can reduce the annotation time for background regions and objects with complex boundaries. Superpixel annotation can be easily deployed to our block annotation setting.

Polygon Boundary Sharing. Boundary sharing reuses existing boundaries so that workers do not need to trace each boundary twice [60]. This approach can be easily deployed in our block annotation setting.

Curves. Bezier tools allow workers to quickly annotate curves [60]. It can be easily deployed in our block annotation setting but it may be less effective on long curves since each part of the curve must be fit separately.

Interactive Segmentation. Recent advances in interactive segmentation (e.g., [1, 38, 2]) utilize neural networks to convert sparse human inputs into high quality segments. For novel domains without large-scale training data, block-annotated images can act as cost-efficient seed data to train these models. Once trained, these methods can be applied directly to each block, although further analysis should be conducted to explore the efficiency of such an approach due to block boundaries splitting semantic regions.

4. Segmentation Performance

How well do block annotations serve as training data for semantic segmentation? In section 4.1, the experimental setup is summarized. In section 4.2, we evaluate the effectiveness of block annotations for semantic segmentation. In section 4.3, we compare block annotation with existing weakly supervised segmentation methods.

4.1. Experimental Setup

Pixel Budget. We vary the "pixel budget" in our experiments to explore segmentation performance across a range available annotated pixels. "Pixel budget" refers to the % of pixels annotated across the training dataset, which can be controlled by varying the number of annotated images, the number annotated blocks per image, and the size of blocks per image. Our block sizes are fixed in our experiments.

Block Size. We divide images into a 10-by-10 grid for our experiments.

Block Selection. We experiment with two block selection strategies: (a) Checkerboard annotation and (b) Pseudo-checkerboard annotation. Checkerboard annotation means that every other block in a variable number of images are annotated. Pseudo-checkerboard annotation means that every N blocks are annotated in every image, where N is $\frac{\texttt{\# pixels in dataset}}{\texttt{pixel budget}}$. For example, with a pixel budget equivalent to 25% of the dataset, every fourth block is annotated for the entire dataset. At pixel budget 50%, checkerboard and pseudo-checkerboard are identical.

For the remainder of the paper, "Block-X%" refers to pseudo-checkerboard annotation in which X% of the blocks per image are annotated.

Sementation Model. We use DeepLabv3+[13] initialized with the official pretrained checkpoint (pretrained on ImageNet [16] + MSCOCO [37] + Pascal VOC [17]). The network is trained for a fixed number of epochs. See supplemental for additional details.

Datasets. Cityscapes is a dataset with ground truth annotations for 19 classes with 2975 training images and 500 validation images. ADE20K contains ground truth annotations for 150 classes with 20210 training images and 2000 validation images. These datasets are chosen for their high

Figure 6: **Semantic segmentation performance**. Training images are annotated with different pixel budgets. Pseudo-checkerboard block annotation outperforms checkerboard and full annotation.

quality dense ground truth annotations and for their differences in number of images / classes and types of scenes represented. The block annotations are synthetically generated from the existing annotations.

4.2. Evaluation

Blocks vs Full Image. How does block annotation compare to full-image annotation for semantic segmentation? We plot the mIOU achieved when trained on a set of annotations against pixel budget in figure 6.

For both Cityscapes and ADE20K, *block annotation significantly outperforms full-image annotation*. The performance gap widens as the pixel budget is decreased – at pixel budget 12%, the reduction in error from full annotation to block annotation is 13% (10%) Cityscapes (ADE20K). Our results indicate that the quantity of annotated images is more valuable than the quantity of annotations per image. The pseudo-checkerboard block selection pattern consistently outperforms the checkerboard block selection pattern and full annotation. For any pixel budget, pseudo-checkerboard block annotation annotates fewer pixels per image which means more images are annotated.

	Optimal (Full)	Block-50%	Block-12%
Cityscapes	77.7	77.7	74.6
ADE20K	37.4	37.2	36.1

Table 4: **Semantic segmentation performance** when trained on all images. Training with block annotations uses fewer annotated pixels than full annotation but achieves equivalent performance.

Blocks vs Optimal Performance. How many blocks need to be annotated for segmentation performance to approach the performance achieved by training on full-image annotations for the entire dataset? In table 4, we show results when the network is trained on the full dataset compared to pseudo-checkerboard blocks. Remarkably, we find that *checkerboard blocks with 50% pixel budget allow the network to achieve similar performance to the full dataset with 100% pixel budget*, indicating that at least 50% of the pixels in Cityscapes and ADE20K are redundant for learning semantic segmentation. Furthermore, with only 12% of the pixels in the dataset annotated, relative error in segmentation performance is within 12%/2% of the optimal for Cityscapes/ADE20K. These results suggest that fewer than 50% of the blocks in an image need to be annotated for training semantic segmentation, reducing the cost of annotation reported in section 3.

4.3. Weakly Supervised Segmentation Comparison

Block annotation can be considered a form of weakly supervised annotation where a small number of pixels in an image are labelled. Representative works in this area include [36, 7, 46, 45, 15]. Table 3 of [36] is replicated here (table 5) for reference, and extended with our results. All existing results show performance with a VGG-16 based model. We train a MobileNet based model which has been shown to achieve similar performance to VGG-16 (71.8% vs 71.5% Top-1 accuracy on ImageNet) while requiring fewer computational resources [25, 51]. Our fully-supervised implementation pretrained on ImageNet achieves 69.6% mIOU on Pascal VOC 2012 [17]; in comparison, the reference DeepLab-VGG16 model achieves 68.7% mIOU [12] and the re-implementation in [36] achieves 68.5% mIOU.

Method	Annotations	mIOU (%)
MIL-FCN [46]	Image-level	25.1
WSSL [45]	Image-level	38.2
point sup. [7]	Point	46.1
ScribbleSup [36]	Point	51.6
WSSL [45]	Box	60.6
BoxSup [15]	Box	62.0
ScribbleSup [36]	Scribble	63.1
Ours: Block-1%	Pixel-level Block	61.2
Ours: Block-5%	Pixel-level Block	67.6
Ours: Block-12%	Pixel-level Block	68.4
Full Supervision	Pixel-level Image	69.6

Table 5: **Weakly-supervised segmentation performance**. Evaluated on Pascal VOC 2012 validation set. Original table from [36]. Blocks (N%) indicates N% of image pixels (N pseudo-checkerboard blocks) are labelled.

Performance Comparison. With only 1% of the pixels annotated, block annotation achieves comparable performance to existing weak supervision methods. Based on our results in section 3.2, the cost of annotation for 1% of pixels with blocks will be 100× less than the cost of full-image annotation. Increasing the budget to 5%-12% significantly increases performance. With 12% of pixels annotated with blocks, the segmentation performance (error) is within 98%

Cityscapes	**Ours:** Block (7 min)	Coarse (7 min [14])	Full Supervision (90 min [14])
mIOU (%)	**72.1**	68.8	77.7

Pascal	**Ours:** Block (25 sec)	Scribbles (25 sec [36])	Full Supervision (4 min [41])
mIOU (%)	**67.2**	63.1 [36]	69.6

Table 6: **Weakly-supervised segmentation performance given equal annotation time**. For time comparison of scribbles against other methods, please refer to [36].

(4%) of segmentation performance (error) with 100% of pixels annotated.

Note that block annotations can be directly transformed into gold-standard fully dense annotations by simply gathering more block annotations within an image. This is not feasible with other annotations such as point clicks, scribbles, and bounding boxes. Furthermore, in section 5, we demonstrate a method to transform block annotations into dense annotations without any additional human effort.

Equal Annotation Time Comparison. Given equal annotation time, block annotation significantly outperforms coarse and scribble annotations by ∼3-4% mIOU (table 6). On Pascal, 97% of full-supervision mIOU is achieved with 1/10 annotation time. We convert annotation time to number of annotated blocks as follows. Block annotation may use up to 2.2× the time of full-image annotation. Given an image divided into 100 blocks, an annotation time of T leads to $\frac{T}{0.022F}$ (eq. 5) blocks annotated, where F is the full-image annotation time.

5. Block-Inpainting Annotations

Although block annotations are useful for learning semantic segmentation, the full structure of images is required for many applications. Understanding the spatial context or affordance relationships [11, 23] between classes relies on understanding the role of each pixel in an image. Shape-based retrieval, object counting [32], or co-occurrence relationships [40] also depend on a global understanding of the image. The naive approach to recover pixel-level labels is to use automatic segmentation to predict labels. However, this does not leverage existing annotations to improve the quality of predicted labels. In section 5.1, we propose a method to inpaint block-annotated images by using annotated blocks as context. In section 5.2, we examine the quality of these inpainted annotations.

5.1. Block-Inpainting Model

The goal of the block-inpainting model is to inpaint labels for unannotated blocks given the labels for annotated blocks in an image. For full implementation details and ablation studies, please refer to the supplemental.

Architecture. The block-inpainting model is based on DeepLabv3+. The input layer is modified so that the RGB image, $I \in \mathbb{R}^{h\times w\times 3}$, is concatenated with multichannel "hint" (ala [62]) of 1-0 class labels $W \in \mathbb{R}^{h\times w\times K}$ where K is the number of classes. At inference time, the hint contains known labels for the annotated blocks of an image which serve as context for the inpainting task. Every hidden layer is augmented with dropout which will be used to control quality by estimating epistemic uncertainty [18, 19].

Estimating Uncertainty. Inpainting fills all missing regions without considering the trade off between quantity and quality. Existing datasets have high-quality annotations for 92-94% of pixels [64, 11]. Therefore, we modify our network to produce uncertainty estimates which allow us to explicitly control this trade off. The uncertainty of predictions is correlated with incorrect predictions [31, 29]. Uncertainty is computed by activating dropout at inference time. The predictions are averaged over the g trials giving us $U \in \mathbb{R}^{h\times w}$, a matrix of uncertainty estimates per image. We take the sample standard deviation corresponding to the predicted class for each pixel to be the uncertainty. For each pixel (i, j), the mean softmax vector over g trials is:

$$\boldsymbol{\mu}^{(i,j)} = \frac{\sum_{t=1}^{g} \mathbf{p}^{(i,j)}(y|I, W)}{g} \tag{2}$$

where $\mathbf{p}(y|I, W) \in \mathbb{R}^K$ is the softmax output of the network. The corresponding uncertainty vector is:

$$\mathbf{U}'^{(i,j)} = \sqrt{\frac{\sum_{t=1}^{g} (\mathbf{p}^{(i,j)}(y|I, W) - \boldsymbol{\mu}^{(i,j)})^2}{g-1}} \tag{3}$$

Thus, the uncertainty for each pixel (i, j) is:

$$U^{(i,j)} = U'^{(i,j)}_m, \text{ where } m = \arg\max_k \mu_k^{(i,j)} \tag{4}$$

Training. Block annotations serve both as hints and targets. This means that no additional data (or human annotation effort) is required to train the block-inpainting model. For our experiments, we use (synthetically generated) Block-50% annotations. For each image, half of annotated blocks are randomly selected online at training time to be hints. All of the annotated blocks are used as targets. This encourages the network to "copy-paste" hints in the final output while leveraging the hints as context to inpaint labels for regions where hints are not provided.

5.2. Evaluation

Quality of Inpainted Labels. How good are inpainted labels? We compare labels produced by the block-inpainting network with low $U^{(i,j)}$ against the known human labels in Cityscapes and ADE20K. *The block-inpainting model produces labels whose human-agreement is competitive with that achieved by human annotators.* We inpaint Block-50% annotations in this experiment. At a relative uncertainty

(a) Full human labels

(b) Original image

(c) Inpainted labels (all)

(d) Label agreement (white)

(e) Inpainted labels (<20% relative uncertainty)

(f) Label agreement (white)

Figure 7: **Block-inpainted labels.** Example of human labels vs human Block-50% + inpainted labels. Void labels are masked out.

threshold of 0.2 (0.4) on Cityscapes (ADE20K), over 94% of the pixels are labelled. The mean pixel agreement is 99.8% (98.7%) and the class-balanced error rate is 3.1% (28%). Previous work show that human label agreement across annotators is 66.8% to 73.6% while annotator self-agreement is 82.4% to 97.0% [64, 11]. Human annotators fail to agree in non-trivial fashion – [64] shows that annotator self-agreement fails in three ways: variations in complex boundaries (32%), incorrect naming of ambiguous classes (34%), and failure to segment small objects (34%). In figure 7, a visualization of labels generated by the block-inpainting model is shown. The number of pixel disagreements decreases with a higher uncertainty threshold.

Block Inpainting vs Automatic Segmentation. Consider a scenario in which a small number of pixels in a dataset are annotated, and the remainder are automatically labelled to produce dense annotations. Why should block inpainting be used instead of automatic segmentation? *Full pixel-level labels produced by block inpainting are superior to automatic segmentation.* On Cityscapes, automatic segmentation achieves 78% validation mIOU while block inpainting Block-50% annotations achieves 92% validation mIOU. With Block-12% annotations, automatic segmentation achieves 75% validation mIOU while block inpainting achieves 82% validation mIOU.

Block Selection vs Block-Inpainting Quality How does the checkerboard pattern compare to other block selection strategies as hints to the block-inpainting model? Intuitively, it is easier to infer labels for pixels that are close to pixels with known labels than for pixels that are further away. Consider a scenario in which every other pixel in an image is annotated. Reasonably good labels for the unannotated pixels can be inferred with a simple nearest-neighbors algorithm. In practice, it is impossible to precisely annotate single pixels in an image. However, we can approximate the same properties of labelling every other pixel by labelling every other block instead (i.e., a checkerboard pattern).

	None	Random (Bndy)	Random (Full)	Checker (10x10)	Every oth. pixel
Rel. mIOU	0.77	0.90	0.92	0.95	1.0

Table 7: **Block-inpainting with different types of hints.** "Every other pixel" annotations are infeasible in practice. Relative performance of hints with respect to "every other pixel" hints is shown. Checkerboard blocks outperform no hints, random blocks (only boundaries within blocks), and random blocks (full blocks).

In table 7, we show the block-inpainting model mIOU when different types of hints are given. The rightmost column ("every other pixel") is not feasible to collect in practice. Checkerboard annotations outperform random block annotations even though the network is trained to expect random block hints. Providing only boundary annotations within each block (i.e. annotating pixels within 10 pixels of each boundary in each block) allows the network to achieve nearly the same performance as full block hints. This suggests that the most informative pixels for the block-inpainting model are those near a boundary.

6. Conclusion

In this paper we have introduced block annotation as a replacement for traditional full-image annotation with public crowdworkers. For semantic segmentation, Block-12% offers strong performance at 1/8th of the monetary cost. Block-5% offers competitive weakly-supervised performance at equal annotation time to existing methods. For optimal semantic segmentation performance, or to recover global structure with inpainting, Block-50% should be utilized.

There are many directions for future work. Our crowdworker tasks are similar to full-image annotation tasks so it may be possible to improve the gains with more exploration and development of boundary marking algorithms. We have explored some block patterns and further exploration may reveal even better trade-offs between annotation quality, cost and image variety. Another interesting direction is acquiring instance-level anntoations by merging segments across block boundaries. Active learning can be used to select blocks of rare classes, and workers can be assigned blocks so that annotation difficulty matches worker skill.

Acknowledgements. We acknowledge support from Google, NSF (CHS-1617861 and CHS-1513967), PERISCOPE MURI Contract #N00014-17-1-2699, and NSERC (PGS-D). We thank the reviewers for their constructive comments. We appreciate the efforts of MTurk workers who participated in our user studies.

References

[1] D. Acuna, H. Ling, A. Kar, and S. Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 859–868, 2018. 2, 5

[2] E. Agustsson, J. R. Uijlings, and V. Ferrari. Interactive full image segmentation by considering all regions jointly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11622–11631, 2019. 2, 5

[3] J. Ahn and S. Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. *arXiv preprint arXiv:1803.10464*, 2018. 2

[4] Y. Aksoy, T.-H. Oh, S. Paris, M. Pollefeys, and W. Matusik. Semantic soft segmentation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 37(4):72:1–72:13, 2018. 2

[5] M. Andriluka, J. R. Uijlings, and V. Ferrari. Fluid annotation: a human-machine collaboration interface for full image annotation. *arXiv preprint arXiv:1806.07527*, 2018. 2

[6] X. Bai and G. Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *International journal of computer vision*, 82(2):113–132, 2009. 2

[7] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European Conference on Computer Vision (ECCV)*, pages 549–565. Springer, 2016. 2, 5, 6

[8] S. Bell, P. Upchurch, N. Snavely, and K. Bala. OpenSurfaces: A richly annotated catalog of surface appearance. *ACM Transactions on Graphics (TOG)*, 32(4), 2013. 1, 2, 3, 4

[9] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Material recognition in the wild with the Materials in Context database. *Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 5

[10] A. S. Boroujerdi, M. Khanian, and M. Breuß. Deep interactive region segmentation and captioning. In *Signal-Image Technology & Internet-Based Systems (SITIS), 2017 13th International Conference on*, pages 103–110. IEEE, 2017. 2

[11] H. Caesar, J. Uijlings, and V. Ferrari. COCO-Stuff: Thing and stuff classes in context. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 1, 2, 4, 5, 7, 8

[12] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. 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, 2018. 6

[13] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 5

[14] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes dataset for semantic urban scene understanding. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016. 1, 2, 4, 7

[15] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1635–1643, 2015. 6

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009. 5

[17] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, June 2010. 5, 6

[18] Y. Gal and Z. Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015. 7

[19] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016. 7

[20] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 564–571, 2013. 3

[21] K. Hara, A. Adams, K. Milland, S. Savage, C. Callison-Burch, and J. P. Bigham. A data-driven analysis of workers' earnings on amazon mechanical turk. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 449. ACM, 2018. 4

[22] K. Hara, A. Adams, K. Milland, S. Savage, B. V. Hanrahan, J. P. Bigham, and C. Callison-Burch. Worker demographics and earnings on amazon mechanical turk: An exploratory analysis. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, page LBW1217. ACM, 2019. 4

[23] M. Hassanin, S. Khan, and M. Tahtali. Visual affordance and function understanding: A survey. *arXiv preprint arXiv:1807.06775*, 2018. 1, 7

[24] S. Hong, D. Yeo, S. Kwak, H. Lee, and B. Han. Weakly supervised semantic segmentation using web-crawled videos. *arXiv preprint arXiv:1701.00352*, 2017. 2

[25] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 6

[26] R. Hu, P. Dollár, K. He, T. Darrell, and R. Girshick. Learning to segment every thing. *Cornell University arXiv Institution: Ithaca, NY, USA*, 2017. 2

[27] E. Huang, H. Zhang, D. C. Parkes, K. Z. Gajos, and Y. Chen. Toward automatic task design: a progress report. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 77–85. ACM, 2010. 3, 4

[28] A. Khoreva, R. Benenson, J. H. Hosang, M. Hein, and B. Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, volume 1, page 3, 2017. 2

[29] N. Laptev, J. Yosinski, L. E. Li, and S. Smyl. Time-series extreme event forecasting with neural networks at uber. In *International Conference on Machine Learning*, number 34, pages 1–5, 2017. 7

[30] H. Le, L. Mai, B. Price, S. Cohen, H. Jin, and F. Liu. Interactive boundary prediction for object selection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 18–33, 2018. 2

[31] C. Leibig, V. Allken, M. S. Ayhan, P. Berens, and S. Wahl. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific reports*, 7(1):17816, 2017. 7

[32] V. Lempitsky and A. Zisserman. Learning to count objects in images. In *Advances in neural information processing systems*, pages 1324–1332, 2010. 7

[33] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):228–242, 2008. 2

[34] A. Levin, A. Rav-Acha, and D. Lischinski. Spectral matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(10):1699–1712, 2008. 2

[35] Z. Li and N. Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–387, 2018. 3

[36] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3159–3167, 2016. 2, 6, 7

[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014. 1, 2, 4, 5

[38] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool. Deep extreme cut: From extreme points to object segmentation. *arXiv preprint arXiv:1711.09081*, 2017. 2, 5

[39] P. Marion, P. R. Florence, L. Manuelli, and R. Tedrake. A pipeline for generating ground truth labels for real rgbd data of cluttered scenes. *arXiv preprint arXiv:1707.04796*, 2017. 2

[40] B. Mičušlík and J. Košecká. Semantic segmentation of street scenes by superpixel co-occurrence and 3d geometry. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 625–632. IEEE, 2009. 7

[41] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2014. 1, 4, 7

[42] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, pages 5000–5009, 2017. 1, 4

[43] G. Neuhold, T. Ollmann, S. R. Bulo, and P. Kontschieder. The Mapillary Vistas dataset for semantic understanding of street scenes. In *International Conference on Computer Vision (ICCV)*, pages 22–29, 2017. 2

[44] D. P. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari. Extreme clicking for efficient object annotation. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 4940–4949. IEEE, 2017. 2

[45] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1742–1750, 2015. 2, 6

[46] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. Fully convolutional multi-class multiple instance learning. *arXiv preprint arXiv:1412.7144*, 2014. 6

[47] X. Qin, S. He, Z. Zhang, M. Dehghan, and M. Jagersand. Bylabel: A boundary based semi-automatic image annotation tool. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1804–1813. IEEE, 2018. 2

[48] A. J. Quinn and B. B. Bederson. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1403–1412. ACM, 2011. 4

[49] T. Remez, J. Huang, and M. Brown. Learning to segment via cut-and-paste. *CoRR*, abs/1803.06414, 2018. 2

[50] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, volume 23, pages 309–314. ACM, 2004. 2

[51] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 6

[52] T. Shen, G. Lin, C. Shen, and I. Reid. Bootstrapping the performance of webly supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1363–1371, 2018. 2

[53] Z. Shi, Y. Yang, T. M. Hospedales, and T. Xiang. Weakly-supervised image annotation and segmentation with objects and attributes. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2525–2538, 2017. 2

[54] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[55] P. Upchurch, D. Sedra, A. Mullen, H. Hirsh, and K. Bala. Interactive consensus agreement games for labeling images. In *AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, October 2016. 2

[56] P. Welinder, S. Branson, P. Perona, and S. J. Belongie. The multidimensional wisdom of crowds. In *Advances in neural information processing systems*, pages 2424–2432, 2010. 2

[57] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang. Deep grabcut for object selection. *arXiv preprint arXiv:1707.00243*, 2017. 2

[58] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang. Deep interactive object selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 373–381, 2016. 2

[59] N. Xu, B. L. Price, S. Cohen, and T. S. Huang. Deep image matting. In *CVPR*, volume 2, page 4, 2017. 2

[60] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. BDD100K: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018. 1, 2, 5

[61] L. Zhang, C. Fu, and J. Li. Collaborative annotation of semantic objects in images with multi-granularity supervisions. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 474–482. ACM, 2018. 2

[62] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (TOG)*, 9(4), 2017. 7

[63] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, and T. Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[64] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 4. IEEE, 2017. 1, 7, 8

[65] A. Zlateski, R. Jaroensri, P. Sharma, and F. Durand. On the importance of label quality for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1479–1487, 2018. 2