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.
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 crowdsourcers annotate polygonal regions to click on boundaries. Employing crowdsurcers 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 [30, 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).

**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.

---

\(^1\) Offloading online methods onto a cloud service offers a different landscape of higher costs (upfront development and ongoing operation costs).
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 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 mis-

### Table 1: Block vs Full Annotation

<table>
<thead>
<tr>
<th></th>
<th>Block</th>
<th>Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>0.253</td>
<td>0.286</td>
</tr>
<tr>
<td>Error (small regions)</td>
<td>0.636</td>
<td>0.677</td>
</tr>
<tr>
<td>$ / hr</td>
<td>$1.40 / hr</td>
<td>$3.12 / hr</td>
</tr>
<tr>
<td>Total cost</td>
<td>$2.00</td>
<td>$2.05</td>
</tr>
<tr>
<td>Total cost (median)</td>
<td>$1.99</td>
<td>$2.23</td>
</tr>
<tr>
<td># segments</td>
<td>95.68</td>
<td>38.95</td>
</tr>
<tr>
<td>$ / segment</td>
<td>$0.0215</td>
<td>$0.0595</td>
</tr>
</tbody>
</table>

### 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.

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

<table>
<thead>
<tr>
<th>Block (Crowd)</th>
<th>Full (Expert [14, 42])</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ / Task</td>
<td>$0.13</td>
</tr>
<tr>
<td>Time / Task</td>
<td>2 min</td>
</tr>
<tr>
<td></td>
<td>1.5 hr</td>
</tr>
</tbody>
</table>

Table 2: **Real-world cost of annotation.** Cost evaluated on Cityscapes. Each block is annotated by MTurk workers. Full-image annotation is performed 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 satisfying these conditions are accepted during the user study. For an overview of QA methods, please refer to [38]. 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):

\[
\text{error rate} = \frac{1}{K} \sum_{c=1}^{K} \left( \frac{FP_c + FN_c}{TP_c + FP_c + FN_c} \right) = 1 - mJOU
\]

### 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 workers to be far worse than expert work, so it is a surprisingly positive result that the quality of the crowdsourced segments is visually comparable to work (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]).

| Time / Task   | 2 min                   |
|               | 1.5 hr                  |

### 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]).

orst 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]).
<table>
<thead>
<tr>
<th>“Nice”</th>
<th>“Good”</th>
<th>“Great”</th>
<th>“Fun”</th>
<th>“Easy”</th>
<th>“Okay”</th>
<th>Release</th>
<th>More HITs</th>
<th>Increase Pay</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>8</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

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.** Beziers 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 \# pixels in dataset \text{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
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.

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

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%
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Ours: Block (7 min)</th>
<th>Coarse (7 min [14])</th>
<th>Full Supervision (90 min [14])</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cityscapes</td>
<td>72.1</td>
<td>68.8</td>
<td>77.7</td>
</tr>
<tr>
<td>Pascal</td>
<td>67.2</td>
<td>63.1 [36]</td>
<td>69.6</td>
</tr>
</tbody>
</table>

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.

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 2.2x the time of full-image annotation. Given an image divided into 100 blocks, an annotation time of T leads to $\frac{T}{10}$ (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:

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

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

$$U^{(i,j)} = \sqrt{\frac{1}{g-1} \sum_{t=1}^{g} (p^{(i,j)}(y|I, W) - \mu^{(i,j)})^2}$$

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

$$U_{m}^{(i,j)} = U^{(i,j)}_{m}$$

where $m = \arg \max_{k} \mu_{k}^{(i,j)}$ (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
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).

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 annotations 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


[29] Nikolay Laptev, Jason Yosinski, Li Erran Li, and Slawek Smyl. Time-series extreme event forecasting with neural networks at uber. In International Conference on Machine Learning, number 34, pages 1–5, 2017. 7


