Quality Evaluation Methods for Crowdsourced Image Segmentation

Doris Jung-Lin Lee
University of Illinois,
Urbana-Champaign
jlee782@illinois.edu

Akash Das Sarma
Facebook, Inc.
akashds@fb.com

Aditya Parameswaran
University of Illinois,
Urbana-Champaign
adityagp@illinois.edu

ABSTRACT

Instance-level image segmentation provides rich information crucial for scene understanding in a variety of real-world applications. In this paper, we evaluate multiple crowdsourced algorithms for the image segmentation problem, including novel worker-aggregation-based methods and retrieval-based methods from prior work. We characterize the different types of worker errors observed in crowdsourced segmentation, and present a clustering algorithm as a preprocessing step that is able to capture and eliminate errors arising due to workers having different semantic perspectives. We demonstrate that aggregation-based algorithms attain higher accuracies than existing retrieval-based approaches, while scaling better with increasing numbers of worker segmentations.

1 INTRODUCTION

Precise, instance-level object segmentation is crucial for identifying and tracking objects in a variety of real-world emergent applications of autonomy, including robotics [13], image organization and retrieval [21], and medicine [10]. To this end, there has been a lot of work on employing crowdsourcing to generate training data for segmentation, including Pascal-VOC [6], LabelMe [18], OpenSurfaces [3], and MS-COCO [11]. Unfortunately, raw data collected from the crowd is known to be noisy due to varying degrees of worker skills, attention, and motivation [2, 20].

To deal with these challenges, many have employed heuristics indicative of crowdsourced segmentation quality to pick the best worker-provided segmentation [17, 19]. However, this approach ends up discarding the majority of the worker segmentations and is limited by what the best worker can do. The contributions of this paper is as follows:

- We introduce a novel class of *aggregation-based* methods that incorporates portions of segmentations from multiple workers into a combined one described in Section 4. By overlaying worker segmentations on top of each other, we can decompose the image into non-overlapping tiles, where each tile has some workers who believe this tile belongs to the object, and others who do not. Each tile can be treated as an independent boolean question, deriving an answer from a worker—does this tile belong to the object or not, following which we may be able to apply Expectation-Maximization (EM) [5] to derive maximum likelihood tiles and worker accuracies, a greedy approach for tile picking based on worker fraction votes, and simple majority vote aggregation.
- To our surprise, despite the intuitive simplicity of aggregation-based methods, we have not seen this class of algorithms described or evaluated in prior work. We evaluate this class of algorithms against existing methods in Section 7 and found that it performs much better than existing approaches
- We formally characterize the types of worker error in crowdsourced image segmentation in Section 3 and describe a well-known multiple perspective issue in crowdsourced image segmentation [8, 12, 17], where workers often segment the wrong objects or erroneously include or exclude large semantically-ambiguous portions of an object in the resulting segmentation.To address this issue, in Section 5, we develop a clustering-based solution which can be applied as a preprocessing step to any quality evaluation methods.

2 RELATED WORK

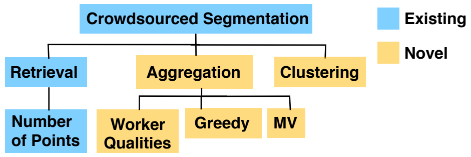

Figure 1: Taxonomy of quality evaluation algorithms for crowdsourced segmentation, including existing methods (blue) and a novel class of algorithms proposed in this paper (yellow).

As shown in Figure 1, quality evaluation methods for crowdsourced segmentation can be classified into two categories:

Retrieval-based methods pick the "best" worker segmentation based on some scoring criteria that evaluates the quality of each segmentation, including vision information [14, 19], and click-stream behavior [4, 15, 17].

Aggregation-based methods combine multiple worker segmentations to produce a final segmentation that is not restricted to any single worker segmentation. An aggregation-based majority vote approach was employed in Sameki et al. [15] to create an expert-established gold standard for characterizing their dataset and algorithmic accuracies, rather than for segmentation quality evaluation as described here.

Vision-based methods There has been a lot of prior work in segmenting objects based on color boundaries[7, 22]. These approaches, however, are typically non-exact, and far from robust. Furthermore, while they segment the entire image into several disjoint pieces, they do not serve to identify objects. Another class of prior works aim to segment specific semantic objects for objects of a specified type (e.g. cars, people)[1, 12, 23]. Object segmentation using purely automated techniques would require training computer vision models on specific object types.

Orthogonal methods to improve segmentation quality include periodic verification [6, 12], specialized interfaces [16], and vision-based supervision [9, 14]. These methods could be used for quality improvement on top of any of the algorithms in this paper. Since these policy-based methods are often interface-dependent or require expensive expert-drawn ground-truth annotations or vision information, their results are not easily reproducible.

3 ERROR ANALYSIS

On collecting and analyzing a number of crowdsourced segmentations (described in Section 6.1), we found that common worker segmentation errors can be classified into three types:

- **Semantic Ambiguity:** workers have differing opinions on whether particular regions belong to an object (Figure 2 left: annotations around 'flower and vase' when 'vase' is requested);
- **Semantic Mistake:** workers annotate the wrong object entirely (Figure 2 right: annotations around 'turtle' and 'monitor' when 'computer' is requested.)
- **Boundary Imperfection:** workers make unintentional mistakes while drawing the boundaries, either due to low image resolution, small area of the object, or lack of drawing skills (Figure 3 left: imprecision around the 'dog' object).

Semantic ambiguity and mistakes have also been observed in prior work [8, 12, 17], which noted that disagreement in worker responses can come from questions that are ambiguous or difficult to answer, such as segmenting a individual person from a crowd. Since there are multiple workers annotating each object, each object can suffer from multiple types of error: we found that out of the 46 objects in our dataset, 9 objects suffered from type one error and 18 objects from type two error. Almost all objects suffer from some form of type three error of varying degrees of imprecision around the object boundary. The main evaluation methods highlighted in Section 4 focuses on resolving the imprecise, "sloppy" bounding box errors. In Section 5, we discuss a preprocessing method eliminates semantic ambiguities and errors.

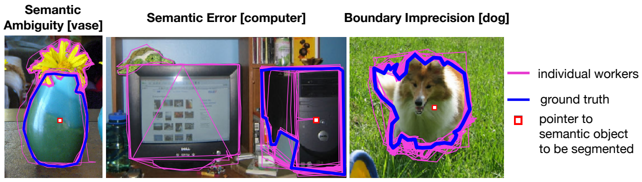
Semantic Ambiguity [vase]
Semantic Error [computer]
Boundary Imprecision [dog]
individual workers
ground truth
pointer to semantic object to be segmented

Figure 2: Examples of common worker mistakes.

4 FIXING BOUNDARY IMPERFECTIONS

4.1 Tile Data Model

At the heart of our aggregation techniques is a *tile data representation*. A tile is the smallest non-overlapping discrete unit created by overlaying all of the workers' segmentations on top of each other. The tile representation allows us to aggregate segmentations from multiple workers, rather than being restricted to a single worker's segmentation. This allows us to fix one worker's errors with help from another worker's segmentation. In Figure 3 (right), we display three worker segmentations for a toy example. Worker 1's segmentation is represented in pink, worker 2's segmentation in yellow, and worker 3's segmentation in blue. These segmentations overlap resulting in a partitioning or tiling of the image with 6 distinct resulting tiles. For instance, tile t_1 is the portion of worker 1's segmentation that is not contained in worker 2 or worker 3's segmentations, tile t_2 is the intersection of all three workers' segmentations, and t_3 is the intersection of worker 2 and worker 3's segmentations excluding worker 1's segmentation. Any subset of these 6 tiles can contribute towards the final segmentation.

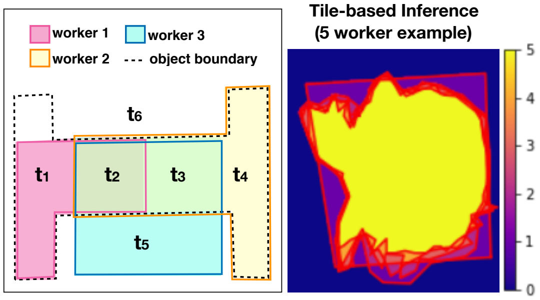

Figure 3: Left: Toy example demonstrating tiles created by three workers' segmentations around a dumbell object delineated by the black dotted line. Right: Segmentation boundaries drawn by five workers shown in red. Overlaid segmentation creates a mask where the color indicates the number of workers who voted for the tile region.

The simple but powerful idea of tiles also allows us to reformulate our problem from one of "generating a segmentation" to a setting that is much more familiar to crowdsourcing researchers. Since tiles are the lowest granularity units created by overlaying all workers' segmentations on top of each other, each tile is either completely contained within or outside a given worker segmentation. Specifically, we can regard a worker segmentation as multiple boolean responses where they have voted 'yes' or 'no' to every tile independently. Intuitively, a worker votes 'yes' for every tile that is contained in their segmentation, and 'no' for every tile that is not. As shown in Figure 3 (right), tile t_2 is voted 'yes' by worker 1, 2, and 3; tile t_3 is voted 'yes' by worker 2 and 3. The goal of our aggregation algorithms is to pick an appropriate set of tiles that effectively trades off precision versus recall. This is equivalent to making a boolean decision of "include tile in output", or "yes" versus "don't include tile in output", or "no" for each tile, given multiple boolean worker votes for each tile. Thus we have projected the original problem of generating a segmentation onto a boolean aggregation problem. We focus on algorithms for the boolean decision problem setting for the remainder of this section.

4.2 Majority Vote Aggregation (MV)

Majority Vote aggregation is a standard technique for boolean aggregation in crowdsourcing. We treat each tile as an independent boolean decision and assign equal weight to each worker's votes, thereby implictly assuming that all workers are equal. We include a tile in the output segmentation if and only at least 50% of all workers have voted "yes" for the tile. In practice, however, not all workers are equal—some workers tend to make more mistakes than others, or have particular biases. Furthermore, not all tile decisions are necessarily independent because our aggregate decisions on some tiles could affect our belief of worker qualities, which in turn could influence our aggregation decisions on other tiles. Next, we extend our approach to capture and incorporate worker qualities into our algorithms.

4.3 Worker Quality-Aware Algorithms

We intuitively describe three worker models that we experiment with below. In our technical report, we formalize the notion of the probability that a set of tiles forms the ground truth, and solve the corresponding maximum likelihood problem, for each of these worker models.

Worker quality models.
Let us first define some useful notation.

Let $\mathcal{T} = \{t_k\}$ be the set of all non-overlapping tiles for an object i. T is the ground truth tile set. T' is some combination of tiles chosen from $\mathcal{T}$. The indicator label l_{kj} is one when worker j votes on the tile t_k (i.e. the bounding box that he draws contains t_k), and zero otherwise. The indicator matrix consisting of tile indicator for all workers is denoted as $\mathbf{l_{kj}}$.

We propose three different worker error models describing the probability of a worker j's vote on a specific tile t_k, given the tile's inclusion in ground truth and a set of worker qualities Q_j.

(1) Basic: single-parameter Bernoulli model, where q_j is the probability of the worker getting a tile correct. A worker is correct when his vote (l_{jk}) matches with the ground truth inclusion of the tile ($t_k \in T$). A worker makes an incorrect response when their vote contradicts with the inclusion of the tile in T ($\{t_k \in \text{T} \quad \& \quad l_{kj} = 0\}, \{t_k \notin \text{T} \quad \& \quad l_{kj} = 1\}$)

$$p(l_{jk}|t_k \in T, Qj) = \begin{cases} q_j, & l_{jk} = 1 \\ 1 - q_j, & l_{jk} = 0 \end{cases} \tag{1}$$

(2) Large Small Area (LSA): The basic model equally weighs all tiles, but intuitively a worker should be rewarded more if they get a large-area tile correct. We use a two-parameter Bernoulli to model two different tile sizes determined by a threshold A^*.

$$p(l_{jk}|t_k \in T, Q_j) = \begin{cases} q_{j1}, & l_{jk} = 1 \& A(t_k) \geq A^* \\ 1 - q_{j1}, & l_{jk} = 0 \& A(t_k) \geq A^* \\ q_{j2}, & l_{jk} = 1 \& A(t_k) < A^* \\ 1 - q_{j2}, & l_{jk} = 0 \& A(t_k) < A^* \end{cases} \tag{2}$$

(3) Ground truth inclusion, large small area (GTLSA): We observe in our experiment that there can be many large area tiles that lies outside of the ground truth drawn by workers who tend to draw loose, overbounding boxes. Our 4 parameter Bernoulli model distinguishes between false and true positive rates, by taking into account the positive and negative regions (i.e. regions that lies inside or outside of T). In the case where $A(t_k) \geq A^*$:

$$p(l_{jk}|t_k \in T, Q_j) = \begin{cases} q_{p1}, & l_{jk} = 1 \\ 1 - q_{p1}, & l_{jk} = 0 \end{cases} \tag{3}$$

$$p(l_{jk}|t_k \notin T, Q_j) = \begin{cases} q_{n1}, & l_{jk} = 0 \\ 1 - q_{n1}, & l_{jk} = 1 \end{cases} \tag{4}$$

From the worker error model, we can also derive the probability that a tile is in ground truth $p(t_k \in T|Q_j, l_{jk})$ using Bayes rule, assuming the prior probabilities as constant.

We can think of workers as agents that look at each pixel in an image and label it as part of the segmentation, or not. Their actual segmentation is the union of all the pixels that they labeled as being part of their segmentations. Each pixel in the image is also either included in the ground truth segmentation or not included in the ground truth segmentation. We can now model worker segmentation as a set of boolean pixel-level (include or don't include) tasks, each having a ground truth boolean value. Based on this idea, we explore three worker quality models:

- *Basic model:* Each worker is captured by a single parameter Bernoulli model, $< q >$, which represents the probability that a worker will label an arbitrary pixel correctly.
- *Ground truth inclusion model (GT):* Two parameter Bernoulli model $< qp, qn >$, capturing false positive and false negative rates of a worker. This helps to separate between workers that tend to overbound and workers that tend to underbound segmentations.
- *Ground truth inclusion, large small area model (GTLSA):* Four parameter model $< qp_l, qn_l, qp_s, qn_s >$, that distinguishes between false positive and false negative rates for large and small tiles. In addition to capturing overbounding and underbounding tendencies, this model captures the fact that workers tend to make more mistakes on small tiles, and penalizes mistakes on large tiles more heavily.

For our problem, we consider only finding tile regions that could be constructed from worker bounding boxes. In other words, our objective is to find the tile combination T' that maximizes the probability that it is the ground truth p(T'=T), given a set of worker qualities Q_j and tile indicator labels l_{jk}:

$$T = \arg\max_{T' \subseteq \mathcal{T}} p(T = T'|\mathbf{l_{kj}}, Q_j) \tag{5}$$

Using Bayes rule we can rewrite this in terms of the posterior probability of the tile-based values($\mathbf{l_{kj}}$) or worker-based values(Q_j), which we can use for the E and M step equations respectively.

4.4 Inference

For the E step, we assume T' is ground truth and estimate the Q_j parameters. We can rewrite Eq.5 as:

$$p(T'|Q_j, \mathbf{l_{kj}}) \approx p(l_{kj}|Q, T') \tag{6}$$

where we treat the priors $p(T'), p(Q_j)$ as constants. Our goal is to find the maximum likelihood parameters of Q_j:

$$\hat{Q}j = \arg\max_{Q_j} p(Q_j|\mathbf{l_{kj}}, T') \tag{7}$$

We use the binary random variable w to indicate whether the worker makes a correct vote (w=1) or an incorrect vote(w=0) for a tile. We can write the worker quality probability as the product of the probabilities that they would assume these two independent states (correct/incorrect).

$$p(Q_j) = \prod_j q_j^{p_j(w=1)} \cdot [1 - q_j]^{p(w=0)} \tag{8}$$

The closed form of the maximum likelihood solution for the Bernoulli distribution reduces down to:

$$\hat{q_j} = \frac{n_{correct}}{n_{total}} \tag{9}$$

For the M step, we maximize the likelihood of the tile combination T' for a fixed set of worker qualities, $\{Q_j\}$. Following Eq.5 from Bayes rule,

$$p(T'|Q_j,\mathbf{l_{kj}}) \approx p(\mathbf{l_{kj}}|Q_j,l_k) \tag{10}$$

Our optimization function is written as:

$$\hat{T}' = \underset{T' \supseteq \{T'\}}{\arg\max} \prod_j p(\mathbf{l_{kj}}|Q_j,l_k) \tag{11}$$

The product over T' can be further decomposed into its tile components. The likelihoods of these tiles can be computed via the worker error model:

$$= \underset{T' \supseteq \{T'\}}{\arg\max} \prod_j \left[\prod_{t_k \in T'} p(t_k \in \mathrm{T}|Q_j,l_k) \prod_{t_k \notin T'} p(t_k \notin \mathrm{T}|Q_j,l_k) \right] \tag{12}$$

Since the space of possible $\{T'\}$ to search through is 2^N where number of tiles (N) for an average object with 30~40 worker is on the order of thousands, we develop several strategies to narrow the search space for making the problem computationally feasible.

Expectation-Maximization (EM). Unlike MV, which assumes that all workers perform uniformly, EM approaches use worker quality models to infer the likelihood that a tile is part of the ground truth segmentation. While simultaneously estimating worker qualities and tile likelihoods as hidden variables, our basic worker quality model that we evaluate in Section 7 assumes a fixed probability for a correct vote. Details of the formal derivation and other more fine-grained worker quality models can be found in our technical report.

Apart from constructing a set of $\{T'\}$ for picking the best T', we can instead directly construct the maximum likelihood tile T^* by choosing tiles that satisfy the criterion:

$$T^* = \{t_k | p(t_k \in T|l_k,Q_j) \geq p(t_k \notin T|l_k,Q_j)\} \tag{13}$$

Proof: We show that this tile-picking heuristic is at least as likely as any tile combination that we would pick with the $\{T'\}$ selection method. Suppose there is a T' such that it consists of the same tiles as T^*, but we randomly drop a tile $t_{k'}$

$$p(T^* = T'|l_k,Q_j) = \prod_{t_k} p(t_k \in T^*) \cdot p(t_{k'} \notin T^*) \tag{14}$$

By definition all tiles in T^* must satisfy $p(t_k \in T|l_k,Q_j) \geq p(t_k \notin T|l_k,Q_j)$, so the dropped tile must have lower probability than T'.

$$p(T = T') = p(T^* \setminus t_k')p(t_k' \notin T^*) \tag{15}$$

$$p(T = T^*) = p(T^* \setminus t_k')p(t_k' \in T^*) \tag{16}$$

By dropping multiple $t_{k'}$ from T^* or adding $t_{k'}$ not previously in T^*, the above result can be generalized to arbitrary T'.

Greedy Tile Picking (greedy). In the previous algorithms, we have tried to capture the probability of a tile being *completely* contained in the ground truth and selected the union of high likelihood tiles as our final output segmentation. In reality, tiles near the boundary of the object being segmented often have a partial overlap with the ground truth. Since our algorithms either include or exclude entire tiles from the final output, it is not possible for the boundary of our segmentation to be perfect. The greedy algorithm we describe in this section attempts to alleviate this problem by heuristically picking tiles based on their cost vs benefit trade-off. This algorithm aims to effectively trade off precision (decreased by including tiles and increased by excluding tiles) vs recall (increased by including tiles and decreased by excluding tiles) to optimize for the jaccard similarity of our output segmentation as compared to the underlying ground truth.

The greedy algorithm picks tiles in descending order based on the ratios of overlap area to non-overlap area (both with respect to ground truth), for as long as the estimated Jaccard similarity of the resulting segmentation continue to increase. Since the tile overlap and non-overlap against ground truth are unknown, we use tile-inclusion probabilities from EM to estimate these areas as a heuristic. Furthermore, since we cannot compute the actual Jaccard similarity against the unknown ground truth, we use a heuristic baseline such as MV as a proxy for the ground truth. Intuitively, tiles that have a high overlap area and low non-overlap area contribute to high recall, at the cost of relatively little precision error. We include a proof in our technical report showing that picking tiles in such an order maximizes the Jaccard similarity of the resulting segmentation locally at every step. While we have focused on optimizing for jaccard score here, the greedy algorithm is flexible and can be easily adapted for any objective metric that we might wish to optimize.

```
Data: fixed $Q_j$
Initialize $T^*$;
for $t_k \in \mathcal{T}$ do
    if $p(t_k \in T) \geq p(t_k \notin T)$ then
        $T^* \leftarrow T^* \cup t_k$;
    end
end
```

Algorithm 1: M step algorithm. For the initialization of T^*, we could start from either an empty set or a high-confidence tileset. The set of $\mathcal{T}$ to chose from can either be the set of all tiles or all tiles adjacent to T^*.

5 PERSPECTIVE RESOLUTION

As discussed in Section 3, disagreements often arise in segmentation due to differing worker perspectives on large tile regions. We developed a clustering-based preprocessing approach to resolve this issue. In order to group together segmentations with similar perspectives, we compute a NxN distance matrix where N is the number of workers based on the Jaccard similarity between each pair of segmentations. Our intuition is that workers with similar perspectives will have segmentations that are close to each other. Using the distance matrix, we then perform spectral clustering to separate the segmentations into clusters.

Figure 4 illustrates how spectral clustering divides the worker segmentations into clusters with meaningful semantic associations, reflecting the diversity of perspectives for the same task. Clustering results can also be used as a preprocessing step for any quality evaluation algorithm by keeping only the segmentations that belong to the largest cluster, which is typically free of semantic errors.

In addition, clustering offers the additional benefit of preserving worker's semantic intentions. For example, while the green cluster in Figure 2 (bottom right) would be considered *bad* segmentations for the particular task ('computer'), this cluster can provide more data for another segmentation task corresponding to 'monitor'. A

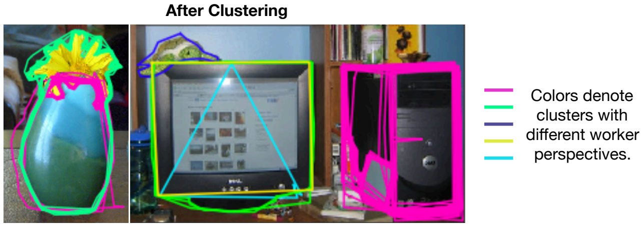

Figure 4: Example image showing clustering performed on the same object from Figure 2 left and middle.

potential future work direction would be to crowdsource the semantic labels for the computed clusters to enable the reuse of segmentations across multiple objects to lower costs.

6 EXPERIMENTAL SETUP

6.1 Dataset Description

We collected crowdsourced segmentations from Amazon Mechanical Turk; each HIT consisted of one segmentation task for a specific pre-labeled object in an image. There were a total of 46 objects in 9 images from the MSCOCO dataset [12] segmented by 40 different workers each, resulting in a total of 1840 segmentations. Each task contained a keyword for the object and a pointer indicating the object to be segmented. Two of the authors generated the ground truth segmentations by carefully segmenting the objects using the same task and interface.

A sub-sampled dataset was created from the full dataset to determine the efficacy of these algorithms on varying number of worker responses. Every object was randomly sampled worker with replacement. For small worker samples, we average our results over larger number of batches than for large worker samples (which have lower variance, since the sample size is close to the original data size).

6.2 Evaluation Metrics

Evaluation metrics used in our experiments measure how well the final segmentation (S) produced by these algorithms compare against ground truth (GT). The most common evaluation metrics used in the literature[4, 12, 15, 16] are area-based methods that take into account the intersection area, $IA = area(S \cap GT)$, or union area, $UA = area(S \cup GT)$ between the worker and ground truth segmentations, including Precision (P) $= \frac{IA(S)}{area(S)}$, Recall (R) $= \frac{IA(S)}{area(GT)}$, and Jaccard (J) $= \frac{UA(S)}{IA(S)}$.

6.3 Baseline Algorithms

Retrieval-based Methods

Number of Control Points (num pts): This algorithm picks the worker segmentation with the largest number of control points around the segmentation boundary (i.e., the most precise drawing) as the output segmentation [17, 19]. Intuitively, workers that have used a larger number of points are likely to have been more precise, and provided a more complex and accurate segmentation.
Average worker: This baseline computes the average Jaccard across all workers, which simulates collecting only a single worker annotation.

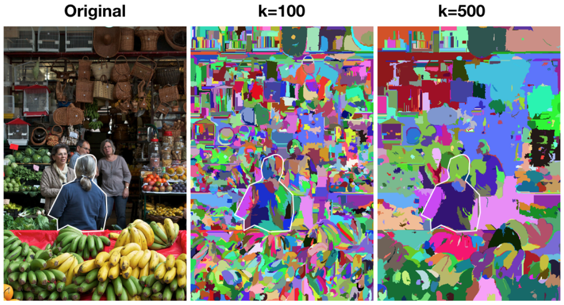

Figure 5: Example of the vision color tiling for different chosen granularities. Left: Raw image. Vision segmentation with $k = 100$(Center) and $k = 500$ (Right). Vision tiles with a significant overlap area with the worker segmentation (white boundaries) is selected.

Best worker: Selecting the best worker based on Jaccard against ground truth.

Vision-based Methods

We implement a semi-supervised algorithm that can produce segmentations for arbitrary objects in the absence of large volumes of tailor-made training data. While this algorithm works largely on raw image data, it requires some external help in the form of one "reference" segmentation. Intuitively, a rough segmentation can be thought of as a pointer for the algorithm to the relevant regions of the image. The algorithm then uses the color profile of the image to segment out the similarly colored regions of the image that overlap with the reference segmentation. Specifically, we begin by splitting the input image into multiple regions, or *tiles* that have the same color using the work of [7]—the desired number of output tiles can be modified using a tuning parameter k, to produce finer or coarser tiles.

We used the popular open source segmentation algorithm developed by Felzenszwalb and Huttenlocher [7]. We fixed the smoothing and minimum component size parameters and varied the threshold determining the how refined the segmentation is. As shown in Figure 5, larger values for k result in larger components in the result. We overlay the given rough segmentation on top of the color tiles.
Average vision:
Best vision: Now, the algorithm focuses on *choosing the right set of tiles based on the given reference segmentation*. Intuitively the algorithm picks color tiles that have significant overlap with the given reference segmentation, i.e., returns the union of all tiles for which greater than a certain area threshold of the tile is intersecting with the reference segmentation. We experiment with different granularities for the vision preprocessing as well as scan a variety of tile filtering area thresholds.

7 EXPERIMENTAL RESULTS

Aggregation-based methods perform significantly better than retrieval-based methods.

In Figure 6, we vary the number of worker segmentations along the x-axis and plot the average Jaccard score on the y-axis across different worker samples of a given size across different algorithms. Figure 6 (left) shows that the performance of aggregation-based algorithms

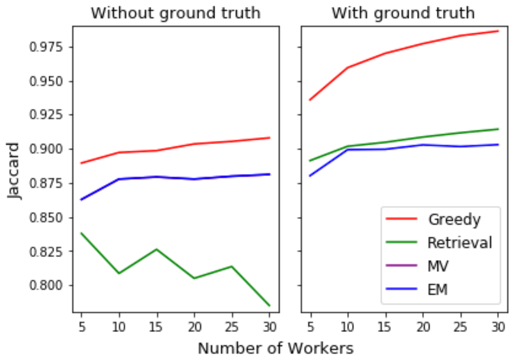

Figure 6: Performance of the original algorithms that do not make use of ground truth information (Left) and ones that do (Right). MV and EM results are so close that they overlay on each other.

(greedy, EM) exceeds the best-achievable through existing retrieval-based method (Retrieval). Then, in Figure 6 (right), we estimate the upper-bound performance of each algorithm by assuming that the 'full information' based on ground truth was given to the algorithm. For greedy, the algorithm is aware of all the actual tile overlap and non-overlap areas against ground truth, and does not need to approximate these values. For EM, we consider the performance of the algorithm if the true worker quality parameter values (under our worker quality model) are known. For retrieval, the full information version directly picks the worker with the highest Jaccard similarity with respect to the ground truth segmentation. By making use of ground truth information (Figure 6 right), the best aggregation-based algorithm can achieve a close-to-perfect average Jaccard score of 0.98 as an upper bound, far exceeding the results achievable by any single 'best' worker (J=0.91). This result demonstrates that aggregation-based methods are able to achieve better performance by performing inference at the tile granularity, which is guaranteed to be finer grained than any individual worker segmentation.

The performance of aggregation-based methods scale well as more worker segmentations are added.

Intuitively, larger numbers of worker segmentations result in finer granularity tiles for the aggregation-based methods. The first row in Table 1 lists the average percentage change in Jaccard between 5-workers and 30-workers samples, demonstrating a monotonically increasing relationship between number of worker segmentations used and the performance. However, retrieval-based methods do not benefit from more segmentations.

	Retrieval-based		Aggregation-based			
Algorithm	num pts	worker*	MV	EM	greedy	greedy*
Worker Scaling	-6.30	2.58	2.12	1.78	2.07	5.38
Clustering Effect	5.92	-0.02	2.05	0.03	5.73	0.283

Table 1: Jaccard percentage change due to worker scaling and clustering. Algorithms with * makes use of ground truth information.

Clustering as preprocessing improves algorithmic performance.

The average percentage change between the no clustering and clustering results is shown in Table 1. Clustering generally results in an accuracy increase. Since the 'full information' variants are already free of semantic ambiguity and errors, clustering does not assist with further improvement.

The clustering preprocessing step can significantly improve performance of algorithms that are not very robust to segmentations with semantic errors or ambiguities, such as the heuristic-based number of points approach. When examining the gap of increase with and without clustering in Figure 7, we find that aggregation-based methods performs better than retrieval-methods exhibits a smaller gap between the performances. This effect is due to aggregation-based method's higher performance in the no cluster case, indicating that it is able to capture some of the semantic ambiguities and errors in the dataset.

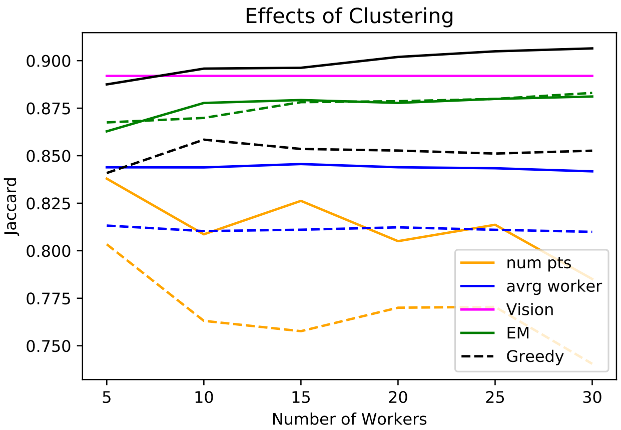

Figure 7: Performance comparisons between averaging over experiments with clustering as a preprocessing step (dotted) and the unclustered results (solid) for different algorithms.

How well does the inferred worker qualities predict individual worker performance?

Correlation of worker qualities against performance To further investigate how the EM models are performing, we looked at whether the model-inferred worker qualities is indicative of the actual quality of a segmentation. We performed linear fitting independently for each sample-objects and computed the R^2 statistics to determine whether worker qualities can accurately predict precision, recall, and Jaccard scores. Visual inspection of the basic worker quality model fitting showed that for objects that suffered from type two errors (semantic ambiguity), the single-parameter worker quality was unable to capture the overbounding behavior, which lead to a low precision and Jaccard. The results are listed in Table 2 to highlight how our advanced worker qualities were able to better capture these scenarios. The clustering preprocessing was not performed for the values in Table 2 to demonstrate the sole effect of the EM algorithm. Nevertheless, our clustered results also show a similar trend, with an average of R^2=0.88 and 0.89 for the GT and GTLSA models across all objects respectively. We also find that in general the linear fit improves as the number of data points increases, which indicates consistency in the fitted model.

Best worker quality retrieval One application of worker qualities is that it could be used as an annotation scoring function for retrieving

N	basic	GT	GTLSA	isobasic	isoGT	isoGTLSA
5	0.601	0.907	0.901	0.576	0.907	0.904
10	0.632	0.895	0.899	0.633	0.895	0.898
15	0.622	0.897	0.898	0.622	0.897	0.897
20	0.636	0.894	0.899	0.637	0.894	0.898
25	0.66	0.901	0.905	0.661	0.901	0.904
30	0.673	0.907	0.914	0.676	0.907	0.913

Table 2: Linear correlation of worker qualities against ground truth performance for different quality models across different number of workers (N). The lower worker samples exhibit lower R^2 due to the variance from smaller number of datapoints for each independent fit.

the best quality worker segmentation. We explore this approach by training a linear regression model for every sample-object and use the worker qualities to predict the precision, recall, and Jaccard of individual worker annotations against ground truth. Then, we query the model with the inferred worker quality and retrieve the worker with the best predicted Jaccard.

The reason why a linear regression model was chosen rather than simply sorting the worker qualities and picking the best is that sorting based on multiple worker qualities (precision, recall, Jaccard) effectively applies equal weighting to all quality attributes, whereas our advanced models are specifically designed to capture cases of false-positives and false-negatives that can yield drastically different recall and precision values. We have tested that the linear regression model performs better on this task that simple sorting is capable of learning the weights that helps it make better predictions. As shown in Table 3, the performance of worker-quality based retrieval is comparable the performance other aggregation-based methods. We find that amongst the different worker quality models, advanced worker quality models perform the best, agreeing with our intuition regarding correlation results observed in Table 2.

algo/N	5	10	15	20	25	30
num points	0.838	0.809	0.826	0.805	0.814	0.785
best worker	0.891	0.902	0.905	0.909	0.912	0.914
MV	0.885	0.893	0.894	0.897	0.898	0.899
EM[basic]	0.884	0.893	0.894	0.897	0.898	0.899
EM[GT]	0.885	0.893	0.894	0.897	0.898	0.899
EM[GTLSA]	0.871	0.892	0.891	0.896	0.897	0.899
greedy	0.888	0.896	0.896	0.902	0.905	0.906
wqr[basic]	0.878	0.877	0.877	0.877	0.878	0.878
wqr[GT]	0.884	0.885	0.885	0.885	0.887	0.887
wqr[GTLSA]	0.874	0.881	0.883	0.885	0.886	0.887

Table 3: Summary of average performance across workers with clustering applied as preprocessing in all algorithms across different number of workers (N). wqr is the abbreviation for best worker quality retrieval methods.

8 CONCLUSION AND FUTURE WORK

We identified three different types of errors for crowdsourced image segmentation, developed a clustering-based method to capture the semantic diversity caused by differing worker perspectives, and introduced novel aggregation-based methods that produce more accurate segmentations than existing retrieval-based methods.

Our paper show that our worker quality models are good indicators of the actual accuracy of worker segmentations. We also observe that the greedy algorithm is capable of achieving close-to-perfect segmentation accuracy with ground truth information. Given the success of aggregation-based methods, including the simple majority vote algorithm, future work includes using our worker quality insights to improve our EM and greedy algorithms and using computer vision signals to further improve our algorithms.

REFERENCES

[1] Adriana Kovashka, Olga Russakovsky, and Li Fei-Fei. 2016. Crowdsourcing in Computer Vision. *Foundations and Trends® in Computer Graphics and Vision* 10, 2 (2016), 103–175. https://doi.org/10.1561/0600000073

[2] Sean Bell, Kavita Bala, and Noah Snavely. 2014. Intrinsic Images in the Wild. *ACM Trans. on Graphics (SIGGRAPH)* 33, 4 (2014).

[3] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. 2015. Material Recognition in the Wild with the Materials in Context Database. *Computer Vision and Pattern Recognition (CVPR)* (2015).

[4] Ferran Cabezas, Axel Carlier, Vincent Charvillat, Amaia Salvador, and Xavier Giro-I-Nieto. 2015. Quality control in crowdsourced object segmentation. *Proceedings of International Conference on Image Processing, ICIP* 2015-Decem (2015), 4243–4247. https://doi.org/10.1109/ICIP.2015.7351606 arXiv:arXiv:1505.00145v1

[5] A. P. Dawid and A. M. Skene. 1979. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. 28, 1 (1979), 20–28.

[6] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. 2015. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision* 111, 1 (Jan. 2015), 98–136.

[7] Pedro F Felzenszwalb and Daniel P Huttenlocher. 2004. Efficient graph-based image segmentation. *International journal of computer vision* 59, 2 (2004), 167–181.

[8] Danna Gurari, Kun He, Bo Xiong, Jianming Zhang, Mehrnoosh Sameki, Suyog Dutt Jain, Stan Sclaroff, Margrit Betke, and Kristen Grauman. 2018. Predicting Foreground Object Ambiguity and Efficiently Crowdsourcing the Segmentation(s). *International Journal of Computer Vision (IJCV)* (2018). https://doi.org/10.1007/s11263-018-1065-7 arXiv:1705.00366

[9] Danna Gurari, Mehrnoosh Sameki, Zheng Wu, and Margrit Betke. 2016. Mixing Crowd and Algorithm Efforts to Segment Objects in Biomedical Images. *Medical Image Computing and Computer Assisted Intervention Interactive Medical Image Computation Workshop* (2016), 1–8.

[10] H Irshad and Montaser-Kouhsari et. al. 2014. Crowdsourcing Image Annotation for Nucleus Detection and Segmentation in Computational Pathology: Evaluating Experts, Automated Methods, and the Crowd. *Biocomputing 2015* (2014), 294–305. https://doi.org/10.1142/9789814644730_0029

[11] Christopher H Lin, Mausam, and Daniel S Weld. 2012. Crowdsourcing control : Moving beyond multiple choice. *AAAI Conference on Human Computation and Crowdsourcing (HCOMP)* (2012), 491–500. arXiv:arXiv preprint arXiv:1210.4870.

[12] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. *European Conference on Computer Vision (ECCV)* 8693 LNCS, PART 5 (2014), 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

[13] E. Natonek. 1998. Fast range image segmentation for servicing robots. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, Vol. 1. 406–411 vol.1. https://doi.org/10.1109/ROBOT.1998.676445

[14] Olga Russakovsky, Li-Jia Li, and Li Fei-Fei. 2015. Best of Both Worlds: Human-Machine Collaboration for Object Annotation. (2015), 2121–2131.

[15] Mehrnoosh Sameki, Danna Gurari, and Margrit Betke. 2015. Characterizing Image Segmentation Behavior of the Crowd. (2015), 1–4.

[16] Jean Y. Song, Raymond Fok, Alan Lundgard, Fang Yang, Juho Kim, and Walter S. Lasecki. 2018. Two Tools are Better Than One : Tool Diversity as a Means of Improving Aggregate Crowd Performance. *Proceedings of the International Conference on Intelligent User Interfaces* (2018).

[17] Alexander Sorokin and David Forsyth. 2008. Utility data annotaton with Amazon Mechanical Turk. *Proceedings of the 1st IEEE Workshop on Internet Vision at CVPR 08* c (2008), 1–8. https://doi.org/10.1109/CVPRW.2008.4562953

[18] Antonio Torralba, Bryan C. Russell, and Jenny Yuen. 2010. LabelMe: Online image annotation and applications. *Proc. IEEE* 98, 8 (2010), 1467–1484. https://doi.org/10.1109/JPROC.2010.2050290

[19] Sirion Vittayakorn and James Hays. 2011. Quality Assessment for Crowdsourced Object Annotations. *Procedings of the British Machine Vision Conference* (2011), 109.1–109.11. https://doi.org/10.5244/C.25.109

[20] Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. 2010. The Multidimensional Wisdom of Crowds. *NIPS (Conference on Neural Information Processing Systems)* 6 (2010), 1–9. https://doi.org/10.1.1.231.1538

[21] Kota Yamaguchi. 2012. Parsing Clothing in Fashion Photographs. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (CVPR '12)*. IEEE Computer Society, Washington, DC, USA, 3570–3577. http://dl.acm.org/citation.cfm?id=2354409.2355126

[22] Y.Y.Boykov and M-P.Jolly. 2001. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in \textup{N}-\textup{D} Images. *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on* July (2001), 105–112.

[23] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. Scene Parsing through ADE20K Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.