Tile-Based Quality Evaluation Methods for Crowdsourcing Image Segmentation

Authors removed for Anonymity

Abstract

Abstract

Introduction

The goal of visual scene understanding enable computers to achieve high-level comprehension from images or videos. While object localization and detection identifies *where* an object is in the image, object segmentation provides rich information regarding *what* the objects look like. Precise, instance-level segmentation is crucial for automation system with visual inputs such as robotics() and autonomous vehicles (CITE), satellite imagery monitoring (CITE?) and biomedical image segmentation (CITE). Vision-based object segmentation algorithms often suffer from oversegmented regions and performs poorly for occluded objects(**?**), clutterred images(**?**), undesirable lighting conditions(**?**). For the purpose of semantic segmentation, CV outputs often needs to be combined with crowdsourced image labels or object segmentation (**?**; **?**; **?**) to address to question of which portion of the image corresponds to the object. Crowdsourcing is —– (**?**).

However, crowdsourcing —— — noisy worker responses. - noisy - CITE previous literature show three types of error : worker response can identify the wrong object (ambiguous), low-precision BBs (sloppy), underdraw/overdraw.

- basic characterization of our dataset to show that the worker responses are indeed noisy

- stats: % overdraw and underdraw - variance in ground truth P,R

Related Works

NOTE: Add related works on CV methods for object segmentation, what their problem is and why we need crowdsourcing responses (CITE GraphCut, CRFs)

A detailed survey on how crowdsourcing is used for CV can be found in (**?**).

Despite several large-scale efforts to collect crowdsourced image segmentation (**?**; **?**; **?**; **?**), most have relied on summarization-based metrics to quantify their segmentation data quality. Summarization-based scores are functions that measures the quality of a worker's bounding box given the ground truth. Common summarization methods include precision, recall, area ratio or number of control points in the bounding box (**?**). While these are fairly discriminative measures of bounding box, the scoring calculation requires the use of ground truth to compare against. Other heuristic approaches that doesn't require ground truth segmentation include quantifying the user types and their click-stream behavior to determine work quality(**?**; **?**) and comparing the worker responses to features extracted from computer vision algorithms(**?**; **?**).

Dawid and Skene(**?**) is the first work that used the EM algorithm to model an individual's biases and skills in the absence of ground truth data, by using a confusion matrix. Welinder and Perona (**?**) proposes a general model that separately models annotator quality and the biases applied to binary, multivalued and continuous annotations. Welinder and Perona (**?**) develops a multidimensional concept of worker qualities and task difficulties by considering object-presence labeling as a noise generation process. The objective truth label is captured by a multidimensional quantity of task-specific measurements and deformed by worker and image related noise, the noisy vector obtained after this process is projected onto the vector of user expertise (which summarizes how well the user perceives each of these measurements), and finally the score is binarized into an inferred label. Many have extended this line of work beyond binary classifications by developing EM-like approaches that works on multiple-choice (**?**) as well as free-form responses (**?**), but these have not been directly applied to the task of object segmentation.

However, while EM algorithms assign probabilities regarding *how good a worker's bounding box is*, for the task of object segmentation, we are ultimately more interested the end goal of *what is the best bounding box that we can get from these data*. Since these formal probabilistic models treats worker bounding box as the base quantity for modeling worker quality, the best bounding box that one could derive from such algorithm can only be as good as the best worker bounding box in the dataset. Even though the annotation probabilities are sufficient for determining the best binary-labels, image information such as overlapping areas would be useful and not account for in these algorithms. We

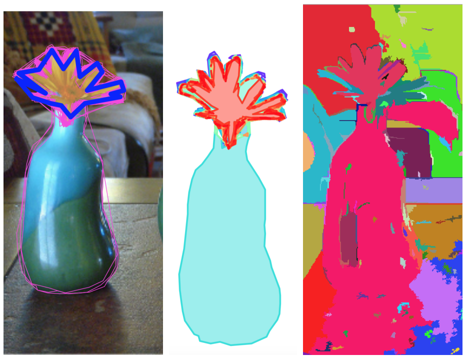

Figure 1: Left: Pink shows 40 worker bounding boxes for the object “flower“ and blue ground truth segmentation, superimposed on the original COCO image. Center: Non-overlapping tiles constructed from worker bounding box. Right: Output from the color-segmentation vision algorithm.

suspect that this is why many area-based metrics are still more commonly used in practice than EM approaches.

Model

Definitions

The basic unit used for conducting inference on object boundary are tiles. We first transform the set of worker bounding boxes into tiles, as shown in Fig. 1. Our goal is to find the best tile combination that makes up the object in the image. Since the basic unit for conducting inference is tiles constructed by worker bounding boxes, the search space of our solutions does not contain any region that have not been proposed by any workers, however, our approach is still more fine-grained than existing models that conducts inference on individual worker bounding boxes. Tiles are finer grained than bounding boxes. Our tile-based approach is inspired by the S-T graph in the classical graph cut problem, where the goal of image segmentation is to find a vertex partition between the object and background regions.

$\mathcal{T} = \{t_k\}$ is 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 $\mathbf{l_{kj}}$.

Worker Error Model

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

Problem Statement

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}:

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

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.

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:

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

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:

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

Optimization

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.

High-confidence snowball The goal of the snowball method is to come up with smaller subsample of tile combinations T' that are good candidates of ground truth. First, we use tile properties such as area or votes as a heuristic to derive a fixed set of high-confidence tiles as the core. Then, using the same heuristic, we randomly generate subsets from other medium confidence tiles and combined with these core tiles. Tiles picked with such heuristics often have high recall, which means that our TileEM algorithm essentially helps us find a more precise T' from $\{T'\}$. In our experiment, we define our confidence score as 2·votes+area, with 3 high-confidence, fixed core tiles and 40 flexible medium confidence tiles.

NOTE: Might want to consider adjacency-based snowball approach too

Maximum likelihood Construction 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'.

```
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^*.

```
Data: fixed $Q_j$
$T^*$ = high confidence tiles;
$d'$=0;
good tile count at $d'$=1;
while good tile count at $d' \neq 0$ do
    $\{t_{k,d=d'}\} \leftarrow$ find all tiles at d=$d'$ shell;
    for $t_k \in \{t_{k,d=d'}\}$ do
        if $p(t_k \in T) \geq p(t_k \notin T)$ then
            $T^* \leftarrow T^* \cup t_k$;
            good tile count at $d'$ ++;
        end
    end
    $d'$++;
end
```

Algorithm 2: Shell-based M step algorithm enforces tiles that are added into T^* must be adjacent to one another.

Experiments

We collected data from Amazon Mechanical Turk where each HIT consisted of one annotation task for a specific prelabelled object in the image. There is a total of 44 objects in 9 images from the MSCOCO dataset (**?**). These tasks represent a variety of image difficulty (clutterness, occlusion, lighting)and levels of task ambiguity. For each object, we collected annotations from a total of 40 independent workers. We eliminated all bounding boxes from workers that were self-intersecting. A sub-sampled dataset was created from the full dataset to determine the efficacy of these algorithms on varying number of worker responses, based on Table.**??**.

Baselines

Summarization-based metrics are computable heuristics that measure the quality of a worker's bounding box given the ground truth. We employ two metric used in (**?**), number of control points and annotation size, which has been shown to perform better than vision-based summarization metrics. The number of control points metric is based on the intuition that a more precise bounding box consisting of a large number of control points usually indicates that the user made an effort to closely follow the object boundary. The annotation size is based on the intuition that larger bounding-box-to-image-area ratio means that the object is easier to annotate, hence its quality should be higher. Both of these metrics optimizes for recall at the cost of precision loss. Summarization based metrics does poorly in cases where the 1-D projection of the BBs fails to capture the worker errors fully. They are good indicators assuming the worker error is only based on the degree of sloppiness of his bounding box rather than mistakes on which regions should be incorporated in the object. For the same reasons, these metric also fail in the case of overbounding or underbounding BBs.

Optional: show example of where this fails from vision GT output

Vision Akash

Best Summarization Summarization scores

Majority Vote variant

show upper limit to tile-based method ¿ upper limit for summarization based method when Nworker low.

Evaluation Metrics

# of workers	5	10	15	20	25	30
# of batches	10	8	6	4	2	1

Table 1: 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).

- plot PR curves for each algorithm, compute AUC
- PR values in a table of all algorithms compared against baseline
- show that our experiment works well on smaller, noisier datasets
- Baselines:
  - summarization-based methods
    - Jaccard, PR
    - Heuristics: NumPts, Area
  - tile based majority vote
  - color-based computer vision
  - Individual worker responses

Conclusion

- Future work: model task difficulty and worker qualities across tasks

	Num Points	Area Ratio	Jaccard [Self]	Precision [Self]	Recall [Self]	Vision
Precision	0.71	0.56	0.95	0.98	0.73	0.91
Recall	0.84	0.87	0.95	0.76	0.99	0.74

NOTE: Will add additional columns from our TileEM algorithm later.

Table 2: The average precision recall compared against ground truth over all objects based on various segmentation algorithm. For the first 5 summarization based method, we pick the best bounding box based on that metric. Only the Num Point, Area Ratio and Vision algorithm doesn't require ground truth to compute.