Tool Diversity as a Means of Improving Aggregate Crowd Performance on Image Segmentation Tasks

Jean Y. Song, Raymond Fok, Fan Yang,
Kyle Wang, Alan Lundgard, Walter S. Lasecki
CROMA Lab | MISC Group
Computer Science and Engineering, University of Michigan
{jyskwon,rayfok,yangtony,wangkyle,arlu,wlasecki}@umich.edu
.

Abstract

Crowdsourcing is a common means of collecting training data, such as image segmentations, for many computer vision applications. However, designing accurate crowd-powered image segmentation systems is challenging because defining the boundaries of an object in an image requires considerable fine motor skills and hand-eye coordination that leads to some level of errors from every participant. Typically, answers from multiple workers are used to generate a more accurate combined result, but biases in how people make mistakes result in shared errors that remain even after aggregation. In this paper, we introduce an approach that leverages *multiple segmentation tools* for the same task to avoid systematic biases introduced by the tools themselves. We illustrate the efficacy of this through FourEyes, a hybrid intelligence system that leverages a set of four image segmentation tools. We show that combining worker answers from multiple tools produces more accurate segmentations than any individual tool.

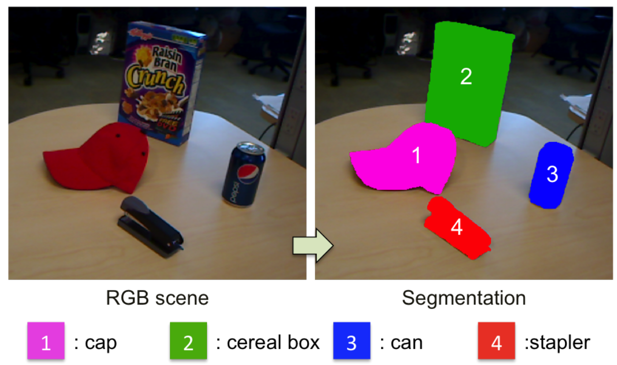

Figure 1: Example of the target image (left), the ground truth object segmentations (right), and the color codes mapped to object annotations (bottom).

Introduction

Image segmentation demarcates objects in a visual scene from the background, allowing computer vision (CV) systems to learn to recognize these specific objects. These CV systems can in turn enable autonomous cars to identify pedestrians, surveillance drones to recognize potential threats, and in-home robots to help people with motor impairments live more comfortably and independently.

Perceiving demarcations of object boundaries in visual scenes comes naturally for people, but remains a challenging open problem for CV systems due to scene semantics. Crowd-powered object segmentation tools can bridge this gap by using human understanding of scenes to produce large manually-demarcated training data sets for automated systems (Gurari, Sameki, and Betke 2016; Lin et al. 2014; Bell et al. 2013). However, designing highly accurate crowdsourcing systems that scale efficiently (with respect to cost / human time) for segmentation tasks is challenging because the manual task of tracing the boundaries requires considerable hand-eye coordination and fine motor skills that result in many errors if performed quickly. Well-designed tools (Bell et al. 2013; Gurari, Sameki, and Betke 2016; Zhong et al. 2015) and even partially automated tools (Bearman et al. 2016; Lin et al. 2016; Carlier et al. 2014) have been introduced to help workers reduce the amount of worker effort needed. These tools introduce effective new ways of helping workers do better, but none completely eliminate the difficulty of image segmentation.

In this work, we present the idea of *tool diversity* as a means of improving aggregate crowd performance. Unlike the standard aggregation methods in crowdsourcing, which search use the best single tool available with many workers to reach high accuracy, we consider the strengths and weaknesses of worker annotations using multiple tools to achieve higher combined accuracy. To illustrate the efficacy of this approach, we introduce a multi-tool crowd-powered image segmentation system (FourEyes) to demonstrate the proposed idea. We show that heterogeneous tool aggregation provides more accurate segmentations than any individual base tool, even with a simple voting strategy.

The key contributions of this work are: 1) a novel aggregation approach that combines input *across different tools* to avoid many error biases that might otherwise result from the use of any one tool alone; 2) FourEyes, a crowd-powered image segmentation system that uses *sets* of tools to outperform any constituent tool; and 3) experimental results on the effects of using multiple tools to improve performance.

Approach

Prior work has used task decomposition—the process of breaking down larger tasks into more manageable, focused pieces of work called subtasks—to make tasks more approachable for non-expert crowd workers. Once task decomposition has been used to break down a larger unit of work as much as possible within a corresponding workflow, most crowdsourcing systems then use multiple workers in parallel to improve accuracy further by aggregating their answers. Our proposed approach fills in the gap where traditional task decomposition leaves off. When a task (or subtask) can no longer be broken down, we propose using multiple different tools across different workers to complete the same [sub]task, instead of having all parallel workers complete the same task with the same interface or tool.

While we demonstrate this new crowdsourcing paradigm using an image segmentation task, it can benefit any task where different approaches to solving the same problem can be devised. Specifically, tasks that have the following properties would be especially amenable to our approach:

1. The task response correctness is cumulative with worker input. In other words, quality improves (converges to correct) as more worker inputs are collected. Problems where majority voting works would belong to this class.

2. The task has an objectively correct answer (i.e., it is not subjective), but also tolerates imperfections in workers' responses. For example, tasks like creative writing do not have a single correct answer, and thus cannot be aggregated. In general, if aggregation is possible, our general approach can be applied (although we only demonstrate this in a single domain within the scope of this paper).

3. The task is tractable enough to yield close-to-correct responses from workers, but responses can be expected to have high chance of imperfection. That is, tasks for which humans are good at providing decent heuristic responses would benefit most from our approach.

4. The expected human error is distributed differently between tools. This way, the diverse tool set can complement a broad range of error types. If this were not the case (i.e., if the error were all biased in the same direction), then multiple tools would not be more effective than one.

Many common crowdsourcing problems (for example, in language processing and information annotation) (Lasecki et al. 2014) have such properties, suggesting that a range of domains beyond the ones explored in this paper may also be able to benefit from our approach. In the following sections, we introduce FourEyes to demonstrate that our crowdsourcing paradigm is beneficial to image segmentation tasks as one example of the potential of this approach.

System Design

FourEyes consists of four crowd-powered object segmentation tools (Figure 2), each with different levels of input required from workers to complete the task (different levels of autonomy). The first tool, **Basic Trace**, allows worker to draw boundaries of objects by holding the mouse button, which is a method commonly found in manual image segmentation tools (Gurari, Sameki, and Betke 2016). The second and third tools, **Drag-and-Drop** and **Pin-Placing**, are motivated by image registration techniques and use less manual interaction as compared to Basic Trace. For the template-based tools, we construct an icon list by downloading images of a particular object from an established image search engine like Google or Bing. These icon images are then filtered for transparency and size, and the first ten are used to construct each icon list. Workers are asked to select the icon that most accurately matches that object in the scene based on the shape, proportion of dimensions, and perspective. Drag-and-Drop allows workers to drag the icon image to place it in desired location, and rotate/scale to best align it with the object in scene. Pin-Placing allows worker to click four locations on their selected icon, and pair them with four corresponding points on the object in the scene. Then an automatic transformation algorithm will run to transform icon image to align corresponding points. The fourth tool, **Floodfill**, requires the least manual interaction. Workers are first asked to click on the object they want to segment, which triggers a flood fill algorithm to highlight all neighboring pixels sharing a RGB value similar to the RGB value of the pixel that was clicked. Workers can then adjust a slider to modify the algorithm's color tolerance parameter.

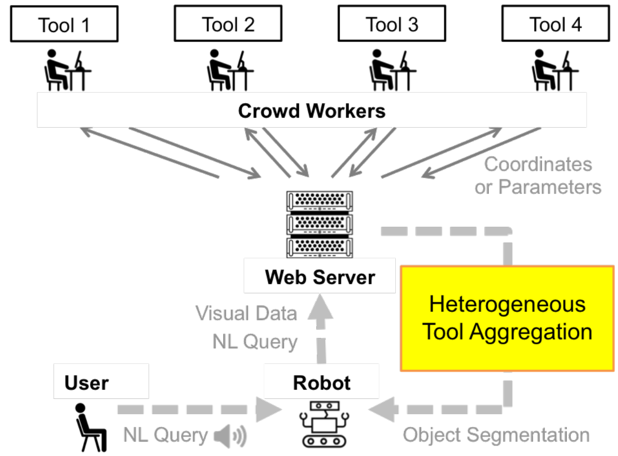

Figure 2: FourEyes aggregates workers' answers from heterogeneous tools. This improves segmentation accuracy by complementing systematic error biases that might otherwise result from the use of a single tool.

Experimental Settings

To understand the effect of multi-tool aggregation, we conduct an experiment with input from crowd workers recruited from Amazon's Mechanical Turk platform. Our data set included 12 different visual scenes, each containing three to seven objects, totaling 51 objects. The scenes were gathered from publicly-available data sets [1,2], and represented typical indoor scenarios with commonplace objects.

Each worker was shown one scene and a list of objects to segment. For each task, the order that the objects were listed in was randomized to avoid bias. FourEyes provided work-

[1] https://rgbd-dataset.cs.washington.edu/dataset.html/

[2] https://www.doc.ic.ac.uk/ ahanda/VaFRIC/iclnuim.html/

ers with one of the tools described above (Basic Trace, Drag-and-Drop, Pin-Placing, and Floodfill) to complete the object segmentation task for all objects in the queue. We recruited six unique workers for each tool-scene pair (288 workers total), resulting in a total of 1224 object segmentations.

Before crowd workers begin the task, they are shown a short instructional video demonstrating the goal of the task, and how to use the tool they will be given to use. They are then shown the FourEyes interface containing the visual scene, name of object to be highlighted, and a task timer. This timer does not impact workers other that to serve as an encouragement to consider time in their work. Task instructions are also accessible at any time if necessary. Each worker was paid between \$0.35 and \$0.60 per task, depending on the number of objects they had to segment or on the level of difficulty of given tool (a pay rate of ~\$10/hr).

Results and Discussion

To measure success on the image segmentation task, we primarily care about the accuracy of the resulting segmentation and the total effort required from the workers (latency). To measure accuracy, we use precision, recall, and F_1 score (the harmonic mean of precision and recall). To calculate these measures, we manually generated a ground truth segmentation for each object in each scene (as in Figure 1). Precision and recall of worker responses were both measured using per-pixel comparisons between worker answers and the ground truth. F_1 is computed from the same measures (e.g., true positive rate) as precision and recall. To calculate latency, we measure overall task time starting from when the worker starts interacting with the task to when the worker clicks 'submit' at the end of the task.

Performance of Individual Tools

There was a statistically significant difference in accuracy measures across the different tools (all $p < 0.0001$). Floodfill's precision was significantly better than the other three tools. On the other hand, its recall was significantly worse than the other three tools. The tool with the highest F_1 score was Basic Trace, performing significantly better than the other three. We observed that with Basic Trace, Drag-and-Drop, and Pin-Placing, workers tended to select objects by putting large margins around the objects, resulting in high recall but low precision. On the other hand, Floodfill gave high precision but low recall because the selection area tended to be smaller than the actual object boundaries due to boundaries that were shaded or colored differently.

Multi-Tool Aggregation

We explored the aggregation result of two different team sizes (four workers and six workers) and all possible agreement thresholds. We implement a pixel-level uniform voting algorithm, with each answer weighted equally. For four workers, we tested agreement thresholds of 25%, 50%, 75%, and 100%. For six workers, we tested agreement thresholds of 16.7%, 33.3%, 50%, 66.7%, 83.3%, and 100%. Notably, the two extreme thresholds (lowest and highest) always give poor F_1 score (under 0.7) regardless the team size or tool pair. Since the extreme cases were so inaccurate, the rest of our experiments used only moderate agreement thresholds.

	Team Size 4				Team Size 6		
	Prec.	Recall	F_1		Prec.	Recall	F_1
T_1	0.679	**0.989**	**0.759**	T_1	0.606	**0.990**	**0.728**
T_2	0.630	0.943	0.725	T_2	0.591	0.940	0.679
T_3	0.633	0.840	0.639	T_3	0.608	0.848	0.593
T_4	**0.856**	0.654	0.691	T_4	**0.830**	0.664	0.679

(a) Homogeneous tool aggregation

	Team Size 4				Team Size 6		
	Prec.	Recall	F_1		Prec.	Recall	F_1
T_{12}	0.689	**0.888**	0.750	T_{12}	0.696	**0.929**	**0.774**
T_{13}	0.662	0.882	0.725	T_{13}	0.683	0.862	0.730
T_{14}	**0.818**	0.853	**0.806**	T_{14}	**0.831**	0.792	0.771
T_{23}	0.621	0.838	0.687	T_{23}	0.648	0.837	0.697
T_{24}	0.791	0.794	0.755	T_{24}	0.780	0.796	0.739
T_{34}	0.800	0.728	0.722	T_{34}	0.809	0.697	0.690
				T_{123}	0.660	0.896	0.722
				T_{124}	0.795	0.845	0.774
				T_{134}	0.778	0.814	0.749
				T_{234}	0.766	0.780	0.722

(b) Heterogeneous tool aggregation

Table 1: Average accuracy across different levels of agreement thresholds. The performance pattern was consistent in different team sizes.

Homogeneous Aggregation As a baseline, we explore segmentation accuracy of homogeneous aggregation (same-tool aggregation). The statistical result of the baseline is shown in Table 1(a). For a compressed summary, each team size is averaged across different agreement thresholds. The abbreviations T_1, T_2, T_3, and T_4 represent Basic Trace, Drag-and-Drop, Pin-Placing, and Floodfill, respectively. The performance of tools was consistent in different team sizes. For both team sizes, combining answers from T_4 gave the highest average precision, and combining answers from T_1 gave the highest average recall and F_1 score.

Heterogeneous Aggregation We then combined workers' answers from *multiple* segmentation tools for the same task. We tested all possible two- and threee-tool pairs. Table 1(b) shows the results of thes combinations. The term T_{ij} represents combination of T_i and T_j, where $i, j = 1, 2, 3, 4$. Note that the three measures for each team size is averaged across different agreement thresholds.

The results show that heterogeneous aggregation improves F_1 score in both team sizes compared to homogeneous aggregation. The maximum F_1 score for homogeneous aggregation was achieved by Basic Trace, and the values were 0.759 and 0.728 for team size four and six, respectively. The maximum F_1 score for heterogeneous aggregation was achieved by Basic Trace $\times$ Floodfill for team size four (0.806) and by Basic Trace $\times$ Drag-and-Drop for team size six (0.774). For both team sizes, heterogeneous aggregation performed better.

Team Size	Voting Threshold	Best Homo	Best Hetero	p-value
4	**50%**	T_4 0.742	$\boldsymbol{T_{14}}$ **0.837**	**0.00143 ($p < 0.005$)**
	75%	T_1 (0.776)	T_{14} (0.776)	0.989
6	33.3%	T_4 0.763	T_{14} 0.802	0.182
	50%	T_1 0.759	$\boldsymbol{T_{14}}$ **0.824**	**0.00168 ($p < 0.005$)**
	66.7%	T_1 0.825	T_{124} 0.835	0.665
	83.3%	T_1 0.797	T_{12} 0.783	0.729

Table 2: The best performing homogeneous tools and heterogeneous tool pairs and their F_1 scores. We ran an ANOVA test to check the statistical significance.

To compare the statistical significance, we ran an ANOVA test on F_1 scores for each agreement threshold. Table 2 shows the best performing homogeneous and heterogeneous tools for each threshold. From heterogeneous tool aggregation, we get a 9% improvement ($p < 0.005$) when agreement threshold is 50%, and no significant decrease in performance in any case. Notably, 50% agreement was not only the case where the heterogeneous pair performed significantly better than the homogeneous pair, but also the case that returned the highest average accuracy across all conditions.

From our experiments, we observe that high-precision (but low-recall) and high-recall (but low-precision) tool pairs gives the highest average F_1 scores. This appears to be a precision-recall trade off that traverses the accuracy space in a more promising point. A related phenomenon was observed in a study investigating effects on accuracy from financial incentives (Mao et al. 2013). Their discussion focused on trading off precision and recall with different payment schemes. In our work, we go further and seek *how* to optimize the performance with a simple uniform voting strategy. We believe that more generally, different tools can compensate for various types of inherent individual systematic error biases.

Conclusion and Future Work

Our study demonstrates that tool diversity can improve aggregate crowd performance on image segmentation tasks. The primary results demonstrate that combining workers' answers from different tools with different systematic error biases produced more accurate segmentations than *any* individual tool, even with a simple uniform voting strategy.

Future work may model the tool diversity problem as a joint optimization problem that maximizes the expected output based on the variations and biases of *both* workers and tools. It could also use **real-time crowdsourcing** to enable image segmentation (or other tasks) in real-time (Bigham et al. 2010; Lasecki et al. 2012) using different tools/interfaces. Better understanding how tool diversity generalizes to other domains holds the promise of creating a new, powerful and complementary crowdsourcing approach.

Acknowledgements

Anonymous for review.

References

Bearman, A.; Russakovsky, O.; Ferrari, V.; and Fei-Fei, L. 2016. Whats the point: Semantic segmentation with point supervision. In *European Conference on Computer Vision*, 549–565. Springer.

Bell, S.; Upchurch, P.; Snavely, N.; and Bala, K. 2013. Opensurfaces: A richly annotated catalog of surface appearance. *ACM Transactions on Graphics (TOG)* 32(4):111.

Bigham, J. P.; Jayant, C.; Ji, H.; Little, G.; Miller, A.; Miller, R. C.; Miller, R.; Tatarowicz, A.; White, B.; White, S.; et al. 2010. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, 333–342. ACM.

Carlier, A.; Charvillat, V.; Salvador, A.; Giro-i Nieto, X.; and Marques, O. 2014. Click'n'cut: Crowdsourced interactive segmentation with object candidates. In *Proceedings of the 2014 International ACM Workshop on Crowdsourcing for Multimedia*, 53–56. ACM.

Gurari, D.; Sameki, M.; and Betke, M. 2016. Investigating the influence of data familiarity to improve the design of a crowdsourcing image annotation system. *HCOMP*.

Lasecki, W.; Miller, C.; Sadilek, A.; Abumoussa, A.; Borrello, D.; Kushalnagar, R.; and Bigham, J. 2012. Real-time captioning by groups of non-experts. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, 23–34. ACM.

Lasecki, W. S.; Gordon, M.; Koutra, D.; Jung, M. F.; Dow, S. P.; and Bigham, J. P. 2014. Glance: Rapidly coding behavioral video with the crowd. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, 551–562. ACM.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. *Microsoft COCO: Common Objects in Context*. Cham: Springer International Publishing. 740–755.

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

Mao, A.; Kamar, E.; Chen, Y.; Horvitz, E.; Schwamb, M. E.; Lintott, C. J.; and Smith, A. M. 2013. Volunteering versus work for pay: Incentives and tradeoffs in crowdsourcing. In *First AAAI conference on human computation and crowdsourcing*.

Zhong, Y.; Lasecki, W. S.; Brady, E.; and Bigham, J. P. 2015. Regionspeak: Quick comprehensive spatial descriptions of complex images for blind users. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2353–2362. ACM.