Evaluating Methods for Improving Crowdsourced Annotations of Images Containing Lego Bricks

Rembrandt Oltmans[1] , **Attila Lengyel**[1] , **Jan van Gemert**[1]

[1]TU Delft

r.f.a.oltmans@student.tudelft.nl, {a.lengyel, j.c.vangemert}@tudelft.nl

Abstract

Data collection by means of crowdsourcing can be costly or produce inaccurate results. Methods have been proposed for solving these problems. However, it remains unclear what methods work best in scenarios with multiple similar objects of interest present in the same image, which is important for training computer vision with applications such as automatic quality control in factories. We researched which parameters are important to optimize, which methods are worth considering and what those selected methods score with regard to the parameters cost and quality. This was done through a literature review and substantiated by an experimental crowdsourcing campaign that focused on the annotation of Legos in images. It was found that the parameters to optimize were cost, optimized by reducing the time workers spent on tasks, and quality, optimized by improving the mean intersection over union value of the annotations. We concluded that majority vote, rejecting workers, majority vote adjusted to be resistant to outliers, rejecting workers with the same adjustments and decomposing tasks were the most promising methods. From our experiment we concluded that a clear trade-off exists between cost and quality. The adjusted rejecting workers method, that uses worker credibility, showed to have the highest mean quality. While the method that decomposed the components of the task and distributed them was the cheapest method to use overall and also best when looking at mean quality over cost, it was worse quality wise. These results were similar to the expected performance of the methods. From this we concluded that the best method for crowdsourcing is dependent on the error tolerance of the computer vision model that will be used and the budget available.

Keywords: Crowdsourcing, Image annotation, Annotation quality, Annotation cost

1 Introduction

Crowdsourcing has been proposed as a cheap and fast solution for solving various problems in the field of data collection. One such problem in the field of data collection is the collection of data sets for the use in machine learning applications. Annotating images, which is part of data collection, can be costly or might produce inaccurate results. Crowdsourcing is the process of outsourcing small tasks to a crowd of workers. Several platforms for crowdsourcing exist, some of the most used are Amazon Mechanical Turk [1] and Clickworker [2]. Crowdsourced annotations can be more accurate than machine-generated annotations [3] but wrong or malicious answers can be provided by the workers. The performance of machine learning approaches is highly dependent on the quality of the training data [4]. Hence methods that improve the quality of crowdsourced annotations and that bring down the costs are beneficial for the machine learning field.

Crowdsourcing poses three main problems: quality control, cost control and latency control [5]. In the case of image annotation, latency control can usually be ignored. Previous research has been done into solving these main problems. Methods for improving the quality of the annotations exist and can work by detecting bad workers and removing them [6], using structured labelling solutions [7] for improving the final labels or by using crowdsourcing itself for breaking up complex tasks [8]. Moreover, methods have been proposed for reducing cost by active learning [9] with only necessary images getting annotated, embracing errors [10] in the initial collecting stage and minimizing the number of workers needed for majority vote [11] by simulating one of the workers using computer vision.

In most of these papers however, the objects that get annotated have completely different characteristics, or the images only contain one object of interest to annotate. For some applications of machine learning, such as quality control, the objects can have similar characteristics and have multiple objects in the same image. An example of this would be quality control on a factory line. It is unclear what methods can be used best, for improving the quality and decreasing the costs of crowdsourcing the annotations in such scenarios.

This paper proposes what parameters should be optimized in these multiple similar objects scenarios. Furthermore, it will show what methods are worth considering and what the collected methods score theoretically and experimentally concerning the parameters. The focus will be on the problem of optimizing collecting annotations of images, with the images having multiple similar objects in the same scene.

The methods are supported by a literature study and substantiated experimentally by applying the collected methods on a practical case. The practical case will be the crowdsourcing of annotating images, with the images being used for the identification of multiple Legos in one image.

The remainder of the paper will be structured as described in Figure 1. First, in section 2, the parameters and annotation

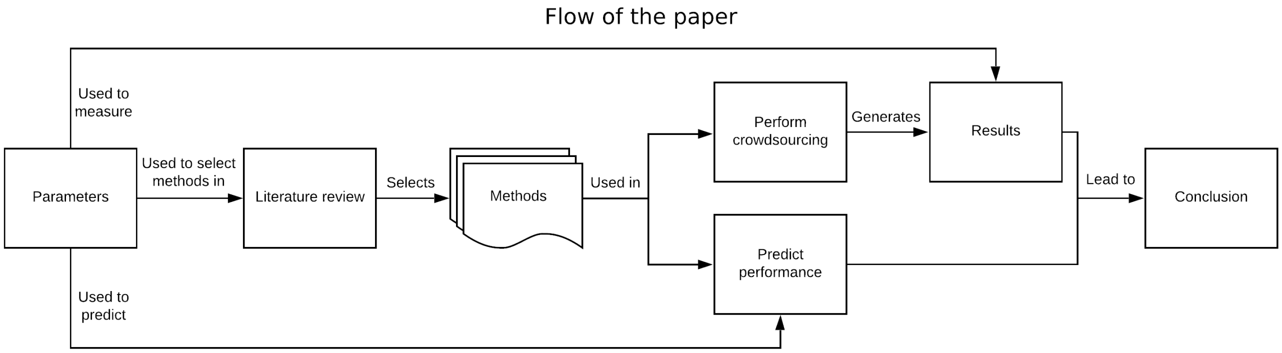


Figure 1: Flow-diagram showing the structured approach of the paper.

form are decided. In section 3 the existing work is reviewed, leading to the selection of the most promising methods from the literature, that are described in section 4. The methods are then evaluated in section 5 on the selected parameters. In section 6 the correctness of the evaluation will be reviewed by testing the methods on real crowdsourcing data. Section 7 describes the ethical aspect of crowdsourcing and the reproducibility when using human workers. Leading to the conclusion in section 8.

2 Methodology

2.1 Annotation type

The annotation of the images consist of a minimum bounding box around each object and a correct label for that object. The tight bounding box needs to enclose the entire object of interest, with only a few pixels margin, to improve the accuracy of the machine learning algorithms using it [12].

A minimum bounding box with a label is used because it is both the most popular method, [12] and the other widely used method pixel segmentation, is time-consuming and expensive to employ [13]. To speed up the process of annotation, an interesting method for creating the tight bounding boxes is used [14]. This method speeds up the process of annotation without quality loss, which is beneficial for all methods that will be tested. The label of the annotation consists of a string that uniquely defines that object. In this paper, a unique identifier is used that is provided by the producer of the blocks [15].

2.2 Parameters

The methods will be judged on relevant parameters to provide a meaningful comparison. The parameters that are deemed relevant are cost and quality [5]. Latency control is not relevant as data sets are generally not needed instantaneous.

The parameter cost will be represented by the time spent by the workers. This will be denoted by time T needed for verification *T(verification)*, drawing *T(drawing)* or labeling *T(labeling)*. Factors such as expertise needed by the workers are considered as a part of the cost. However, those factors stay constant throughout all considered methods and therefore will be ignored. Cost from the crowdsourcing platform will also be excluded as that is dependent on the platform and free or custom platforms could be used. Furthermore it is assumed that the reward provided for the tasks does not directly influence the quality of the annotation as theorized by previous research [16], although workers should still be enticed to perform the task. Thus, resulting in the parameter, cost only being dependent on the time spent by the workers.

Quality will be represented by the Intersection over Union [14], IoU for short, for the individual annotations. This will be measured in two ways, as mean IoU denoted as mIoU, and with the number of image annotations that have an mIoU of above a certain threshold. IoU is used because it not only considers both the amount that the bounding boxes overlap, but it also accounts for the size of the union. Therefore, a worker who in an extreme case could give a bounding box the size of the image, will not be awarded a high IoU score. Even when the bounding boxes would overlap entirely.

2.3 Selection and comparison

The collecting of methods will be performed by taking state-of-the-art methods from recent papers on crowdsourcing, that fall under promising categories of methods considering the desired use case. To decide what kind of methods are interesting to consider the most relevant categories of methods will be determined. This will be determined by taking the possible range of categories from a recent survey [17] that categorized the field of quality improvement in crowdsourcing data. Subsequently, it will be decided, with the use of exclusion criteria, what the most relevant categories are. Furthermore, a baseline method will be taken to compare if the methods that were selected perform better than the baseline.

The methods will be compared first theoretically by analysing the methods and scoring them on the chosen parameters using, for example, that workers spent less time verifying than drawing a box [18]. Subsequently, the performance of the individual methods will be tested experimentally. Using the methods for annotating and improving the quality of a batch of images, while crowdsourcing. The batches of images are drawn without replacement from the total set of images of Legos. The final annotations, produced by applying or using the methods, will be compared to gold-standard test data. The gold-standard test data will be assumed correct.

3 Relevant work

The categories considered for assuring quality were taken from a recent survey [17] on quality control in crowdsourcing. The methods can be seen in the linked survey in Table V of the appendix reference maps. To limit the number of methods that will be tested to a reasonable amount, methods that are unsuitable are not tested. The following categories are considered to be unsuitable for various reasons.

3.1 Related methods

First a few related categories will be reviewed and it will be argumented why they are not valid for application in the desired use case, whereafter the exclusion criteria will be presented that exclude multiple other categories.

Embracing errors

Quality assurance methods generally improve the quality of annotations by reducing the number of errors. However, by considerably speeding up the process of annotating while not focusing solely on quality, many more annotations could be produced for the similar costs [10]. In combination with other methods, the quality of the annotations could be improved resulting in a higher quality and lower cost data set, depending on the speedup and quality of the original annotations.

This method seems promising to use however the papers that focus on this subject available e.g. Krishna et al. [10] are not compatible with the form of annotation with box annotations. However, creating error embracing methods applicable with the multi object use case would be considered an interesting research topic for future studies.

Filter outputs

Instead of assigning the same task to multiple workers, the output of a single worker or algorithm could be reviewed by other workers through reviews from peers, the requester or by experts. To reduce the number of workers even further the output could instead be generated by a computer.

A state-of-the-art algorithm [18] was proposed by Konyushkova et al. that uses the concept of filtering outputs. The algorithm automatically decides if it is better to let the worker draw a bounding box or validate an automatically generated bounding box. The algorithm reduces the number of bounding boxes that will be drawn by the workers manually, which lowers the total costs.

The performance of the method was only tested theoretically in the paper. The groundtruth was used to simulate a worker drawing and verifying the bounding boxes. Meaning that the assumption was made that all workers provide a 100 percent correct action, which is far from the truth. The method also provides no verification of the workers drawing or validation, therefore this method will not be used.

Aggregate outputs

A popular method for improving the quality of crowdsourced annotations is by assigning all tasks to multiple workers and then merging the answers. The baseline approach for this is by using *majority vote* [19], however other methods have proven to be more successful in certain situations [20].

Branson et al. provides a method [11] that works on the concept that when two workers agree on the same pixel locations for the bounding box, then the chance that it is a coincidence is small. Instead of letting workers agree with each other, the method works by also letting workers agree with a trained computer vision model. Therefore the number of workers for each task is reduced.

The method however has a weakness, by using a computer vision model to train itself bias is introduced [21]. The method can be seen as a semi-supervised self-training model as only images that get drawn by the workers are labeled and the model generates bounding boxes itself for each task. When workers propose bounding boxes with an common mistake, the model will also start proposing bounding boxes with the same mistake, and will only agree with annotations of workers that make the same mistake. Therefore this method will be excluded.

3.2 Exclusion criteria

Several categories of methods will not be used for varying reasons. Primarily because of the use of multiple labels and bounding boxes per image, several labels could be slightly or entirely misplaced on the image. Therefore annotations can be partially wrong instead of binary. Methods, such as asking binary questions [22], can therefore not be used or would perform inadequately. Methods that focus or depend on the use of relational data [23] e.g. when a label with ”sky” is present a label ”bird” is often also present will not be considered as this relation cannot be assumed with the desired applications. Furthermore, methods that focus solely or depend on discrepancies between the content [24, 25] of ”free labels” given by the workers and conflict resolution [26] for this are also not considered . Concrete identifiers are given for the objects thus no bias or subjective naming of objects can occur.

Jobs will be assumed to be instantly available and workers will be assumed to be able to take breaks themself, therefore methods for reserving workers [27] and introducing breaks [28] will not be considered. Task control order and intertask coordination [29] are also not needed as tasks do not depend on each other for results. Furthermore splitting batches up [30] and flooding the tasks list with small batches is not seen as sustainable and beneficial and therefore will not be considered as a viable category of methods. Methods that use the amount of payment per task or use a dynamic reward [31] will not get considered, as financial benefits do not give certainty for higher quality annotations [16]. Social transparency and sharing of the purpose [32] could be used to create more trust between the workers and the requester. However, this can not be assumed when a crowdsourcing campaign is an one-time event and not all assumed applications have a great social purpose that will lead to better results.

Recommending or promoting tasks [33], situated crowdsourcing and recruiting teams [34], priming and teaching workers [35], gamification [36], assigning better workers to harder tasks [27], providing feedback [37] and filtering workers based on factors like age, profession or motivation [38] will be assumed out of scope, even though they could provide added benefit in some cases.

4 Methods

By taking into account the exclusion criteria and combining the categories lower complexity, separate duties, iterative improvement and cleansing data into the decomposing tasks category, the following categories are considered:

Reject workers

The financial incentive of completing a task leads some workers to spam answers or to collude with other workers [39]. Although collusion in this case is not beneficial for workers, as copying the tasks takes as long as completing them and bots are automatically detected by most platforms. Detecting bad workers or spammers and then rejecting them could be beneficial for the quality of the annotated data set. Some methods use the duration the worker spends on the task to detect spammers but that can provide false positives. Additionally, spammers have been reported cooking the timer [40], by opening the task and letting it sit for awhile, preventing such methods from being effective.

A method that avoids these problems [41] uses a form of majority vote to assign metrics to the workers involved. The metrics that are assigned such as credibility of the worker decide what weights get assigned to their answers. Malicious workers get detected and their answer can be excluded completely. Therefore workers that provide bad or completely arbitrary answers will not be given the same weight in the voting process, thus resulting in higher quality annotations. The penalty of answering a question wrong, is decided by the certainty that the answer that is voted for is correct. Thus getting a hard question wrong will not be as punishing as getting an easy one wrong. However as the bounding box annotation style can have numerous answers, this will usually lead to every answer being only voted correct by the worker proposing it. Therefore the certainty of the answer's correctness is calculated by the mIoU of the worker boxes in regard to the final weighted majority voted box. To check if the workers proposed bounding box in the end is correct, the IoU with regard to the final box will be used. When the IoU is above a certain threshold it is considered correct.

Reject workers plus

The *rejecting workers* method does not take into account what the location of the current worker their bounding box is in relation to the other workers their boxes. Therefore instead of only relying on the credit of the worker for calculating their weights in the majority vote, the similarity of the worker their box in respect to the other answers will be used. This adds the certainty of a workers proposed box being right before the weighted majority vote. Hence adding resilience against outlier answers while retaining all advantages of the regular *rejecting workers* method.

Decompose tasks

By decomposing tasks, multiple workers are involved in a single task. Bad annotations could therefore be detected by other workers instead of by algorithms. The method can be applied, without any added costs, in tasks that can be split into parts when the new tasks take as long as a single worker doing the original task. This reduces the complexity of the tasks, which decreases the chance that workers make errors. Multiple frameworks exist for task decomposition [42]. However, as the annotation in the use case considered can only be realistically divided into two parts, the drawing of the bounding box and the labelling of the box, a framework would add unnecessary complexity.

To prevent that unnecessary complexity, an approach [43] will be used that first lets a worker draw the bounding boxes and subsequently a different worker will label the boxes. Through the use of clicking on the theoretical centre of the bounding box with the label selected. This method, that is commonly used for the supervision of training object detectors, can also be used for the supervision of workers. The worker clicks on the theoretical centre of the bounding box without seeing the proposed box. Through measuring the distance between the click and of the proposed bounding box centre the likely-hood of correctness of the bounding box and click is gained. This information will be used to improve the quality of the final bounding box data without any added cost.

Majority vote

The most used [44], and therefore baseline approach, for improving quality in crowdsourcing is *majority vote*. Multiple workers their answers are aggregated with the same weight, to improve the quality of the resulting data set. However this method is more difficult to apply in a case with many similar answers that are all correct. Therefore in the case of bounding boxes, the bounding boxes will be reconciled into one average bounding box [13].

Majority vote plus

The naive implementation of the baseline method *majority vote* contains the weakness that outliers have a big impact on the accuracy of the final bounding box location and size. Therefore an improved method of *majority vote* is proposed that is more resilient against outliers. Boxes that have a summed IoU value of zero in respect to the boxes submitted by other workers will not be taken into account when reconciling the bounding boxes into one average box. Preventing outliers from having an impact on the IoU value of the final weighted bounding box.

5 Method evaluation

First the quality of the selected methods will be estimated by looking at which techniques for improving quality are used by the methods. Then the cost will be estimated by looking at how the methods work and what the minimum and maximum time is that workers can spend on one assignment.

5.1 Quality estimation

Because of the lack of ground truth bounding boxes with labels an assumption [4] is made that the average IoU is above 0.5 for the workers. That is, we assume that more than half of the workers in the batch provide a correct annotation. A correct annotation exists of a bounding box with an IoU value of 0.5 or higher [11, 43]. From this assumption we can derive that while using a form of majority vote, in more than half of the cases there are more workers giving correct answers than wrong answers.

Rejecting workers removes workers that have given a ratio of bad annotations above a certain threshold deemed malicious. Furthermore, it assigns worse credit to workers that give bad annotations. Because the weighted majority vote used by the *rejecting workers* method uses the worker credit as weights, inaccurate workers their annotations are assigned less weight than good workers. Therefore the resulting annotation quality should improve. Because the method also uses majority vote, it has the same problem as the baseline method, where outliers can severely influence the final result. Thus during the crowdsourcing the model will perform worse when numerous outliers are present. When spammers complete large amounts of tasks the method will perform better in relation to most other methods.

Rejecting workers plus works the same as the regular *rejecting workers* method however it mitigates the problem that outliers severely influence the final result. Thus it will excel in the same cases as the regular *rejecting workers* method. However it will perform better when outlier bounding boxes are present in the data.

Decomposing tasks achieves by decomposing the task, that the location of the drawn bounding box is verified by the worker labelling the boxes. Thus in case that the drawing and labelling worker both do not choose the same wrong location to use, the task can be repeated and improved. This is however only true for the bounding box, the label is not verified by another worker. Thus this method might perform poorly when the workers that label the images assign the wrong labels. Furthermore the performance of this method is dependent on the uncertainty that the distance between the centre click and the centre of the bounding box is related to the quality of the annotation.

Majority vote depends on the assumption that more than half of the workers provide the same correct answer. In the case with bounding boxes, this is however somewhat more difficult because many answers, with different bounding boxes, could be seen as correct. A naive approach for combining the provided answers into what the majority chose is through taking the mean of all provided bounding box answers. However this method is susceptible to outlier bounding boxes. Thus if in the real crowdsourcing campaign many outliers are present naïve *majority vote* will perform poorly.

Majority vote plus achieves through not taking bounding boxes into account, that have an IoU value of zero with all similar bounding boxes submitted by the other workers, that it is less susceptible for outliers. It also uses the same technique that taking the average of multiple samples differs less [45], also colloquially known as the wisdom of the crowd. Resulting in the expectation that *majority vote plus* will perform better than the regular version, because of outlier protection.

In summary, it is expected that methods using more techniques for improving the quality of the annotations perform better than methods that only use a subset of those techniques. In Table 1 an example of this is that *majority vote plus* is expected to perform better than the regular *majority vote* because the improved version uses the same technique of using averages and has 2 more techniques for improving quality. In section 6 it is tested if these hypotheses hold in practise.

Quality improvement techniques of the methods

	Resilient against outliers	Rejects bad annotations	Removes bad workers	Uses average
Rejecting workers	No	No	Yes	Yes
Rejecting workers plus	Yes	Yes	Yes	Yes
Decomposing tasks	Yes*	Yes*	No	No
Majority vote	No	No	No	Yes
Majority vote plus	Yes	Yes	No	Yes

Table 1: Techniques used by the selected methods for improving quality. A method such as *Rejecting workers plus* uses far more techniques for improving the quality in comparison to *majority vote*. *Only bounding boxes, not labels.

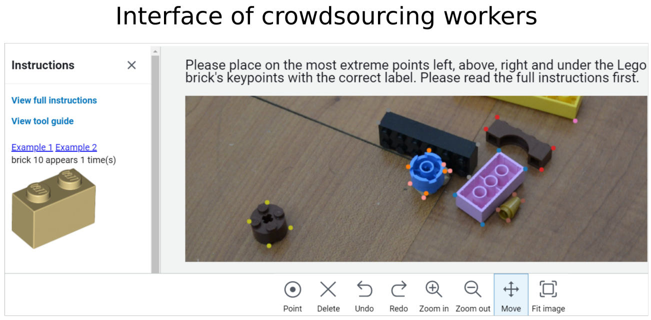
5.2 Cost estimation

The cost of the methods is decided by the time T workers need for verification *T(verification)*, drawing *T(drawing)* and labelling *T(labelling)*. The term ”single task” will express the drawing and labelling of one bounding box. The cost for a single task performed by a method is denoted by *C(method)* and is the cost per time unit for the workers multiplied with the time the method takes.

Majority vote and *majority vote plus* both make, similar to the *rejecting workers* method, use of n workers that perform the same drawing and labelling task. However, the amount of workers that perform these tasks is always the same, namely n. Thus the time for every task with the baseline and the *majority vote plus* approach is $T(MajorityVote) = n * (T(drawing) + T(labelling))$ with $n = maxWorkers$.

Rejecting workers and *rejecting workers plus* make use of n workers that perform the same drawing and labelling task. The value of n is larger than or equal to two since at least two workers are needed for calculating the certainty. The maximal amount of workers for a single task is dependent on the maximal amount of workers allowed and how quickly the minimal certainty is reached for each task. The worst-case amount of workers is only dependent on the maximal amount of workers allowed. Therefore the time needed for a single task with the *rejecting workers* methods is $T(RejectWorkers) = n * (T(drawing) + T(labelling))$ with $2 \leq n \leq maxWorkers$. From these formulas and equal *maxWorkers* it can be concluded that the cost of *reject workers C(RejectWorkers)* is smaller than or equal to the cost of *majority vote C(MajorityVote)* and *majority vote plus C(MajorityVotePlus)*. Thus *rejecting workers* cost the same or less than, the baseline method, *majority vote*

The *decomposing tasks* method uses one worker for drawing and one for labelling. When a labelling keypoint is placed a certain distance away from the bounding box drawn by another worker both tasks need to be redone. Since it is uncertain if the labelling or drawing is done incorrectly. This can be repeated till the maximal allowed amount of repeats n is reached. With n is equal to or larger than one. Resulting in the time needed for a single task with the *decomposing tasks* method is $T(DecomposingTasks) = n * (T(drawing) + T(labelling))$ with $1 \leq n \leq maxRepeats$. The amount of repeats is dependent on the workers quality and the distance that is chosen for having to repeat the tasks. However when *maxRepeats* is chosen equal to *maxWorkers* than with similar reasoning to the *rejecting workers* method can be derived that *decomposing tasks* cost less than or equal to the baseline and improved baseline method.


Interface of crowdsourcing workers

Instructions

View full instructions

View tool guide

Example 1 Example 2

brick 10 appears 1 time(s)

Please place on the most extreme points left, above, right and under the Lego brick's keypoints with the correct label. Please read the full instructions first.

Figure 2: Interface the workers use while annotating images for majority vote, majority vote plus, reject workers and reject workers plus. The block the worker currently has to annotate can be seen on the left, which is one of the needed features Amazon Mechanical Turk allows.

6 Experimental Setup and Results

First the setup of the experiment will be discussed. Whereafter the experiment will show what the selected methods score experimentally concerning quality and cost.

6.1 Crowdsourcing setup

The number of and which blocks are present in each image is known [46]. Workers will be asked to only provide annotations for the blocks that are present. The workers are shown the image of the block they have to label automatically through the use of querying the Lego database with the block ID, hence no knowledge of block ID's is needed by the workers. Only submissions that contain the correct number of occurrences of the labels are allowed to be submitted.

Amazon Mechanical Turk was chosen as the platform to use. It contains all features needed for running the crowdsourcing experiment, in Figure 2 some of the functions can be seen. Additionally it allows for annotating images by non-workers therefore enabling the manual annotation of the ground truth boxes on the same platform. It is, besides the functionality, also the most used platform and therefore the industry standard for crowdsourcing experiments [47].

The images were collected by photographing sets of Lego bricks that were generated by a computer to be completely random [46]. This resulted in a set of images that had a roughly even distribution of 1 to 13 bricks per image. A total of around 3000 images will be annotated, by 5 methods in total. Four methods, namely *reject workers, reject workers plus, majority vote, majority vote plus* will be used on the same data. Resulting in a 2 way split of about 1500 unique images each, drawn from the images with random sampling without replacement. An example of one of the images can be seen in Figure 2, although zoomed in. The data set is split up into 2 separate parts to maximize the number of images annotated at the end, without adding costs.

The gold standard annotations consist of 750 images sampled without replacement wherefrom each of the 2 data sets 375 images are present. The set will serve as ground truth for testing the correctness of the methods. The golden data set was annotated by an in-house team, that used a similar interface as the workers, in the Amazon Mechanical Turk's development sandbox.

To prevent workers, that submitted answers in the drawing part of the task decomposition method, to validate their own drawing in the second part of the method a qualification was awarded to the workers that participated in the drawing part. Having this qualification prevented workers from being allowed to work on the validation part of the method, resulting in that workers could not validate their own drawing.

The parameters for *reject workers* and *reject workers plus* were chosen as follows: minimal submission for being considered malicious 15, minimal maliciousness threshold 0.8, minimal IoU of the box for being considered correct 0.6 and 0.9 as the minimal threshold for certainty. These parameters were chosen as they result in the highest IoU in a test set. The minimal amount of submissions is in the form of boxes, which is around 3 images. The number of workers that perform the same task is 3 as this is the lowest odd amount of workers that can be used. All other methods use the same maximum of 3 workers, therefore keeping the total cost low.

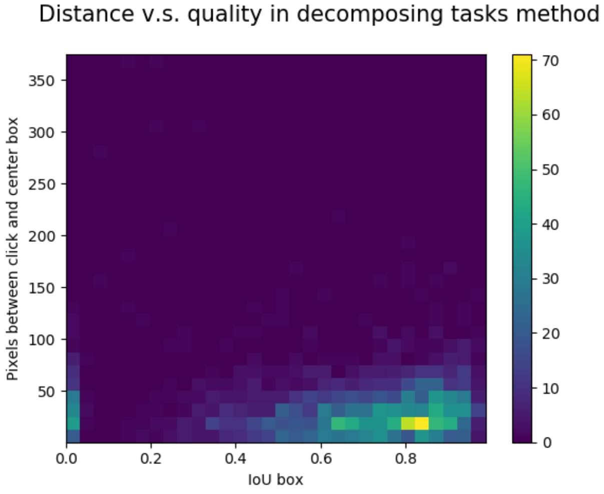

Distance v.s. quality in decomposing tasks method

Figure 3: Distance between bounding box centre and label point compared to IoU of the resulting box in decomposing tasks method. Showing that no relation exists between the quality of the annotation and the distance from the box centre to the label point.

Even though the *decomposing tasks* method has a metric that could denote the accuracy of the annotation, namely the distance between the centre of the bounding box drawn by one worker and the location of a label placed by another worker, submissions with large distances are not repeated. As it can be seen in Figure 3, no apparent relation between the quality of the final box and the distance between click and box exists. Therefore trying to improve the quality would only increase the cost without added quality benefits, as can be seen in the heat map. Rather only assignments with an click outside the box were repeated because the drawn boxes need to be labelled. When an image is not correctly labeled after 3 tries it is discarded and only taken into account for the cost, not the final quality.

A large number of boxes in the left lower half of the heat map Figure 3 can be explained by the fact that boxes get assigned an IoU of zero when they do not intersect with the corresponding ground truth. However, the part of the image that does not contain the ground truth box is generally several times larger than the part of the image that does. All boxes that are in that part of the image that does not intersect get clustered in the zero IoU bins, explaining the high number of instances in those bins.

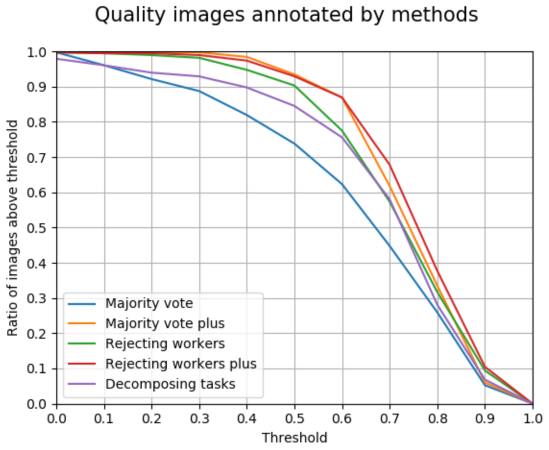


Figure 4: Ratio of images with an IoU above thresholds. All tested methods perform better overall than the baseline method *majority vote*.

6.2 Results crowdsourcing

The quality of the images produced by *decomposing tasks* is on average better than the baseline method. However from the results of the crowdsourcing in Figure 4, it can be observed that *decomposing tasks* has a lower amount of images with mIoU of above a half, thus that is considered correct by our threshold, than the other methods. Furthermore, in Figure 4 it is visible that most images rarely end up with a mean IoU of near zero as in most cases at least one box in the image is correctly annotated, yet in the case of *decomposing tasks* this does happen. This is partly caused by the fact that the *decomposing tasks* method offers no protection against wrong labelling, thus workers who label all boxes wrong in an image can easily cause an IoU near-zero. The other methods also have the advantage of averaging multiple workers, the chance that all 3 workers have near-zero annotation is small, therefore the average being also near zero is also small. Nevertheless, *decomposing tasks* is the cheapest method used, it costs almost the same as an annotation without quality improvement. Therefore, the mean correct image per dollar is higher for *decomposing tasks* than for all other methods, as visible in Table 2. This means that *decomposing tasks* can be used when cheap annotations are needed that do not need to be correct all the time, or that it could be used in combination with other quality improvement methods. The method resulted in 3 images not being used as after the maximal amount of repeats they still were not labelled correctly.

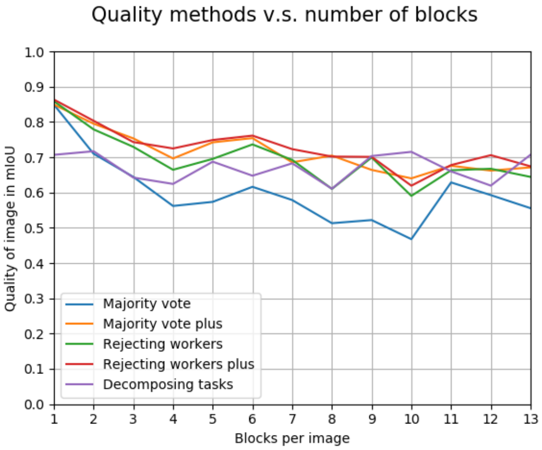
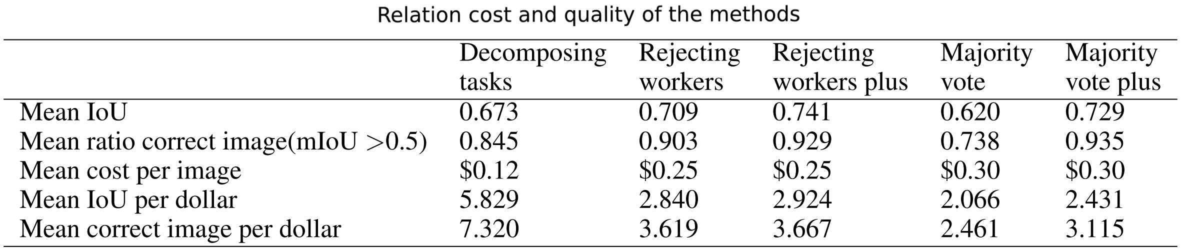


Figure 5: Mean IoU per image for n bricks. A slight decline in quality can be observed for larger numbers of blocks in an image.

From Table 2 it can be seen that *majority vote plus*, the improved baseline method, has the highest amount of correct images. Although together with the regular *majority vote* is also the most expensive, and the cost is fixed to the number of workers thus lowering the cost by changing certainty thresholds is not possible. Additionally because the method uses averaging the answers it has also fewer answers on the high end of the quality spectrum, as visible in Figure 4, and the mIoU is lower than the *rejecting workers plus* method. Furthermore, when only one block is present in the image, the higher mIoU than other majority vote containing methods is not obvious. In the case of one block, all methods that rely on a form of averaging workers submissions perform approximately the same, which implies that when only one block needs to be annotated, less outliers are present. Reinstating that images with multiple objects in an image are a different problem than with a single object.

Rejecting workers scores above the baseline method hence demonstrating that taking into account the workers past performance has a positive effect on the subsequent aggregating of bounding boxes. However, it still falls behind on *majority vote plus* which could be because of the susceptibility to outliers, a problem of which *majority vote* suffers too. The improved method for rejecting workers, *rejecting workers plus*, achieves the highest mean IoU but a lower number of correct images, but does this with a lower cost. The cost of



Relation cost and quality of the methods

	Decomposing tasks	Rejecting workers	Rejecting workers plus	Majority vote	Majority vote plus
Mean IoU	0.673	0.709	0.741	0.620	0.729
Mean ratio correct image(mIoU >0.5)	0.845	0.903	0.929	0.738	0.935
Mean cost per image	$0.12	$0.25	$0.25	$0.30	$0.30
Mean IoU per dollar	5.829	2.840	2.924	2.066	2.431
Mean correct image per dollar	7.320	3.619	3.667	2.461	3.115

Table 2: Relation between cost and quality of the selected methods. *Majority vote plus* can be seen as having the highest ratio of correct images and *rejecting workers plus* has the highest mean IoU. However, these methods are expensive to use, therefore *decomposing tasks* with the lowest cost per annotation has also the highest mean correct image per dollar. Numbers are rounded to 3 decimals.

both *rejecting workers* methods could be lowered by changing the correctness threshold, but this would also result in worse quality.

From Figure 5 it can be concluded that causation exists between the quality of the annotations and the number of blocks in the image. With more blocks in one picture the visual complexity and the number of blocks that could be mistaken for each other grow, thus it is uncertain if these same results hold for large amounts of bricks in images.

7 Responsible Research

Workers were paid $0.05 for labelling all the boxes in an image and also paid $0.05 for drawing the boxes around all Lego bricks in the image. The total reward a worker receives for an hour of annotating would unquestionably be below the minimum wage standard of most countries. This could be considered an unfair reward [48]. However because it is unknown if workers perform these tasks as a main form of income and it is unknown where most of the workers reside [49]. It is hard to calculate a fair reward while also being able to perform research with large data sets.

All resources such as code, image sets, crowdsourcing interfaces and data sets are available on GitLab [50]. Furthermore, the crowdsourcing campaign execution is described in detail in the paper. Additionally, the code for generating all figures and Table 2 is available, on the same GitLab, therefore enabling reproducing this research.

A large data set was created to mitigate the effect a single worker has on the final results. However, because of the unreliability of using humans, negligible differences in the performance of the methods could be obtained when trying to recreate the crowdsourced data.

8 Conclusions and Future Work

We have shown that in creating bounding box annotations for multiple objects in a single image it is best to minimize the cost by reducing the overall time spent by workers and to improve the quality by maximising the intersection over union value of the annotations. Additionally, we selected through the use of exclusion criteria the most promising methods that applied to the use case of multiple Legos. We looked at the theoretical cost and to what mechanisms the selected methods use for optimising the crowdsourcing. After by means of a crowdsourcing campaign the performance of the methods on images of Legos was tested.

We conclude from the data provided by the crowdsourcing campaign that *decomposing tasks* is the cheapest method for attaining correct images, and performs better than the baseline. However, it also produces a larger number of low-quality annotations compared to the other methods.

Rather a method such as the *majority vote plus* or *reject workers plus* could be used when higher mean quality annotations are required, with *reject workers plus* producing cheaper annotations than *majority vote plus* with an insignificant lower amount of correct images. These results are in line with the predictions made while reviewing the methods underlying mechanisms.

The unimproved methods *majority vote* and *reject workers* can be concluded to neither provide high-quality annotations nor provide low-cost annotations.

Future research is needed to determine if similar results hold under circumstances where more than a maximum of 3 workers are used. Moreover research should be conducted to find out if these conclusion are still relevant when large amount of blocks are in the image, with large amounts defined as 100 blocks or more.

Acknowledgements

I wish to thank various people for their contribution to making the experiments in this paper possible. Firstly my bachelor research project groupmates who took the pictures and helped with annotating them. Also, I would like to thank all anonymous workers on Amazon Mechanical Turk who annotated the images.

Author Disclosure Statement

The author has no conflicting interest.

References

[1] “Amazon mechanical turk.” https://web.archive.org/web/20200331014602/https://www.mturk.com. Accessed: 2020-04-23.

[2] “Clickworker.” https://web.archive.org/web/20200207084623/https://www.clickworker.com. Accessed: 2020-04-23.

[3] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data – ai integration perspective," 2018.

[4] Y. Liu and M. Liu, "An online learning approach to improving the quality of crowd-sourcing," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2166–2179, 2017.

[5] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, "Crowdsourced data management: A survey," *IEEE Trans. on Knowl. and Data Eng.*, vol. 28, p. 2296–2319, Sept. 2016.

[6] O. Dekel and O. Shamir, "Vox populi: Collecting high-quality labels from a crowd," in *Proceedings of the Twenty-Second Annual Conference on Learning Theory*, June 2009.

[7] T. Kulesza, S. Amershi, R. Caruana, D. Fisher, and D. Charles, "Structured labeling for facilitating concept evolution in machine learning," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, (New York, NY, USA), p. 3075–3084, Association for Computing Machinery, 2014.

[8] A. Kulkarni, M. Can, and B. Hartmann, "Collaboratively crowdsourcing workflows with turkomatic," pp. 1003–1012, 05 2012.

[9] G. Qi, X. Hua, Y. Rui, J. Tang, and H. Zhang, "Two-dimensional multilabel active learning with an efficient online adaptation model for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1880–1897, 2009.

[10] R. A. Krishna, K. Hata, S. Chen, J. Kravitz, D. A. Shamma, L. Fei-Fei, and M. S. Bernstein, "Embracing error to enable rapid crowdsourcing," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, (New York, NY, USA), p. 3167–3179, Association for Computing Machinery, 2016.

[11] S. Branson, G. Van Horn, and P. Perona, "Lean crowdsourcing: Combining humans and machines in an online system," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6109–6118, 2017.

[12] G. Rathore, W. Lin, and J. E. Kim, "Deepbbox: Accelerating precise ground truth generation for autonomous driving datasets," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 1871–1876, 2019.

[13] A. Kovashka, O. Russakovsky, L. Fei-Fei, and K. Grauman, "Crowdsourcing in computer vision," *Foundations and Trends® in Computer Graphics and Vision*, vol. 10, no. 3, pp. 177–243, 2016.

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

[15] "The lego group." https://web.https://web.archive.org/web/20200401030606/https://www.lego.com/en-us/aboutus/lego-group/. Accessed: 2020-04-28.

[16] W. Mason and D. Watts, "Financial incentives and the "performance of crowds"," *SIGKDD Explorations*, vol. 11, pp. 100–108, 06 2009.

[17] F. Daniel, P. Kucherbaev, C. Cappiello, B. Benatallah, and M. Allahbakhsh, "Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions," *ACM Computing Surveys*, vol. 51, pp. 1–40, 01 2018.

[18] K. Konyushkova, J. Uijlings, C. Lampert, and V. Ferrari, "Learning intelligent dialogs for bounding box annotation," 2017.

[19] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, (New York, NY, USA), p. 614–622, Association for Computing Machinery, 2008.

[20] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: Is the problem solved?," *Proc. VLDB Endow.*, vol. 10, p. 541–552, Jan. 2017.

[21] Y. He and D. Zhou, "Self-training from labeled features for sentiment analysis," *INFORMATION PROCESSING & MANAGEMENT*, vol. 47, pp. 606–616, JUL 2011.

[22] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie, "Visual recognition with humans in the loop," in *Computer Vision – ECCV 2010* (K. Daniilidis, P. Maragos, and N. Paragios, eds.), (Berlin, Heidelberg), pp. 438–451, Springer Berlin Heidelberg, 2010.

[23] B. Siddiquie and A. Gupta, "Beyond active noun tagging: Modeling contextual interactions for multi-class active learning," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2979–2986, 2010.

[24] T. Kulesza, S. Amershi, R. Caruana, D. Fisher, and D. Charles, "Structured labeling to facilitate concept evolution in machine learning," 2014.

[25] J. C. Chang, S. Amershi, and E. Kamar, "Revolt: Collaborative crowdsourcing for labeling machine learning datasets," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI 2017)*, ACM - Association for Computing Machinery, May 2017.

[26] M. Schaekermann, J. Goh, K. Larson, and E. Law, "Resolvable vs. irresolvable disagreement: A study on worker deliberation in crowd work," *Proc. ACM Hum.-Comput. Interact.*, vol. 2, Nov. 2018.

[27] D. R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *Advances in Neural Information Processing Systems 24* (J. Shawe-Taylor,

R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, eds.), pp. 1953–1961, Curran Associates, Inc., 2011.

[28] J. Rzeszotarski, E. Chi, P. Paritosh, and P. Dai, "Inserting micro-breaks into crowdsourcing workflows," *AAAI Workshop - Technical Report*, pp. 62–63, 01 2013.

[29] H. Jiang and S. Matsubara, "Efficient task decomposition in crowdsourcing," in *PRIMA 2014: Principles and Practice of Multi-Agent Systems* (H. K. Dam, J. Pitt, Y. Xu, G. Governatori, and T. Ito, eds.), (Cham), pp. 65–73, Springer International Publishing, 2014.

[30] D. Difallah, A. Checco, G. Demartini, and P. Cudré-Mauroux, "Deadline-aware fair scheduling for multi-tenant crowd-powered systems," *Trans. Soc. Comput.*, vol. 2, Feb. 2019.

[31] A. Finnerty, P. Kucherbaev, S. Tranquillini, and G. Convertino, "Keep it simple: Reward and task design in crowdsourcing," in *Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI*, CHItaly '13, (New York, NY, USA), Association for Computing Machinery, 2013.

[32] L. Yu, P. André, A. Kittur, and R. Kraut, "A comparison of social, learning, and financial strategies on crowd engagement and output quality," in *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '14, (New York, NY, USA), p. 967–978, Association for Computing Machinery, 2014.

[33] M.-C. Yuen, I. King, and K. Leung, "Taskrec: A task recommendation framework in crowdsourcing systems," *Neural Processing Letters*, vol. 41, 04 2014.

[34] D. Retelny, S. Robaszkiewicz, A. To, W. S. Lasecki, J. Patel, N. Rahmati, T. Doshi, M. Valentine, and M. S. Bernstein, "Expert crowdsourcing with flash teams," in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, (New York, NY, USA), p. 75–85, Association for Computing Machinery, 2014.

[35] U. Gadiraju, B. Fetahu, and R. Kawase, "Training workers for improving performance in crowdsourcing microtasks," in *Design for Teaching and Learning in a Networked World* (G. Conole, T. Klobučar, C. Rensing, J. Konert, and E. Lavoué, eds.), (Cham), pp. 100–114, Springer International Publishing, 2015.

[36] L. von Ahn and L. Dabbish, "Labeling images with a computer game," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, (New York, NY, USA), p. 319–326, Association for Computing Machinery, 2004.

[37] S. Dow, A. Kulkarni, B. Bunge, T. Nguyen, S. Klemmer, and B. Hartmann, "Shepherding the crowd: Managing and providing feedback to crowd workers," pp. 1669–1674, 01 2011.

[38] G. Kazai, J. Kamps, and N. Milic-Frayling, "An analysis of human factors and label accuracy in crowdsourcing relevance judgments," *Information Retrieval*, vol. 16, 04 2012.

[39] A. Marcus, D. Karger, S. Madden, R. Miller, and S. Oh, "Counting with the crowd," vol. 6, pp. 109–120, 12 2012.

[40] K. Sharpe Wessling, J. Huber, and O. Netzer, "MTurk Character Misrepresentation: Assessment and Solutions," *Journal of Consumer Research*, vol. 44, pp. 211–230, 04 2017.

[41] M. Nazariani and A. A. Barforoush, "Dynamic weighted majority approach for detecting malicious crowd workers," *Canadian Journal of Electrical and Computer Engineering*, vol. 42, no. 2, pp. 108–113, 2019.

[42] Y. Tong, L. Chen, Z. Zhou, H. V. Jagadish, L. Shou, and W. Lv, "Slade: A smart large-scale task decomposer in crowdsourcing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 8, pp. 1588–1601, 2018.

[43] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari, "Training object class detectors with click supervision," 2017.

[44] E. Kamar, S. Hacker, and E. Horvitz, "Combining human and machine intelligence in large-scale crowdsourcing," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '12, (Richland, SC), p. 467–474, International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[45] J. Ugander, R. Drapeau, and C. Guestrin, "The wisdom of multiple guesses," in *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, EC '15, (New York, NY, USA), p. 643–660, Association for Computing Machinery, 2015.

[46] B. Kam, "Dataset generation methods for multi-label images of lego bricks." unpublished, 2020.

[47] T. Ambreen and N. Ikram, "A state-of-the-art of empirical literature of crowdsourcing in computing," in *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, pp. 189–190, 2016.

[48] N. Salehi, L. C. Irani, M. S. Bernstein, A. Alkhatib, E. Ogbe, K. Milland, and Clickhappier, "We are dynamo: Overcoming stalling and friction in collective action for crowd workers," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, (New York, NY, USA), p. 1621–1630, Association for Computing Machinery, 2015.

[49] R. Kennedy, S. Clifford, T. Burleigh, R. Jewell, and P. Waggoner, "The shape of and solutions to the mturk quality crisis," 10 2018.

[50] R. Oltmans, "Crowdsourcing annotations lego." https://gitlab.com/lego-project-group/crowdsourcing-annotations-lego, 2020.