Faster Human-Machine Collaboration Bounding Box Annotation Framework Based on Active Learning

Minzhe Liu \(^1\) Li Du \(^1\) Yuan Du \(^1\) Ruofan Guo \(^1\) Xiaoliang Chen \(^2\)

**Abstract**

The conventional method of annotating large object detection datasets is highly time-consuming. To accelerate human-machine collaboration bounding box annotation, a visual framework with human in the loop based on active learning is proposed. Different strategies to prioritize manually annotated images are studied, compared and optimized. Our framework is evaluated on CityPersons dataset. Compared with other state-of-the-art bounding box annotation methods, our proposed approach decreases the total annotation workload by 6.2%.

1. Introduction

Preparing a dataset with rich images and annotations is essential when training neural networks, such as image classification and object detection. Although there are many open-source and easy-to-operate annotation tools, it is also highly time-and-resource-consuming. In recent years, the semantic annotation of objects has been extensively studied, such as Extreme clicking (Papadopoulos et al., 2017b), Polygon-RNN++ (Acuna et al., 2018) and Curve-GCN (Ling et al., 2019). However, in practical applications and industrial applications, bounding box annotations are still the most common.

In order to annotate bounding box faster, Adhikari et al. (2018) proposed a semi-automatic bounding box image annotation method, which divides the dataset into two parts randomly, the first part is manually annotated, and the second part is annotated according to annotation proposals generated by a model trained with the first part annotations. The total workload of their method consists of the first part manual annotation plus the manual corrections required for the second part. Meanwhile, they evaluated the effect of different division ratios on the total annotation time and found that the optimal division ratio is 4-8%. However, for small datasets, inaccurate annotation proposals will be generated by a poor-robustness model trained with only 4-8% annotated images and it is actually inconvenient to modify annotation proposals, if a *graphical user interface* is missing which presents the unlabeled image and bounding boxes annotation proposals. In addition, instead of open-source datasets, they only validated their methods on an *indoor scene* dataset they collected.

Active learning, a type of iterative supervised learning, is effective in decreasing the amount of annotation required to achieve target model performance via annotating images selectively (Settles, 2009). Initially, in active learning, a baseline model is trained with a small labeled dataset and then applied to unlabeled images. Active learning selection metrics are utilized to calculate each unlabeled image’s incremental information score which estimates whether this image contains critical information that has not been learned by the baseline model. Once an unlabeled image with a higher incremental information score is annotated, it improves the better performance of the model than others. These images with a high incremental information score will be prioritized to annotate. Whenever a fixed amount of images is selectively annotated, the model is trained repeatedly to update the incremental information score of each image for the next iteration. Compared with selecting randomly, multiple metrics for object detection have been proved to achieve better performance of object detector with less annotation (Roy et al., 2019; Brust et al., 2019; Kao et al., 2019; Aghdam et al., 2019).

In this paper, we utilize active learning for bounding box annotation and propose a faster human-machine collaboration bounding box annotation framework based on active learning. For easy-to-operate annotation proposal modification, we design a *graphical user interface* (GUI), which presents annotation proposals for each unlabeled image. We evaluate our framework on *CityPersons* dataset, where our method outperforms manual annotation and the methods proposed by Adhikari et al. (2018). We also evaluated which selection metric in active learning can better decrease the annotation workload.
The rest of this paper is organized as follows. Section 2 introduces some prior knowledge related to this paper. Section 3 introduces our graphical user interface (GUI). Section 4 introduces how annotation workload and selection metrics are defined, and how the human-machine collaboration annotation framework based on active learning is constructed. Section 5 provides experimental results and analysis. Finally, Section 6 concludes the contributions of this paper.

2. Related Work

2.1. Object detection

Currently, state-of-the-art object detectors are divided into two categories, two-stage object detector represented by R-CNN series (Ren et al., 2017) and single-stage object detector represented by the YOLO series (Redmon & Farhadi, 2018). Given a picture, the two-stage object detector will generate its region proposals and then finetune proposals based on coordinates regression; while the single-stage object detector does not have intermediate region proposals, and the object location and class are obtained directly from the pictures. Compared with the single-stage object detector, the accuracy of the two-stage object detector is higher, but the delay is also higher.

2.2. Active learning

Active learning has been proved to be effective in image classification, but it has quite a little progress in object detection. Existing research can be divided into black-box and white-box methods.

- **Black-box methods** use the confidence scores of the bounding box which network outputs for selecting images without depending on the network architecture. These methods rely on the intuition that valuable samples are often uncertain (Settles, 2009). Roy et al. (2019) proposed Minmax, Maximum Entropy and Sum Entropy etc. black-box methods. Brust et al. (2019) computed the marginal score, based on the confidence scores from the SoftMax layer, to query image. Irshad et al. (2018) divided the bounding boxes into three buckets according to confidence score, which are high confidence bucket, low confidence bucket and no prediction bucket, and selected images of different scales from the three buckets.

- **White-box methods** have knowledge of the network architecture. Some of white-box methods design a special network architecture to output incremental information scores, and some extract the feature map in the middle of the network to calculate incremental information scores to select images. Kao et al. (2019) proposed the localization tightness and stability. The former measures how tight detected bounding boxes are, and the later estimates how stable they are in the original image and a noisy version of it. Aghdam et al.
(2019) proposed pixel-level scores which represent the importance of each pixel for improving the detector. They designed a detection network to output pixel-level scores and aggregated them to obtain an image-level score which is the basis to select images.

3. Graphical User Interface

As shown in Figure 1, we learn from Adhikari et al. (2018) and design a graphical user interface (GUI). The GUI presents each image’s annotation proposals predicted by pre-trained models, as shown in Figure 1(a). If annotation proposals are accurate enough, there is no need for manual modification. Even if annotation proposals are not accurate enough, the manual correction is also very simple (only need to drag), as shown in Figure 1(b) and Figure 1(c). The time for correcting is much shorter than manual annotation. This GUI can easily embed various open-source object detectors, and it can also easily add the novel active learning selection metrics.

4. Proposed Method

4.1. Annotation workload

We utilize human-machine collaboration annotation method to decrease manual annotation time. However, the evaluation metrics based on object detection, such as Recall and Precise, cannot intuitively evaluate our work. Therefore, we propose a new evaluation metric named annotation workload to evaluate how much annotation time cost with our method.

Papadopoulos et al. (2017a) and Su et al. (2012) reported the median annotation times on ImageNet: 25.5 seconds for drawing one box, 9 seconds for verifying its quality and 7.8 seconds for checking whether there are other objects of the same class yet to be annotated. Aghdam et al. (2019) reported 35 seconds for labeling pedestrians of a typical urban road scene on average when using six different labeling tools. Combining their experience and the GUI we designed, we believe that the time to annotate an image can be divided into the following three parts, drawing for a bounding box, verifying for a bounding box and checking for a whole image.

4.1.1. Drawing and verifying for a bounding box

If there is no annotation proposal for annotating a image on the GUI, annotating a bounding box is very time-consuming. According to Papadopoulos et al. (2017a), Su et al. (2012) and our experience, we consider 30 seconds for annotating one bounding box manually including 5 seconds for searching the box, 20 seconds for drawing the box and 5 seconds for verifying the box. If annotation proposals appear on the GUI, the annotation time of a box will be saved. We judge in the following situations:

- **Perfect annotation proposal**: These annotation proposals surround the object well. If there is ground truth, the intersection-over-union (IoU) between them will be bigger than 0.9. We think these annotation proposals do not need to be modified, it only takes 5 seconds to verify its quality.
- **Bad annotation proposal**: The location of these annotation proposals is much different from the object (IoU<0.5). We think these annotation proposals need to annotate manually again but we can save searching time because annotation proposal shows the approximate location of object. It takes 20 seconds for drawing the box and 5 seconds for verifying the box.
- **Common annotation proposal**: The location of these annotation proposals is not much different from the object (0.5<IoU<0.9). We can modify these annotation proposals easily by dragging the vertices of bounding boxes. According to our experience, we consider that the more accurate the annotation proposal, the less time required for manual modification. We assume that it satisfies the linear equation according to the above two boundary information (5 seconds for IoU=0.9 and 25 seconds for IoU=0.5), as shown in equation 1.

\[
T_{\text{annotation}(s)} = -50 \times \text{IoU} + 50 \quad (1)
\]

Above of all, \(T_{\text{annotation}(s)}\) as shown in equation 2.

\[
T(s) = \begin{cases} 
30 & \text{No annotation proposal} \\
5 & \text{IoU} > 0.9 \\
-50 \times \text{IoU} + 50 & 0.5 \leq \text{IoU} < 0.9 \\
25 & 0.3 \leq \text{IoU} < 0.5 
\end{cases} \quad (2)
\]

4.1.2. Checking for a whole image

After annotating all objects, we need to check whether all objects of the image are annotated. According to Su et al. (2012) and our experience, we consider 7 seconds for checking the whole image. However, if the GUI presents annotation proposals for almost all objects, the checking will be much easier, and the checking time will be decreased accordingly. In other words, checking time is related to the missret of annotation proposal. The higher the missret, the more checking time required. We also assume that this is also a linear equation and 2 seconds for checking at least. As shown in equation 3.

\[
T_{\text{checking}(s)} = 2 + 5 \times \text{missret} \quad (3)
\]
4.1.3. Annotation workload for one image
We suppose one image has \( K \) objects to annotate, as shown in equation 4, and we hope the total annotation workload is minimal:

\[
T_{\text{annotation workload}}(s) = K \cdot T_{\text{annotation}}(s) + T_{\text{checking}}(s)
\] (4)

4.2. Selection metrics based on active learning

4.2.1. Incremental information score
We use the black-box methods proposed by Roy et al. (2019) and Irshad et al. (2018) as the active learning selection metrics. Let us assume that, for each image, we have \( \{a_c\} \) annotation proposals corresponding to each class \( c \). We calculate incremental information score for each image with their annotation proposals’ confidence score \( S_i \) and rank them to select images. We assume that the higher incremental information score, more information the image has.

- Maximum confidence score (conf-max) (Roy et al., 2019):
  \[
  1 - \max_c \max_{i \in \{a_c\}} S_i
  \] (5)

- Sum entropy (ent-sum) (Roy et al., 2019): For each image, we calculate the sum of entropies of the annotation proposals with the same class as the entropy of each class in this image. We aggregate entropies of each class as incremental information score.
  \[
  \sum_c \sum_{i \in \{a_c\}} -S_i \log S_i
  \] (6)

4.2.2. Selection metrics
Once we calculate incremental information scores according to one of the above methods, we rank these incremental information scores and select a fixed \( N \) number of images for manual modification according to some metrics.

- Top \( N \) pictures: images with more image information (incremental information score) are generally annotated in active learning because active learning believes that images with more image information need to be learned by object detector.

- Bottom \( N \) pictures: more image information means more annotation workload, so we select images with less information.

- Multi-scale \( N \) pictures: For comprehensiveness, we also consider selecting from the top and bottom separately according to a different scale. We refer to Irshad
et al. (2018) to divide all images into three buckets based on their incremental information scores: No prediction bucket, low incremental information score bucket (account for 80% of images with prediction) and high incremental information score bucket (account for 20% of images with prediction). We pick 20%N at the bottom of high incremental information score bucket, 30%N at the top of low incremental information score bucket, 30%N at random in the rest of low incremental information score bucket, and 20%N at random in no prediction bucket.

4.3. Framework

Our human-machine collaboration annotation framework based on active learning is shown in Figure 2. Initially, we assume that a labeled dataset called $X_{labeled}$, which takes $W_{init}$ to annotate, is used to train an object detector. $W_{init}$ was added to annotation workload pool for labeled images. Active learning will apply the object detector to a set of unlabeled images called $X_{unlabeled}$ and get their annotation workload called $W_{unlabeled}$. The total annotation workload is annotation workload pool for labeled images and annotation workload for unlabeled images. Then we select $N$ images called $X_{selected}$ to annotate and add their annotation workload called $W_{selected}$ to annotation workload pool for labeled images. We call this process one iteration. When unlabeled images are exhausted, the iteration ends. So, the total annotation workload for the $i$-th iteration is shown in equation 7:

$$W_{i-th} = W_{init} + \sum_{i=0}^{i-1} W_{i-th selected} + W_{i-th unlabeled}$$  (7)

5. Experiment

5.1. Dataset

We choose CityPersons (Zhang et al., 2017) dataset, which is a single category (person) dataset, to experiment. CityPersons dataset has more small-sized objects, more occlusion and diverse environments compared with the indoor scene dataset Adhikari et al. (2018) utilized. We divide the original image into two images of size 1024 * 1024. We select 3500 pictures of them as unlabeled images to evaluate annotation workload decrease and 700 pictures as test set to evaluate model performance.

5.2. Implementation details

We train object detector with MMDetection (Chen et al., 2019), which is an open-source object detection toolbox based on PyTorch. We choose Faster R-CNN as the architecture of object detector and X-101-64x4d-FPN as its backbone. Other training settings use the default configuration of MMDetection. Initially, we randomly select 500 pictures to annotate, which is consistent with the semi-automatic bounding box image annotation method proposed by Adhikari et al. (2018). After that, in each iteration, we select different 250 pictures to annotate according to different selection metrics illustrated in section 4.2. Our method based on active learning will be proved better than the above method proposed by Adhikari et al. (2018) and we also evaluate which selection metric is better according to the extent of annotation workload decrease.

5.3. Model performance each iteration

We evaluate the model performance $mAP$ ($IoU=0.5:0.95$) (left), $mAP$ ($IoU=0.5$) (middle), $mAP$ ($IoU=0.75$) (right) on test set at each iteration, as shown in Figure 3. Since we use the highest $mAP$ ($IoU=0.5:0.95$) of models in each iteration, the rising curve in Figure 3(a) is more gentle than others. Although the rising curves in Figure 3(b) and Figure 3(c) are
slightly fluctuating, they still have a considerable distinction.
Each curve in each subfigure starts from the same point
because the initial 500 pictures trained in different selection
metric are the same. After annotating different 250 pictures
in each iteration, each selection metric represented by each
curve changes differently.

Especially, Top-ent-sum rises rapidly, Top-conf-max and
Bottom-ent-sum rise slowly. Because the selection methods
of Top-ent-sum and Bottom-ent-sum are opposite, their
curves are the best and the worst, which can be reasonably
explained. This also shows that there are indeed some images
that can significantly affect the improvement of the model,
and some images have little effect on the improvement
of the model. Random performs better than some of the
metrics, which for the reason that the black-box method
especially uses uncertainty, and Random also has uncertainty.

Generally, different selection metrics show obvious differ-
ences in the early iteration and they tend to converge in the
later iteration, where data of training set is roughly the same.
This is consistent with active learning.

5.4. Annotation workload decrease

Table 1 and Figure 4 show that our method can decrease
the percentage of annotation workload compared with manual
annotation. When the number of labeled images is
the initial 500, this is the same as the method of Adhikari
et al. (2018). According to our experiment, the method of
Adhikari et al. (2018) can decrease annotation workload
by 44% compared with manual annotation. Considering
the difference of dataset's difficulty, annotation workload
formula and object detector, the experimental results also
show that the method of Adhikari et al. (2018) can greatly
decrease annotation workload compared with manual anno-
ation. As the number of labeled images increases, the rising
in the curve shows that our method based on active learning
will decrease more annotation workload than Adhikari et al.
(2018).

Table 2 and Figure 5 show that our method can decrease the
percentage of annotation workload compared with Adhikari
et al. (2018). Seven curves rise in varying extents, which
shows that active learning is effective for continuously de-
creasing annotation workload. At the same as the above
model performance, Top-ent-sum shows better performance
in annotation workload decrease than other selection met-
rics. This curve rises faster in the early iteration, and the
overall performance is smoother, indicating that this selec-
tion metric is more stable and robust than others. As shown
in Table 2, compared with Adhikari et al. (2018), using our
method to annotate images can decrease annotation work-
load by at least 4.6% and up to 6.2%. As shown in Table 1,
compared with manual annotation, using our method to an-
notate images can decrease annotation workload by at least
46.6% and up to 47.5%.

The experiment shows that the entropy can reflect the un-
certainty of the image better than the confidence, and using
entropy as an image's incremental information score is bet-
ter in black-box methods. The rapid rise in the first half of
curves shows that even if we just constantly annotate half of
the images with a framework based on active learning, we
can achieve a relatively high saving in practical applications.

---

**Table 1. Annotation workload decrease compared with manual annotation.**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Labeled pictures</th>
<th>500</th>
<th>750</th>
<th>1000</th>
<th>1250</th>
<th>1500</th>
<th>1750</th>
<th>2000</th>
<th>2250</th>
<th>2500</th>
<th>2750</th>
<th>3000</th>
<th>3250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td></td>
<td>44%</td>
<td>44.9%</td>
<td>45.5%</td>
<td>46.4%</td>
<td>46.7%</td>
<td>46.7%</td>
<td>47%</td>
<td>47%</td>
<td>47%</td>
<td>47%</td>
<td>47%</td>
<td>47%</td>
</tr>
<tr>
<td>Top-conf_max</td>
<td></td>
<td>44%</td>
<td>43.9%</td>
<td>45.3%</td>
<td>45.6%</td>
<td>45.8%</td>
<td>46.6%</td>
<td>46.6%</td>
<td>46.9%</td>
<td>46.9%</td>
<td>47%</td>
<td>47%</td>
<td>47%</td>
</tr>
<tr>
<td>Bottom-conf_max</td>
<td></td>
<td>44%</td>
<td>44.6%</td>
<td>45.8%</td>
<td>45.8%</td>
<td>46.1%</td>
<td>46.9%</td>
<td>46.7%</td>
<td>46.8%</td>
<td>47%</td>
<td>47%</td>
<td>47%</td>
<td>47%</td>
</tr>
<tr>
<td>Multi-scale_conf_max</td>
<td></td>
<td>44%</td>
<td>44.9%</td>
<td>45.5%</td>
<td>46.2%</td>
<td>46.5%</td>
<td>47%</td>
<td>47%</td>
<td>47%</td>
<td>47%</td>
<td>47%</td>
<td>47%</td>
<td>47%</td>
</tr>
<tr>
<td>Top-ent_sum</td>
<td></td>
<td>44%</td>
<td>46%</td>
<td>46.3%</td>
<td>46.5%</td>
<td>46.9%</td>
<td>47.1%</td>
<td>47.2%</td>
<td>47.3%</td>
<td>47%</td>
<td>47.5%</td>
<td>47%</td>
<td>47.5%</td>
</tr>
<tr>
<td>Bottom-ent_sum</td>
<td></td>
<td>44%</td>
<td>43.9%</td>
<td>44.5%</td>
<td>44.8%</td>
<td>45.1%</td>
<td>45.2%</td>
<td>45.6%</td>
<td>45.8%</td>
<td>46.2%</td>
<td>46.4%</td>
<td>46.5%</td>
<td>46.6%</td>
</tr>
<tr>
<td>Multi-scale_ent_sum</td>
<td></td>
<td>44%</td>
<td>45%</td>
<td>45.9%</td>
<td>46.1%</td>
<td>46.8%</td>
<td>46.8%</td>
<td>47%</td>
<td>47%</td>
<td>47%</td>
<td>47.3%</td>
<td>47.2%</td>
<td>47.2%</td>
</tr>
</tbody>
</table>

**Table 2. Annotation workload decrease compared with Adhikari et al. (2018).**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Labeled pictures</th>
<th>500</th>
<th>750</th>
<th>1000</th>
<th>1250</th>
<th>1500</th>
<th>1750</th>
<th>2000</th>
<th>2250</th>
<th>2500</th>
<th>2750</th>
<th>3000</th>
<th>3250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td></td>
<td>0</td>
<td>1.6%</td>
<td>2.7%</td>
<td>4.2%</td>
<td>4.8%</td>
<td>4.7%</td>
<td>5.4%</td>
<td>5.3%</td>
<td>5.6%</td>
<td>5.6%</td>
<td>5.6%</td>
<td>5.6%</td>
</tr>
<tr>
<td>Top-conf_max</td>
<td></td>
<td>0</td>
<td>-0.2%</td>
<td>2.3%</td>
<td>2.8%</td>
<td>3.2%</td>
<td>4.6%</td>
<td>4.6%</td>
<td>5.2%</td>
<td>5.2%</td>
<td>5.3%</td>
<td>5.5%</td>
<td>5.5%</td>
</tr>
<tr>
<td>Bottom-conf_max</td>
<td></td>
<td>0</td>
<td>1.1%</td>
<td>3.2%</td>
<td>3.3%</td>
<td>3.8%</td>
<td>5.2%</td>
<td>4.8%</td>
<td>4.9%</td>
<td>5.3%</td>
<td>5.3%</td>
<td>5.4%</td>
<td>5.4%</td>
</tr>
<tr>
<td>Multi-scale_conf_max</td>
<td></td>
<td>0</td>
<td>1.5%</td>
<td>2.7%</td>
<td>3.8%</td>
<td>4.5%</td>
<td>5.3%</td>
<td>5.3%</td>
<td>5.6%</td>
<td>5.6%</td>
<td>5.6%</td>
<td>5.7%</td>
<td>5.6%</td>
</tr>
<tr>
<td>Top-ent_sum</td>
<td></td>
<td>0</td>
<td>3.5%</td>
<td>4.1%</td>
<td>4.4%</td>
<td>5.2%</td>
<td>5.6%</td>
<td>5.6%</td>
<td>5.8%</td>
<td>6%</td>
<td>6.1%</td>
<td>6.2%</td>
<td>6.2%</td>
</tr>
<tr>
<td>Bottom-ent_sum</td>
<td></td>
<td>0</td>
<td>-0.2%</td>
<td>0.9%</td>
<td>1.3%</td>
<td>2%</td>
<td>2.2%</td>
<td>2.9%</td>
<td>3.2%</td>
<td>3.8%</td>
<td>4.3%</td>
<td>4.4%</td>
<td>4.6%</td>
</tr>
<tr>
<td>Multi-scale_ent_sum</td>
<td></td>
<td>0</td>
<td>1.7%</td>
<td>3.3%</td>
<td>3.8%</td>
<td>4.9%</td>
<td>4.9%</td>
<td>5.2%</td>
<td>5.5%</td>
<td>5.7%</td>
<td>5.8%</td>
<td>5.8%</td>
<td>5.8%</td>
</tr>
</tbody>
</table>
6. Conclusion

In this paper, we proposed a faster human-machine collaboration annotation framework based on active learning to decrease annotation workload. Compared with manual annotation, our experiment showed that our method can decrease annotation workload by 47.5% in CityPersons dataset. In addition, our method decreases annotation workload by 6.2% compared with the method of Adhikari et al. (2018).

For future work, we will explore more possibilities of active learning on bounding box annotation. We will learn from more active learning methods, and experiment them on more types of datasets.

Figure 4. Annotation workload decrease compared with manual annotation.

Figure 5. Annotation workload decrease compared with Adhikari et al. (2018).

References


Faster Human-Machine Collaboration Bounding Box Annotation Framework Based on Active Learning


