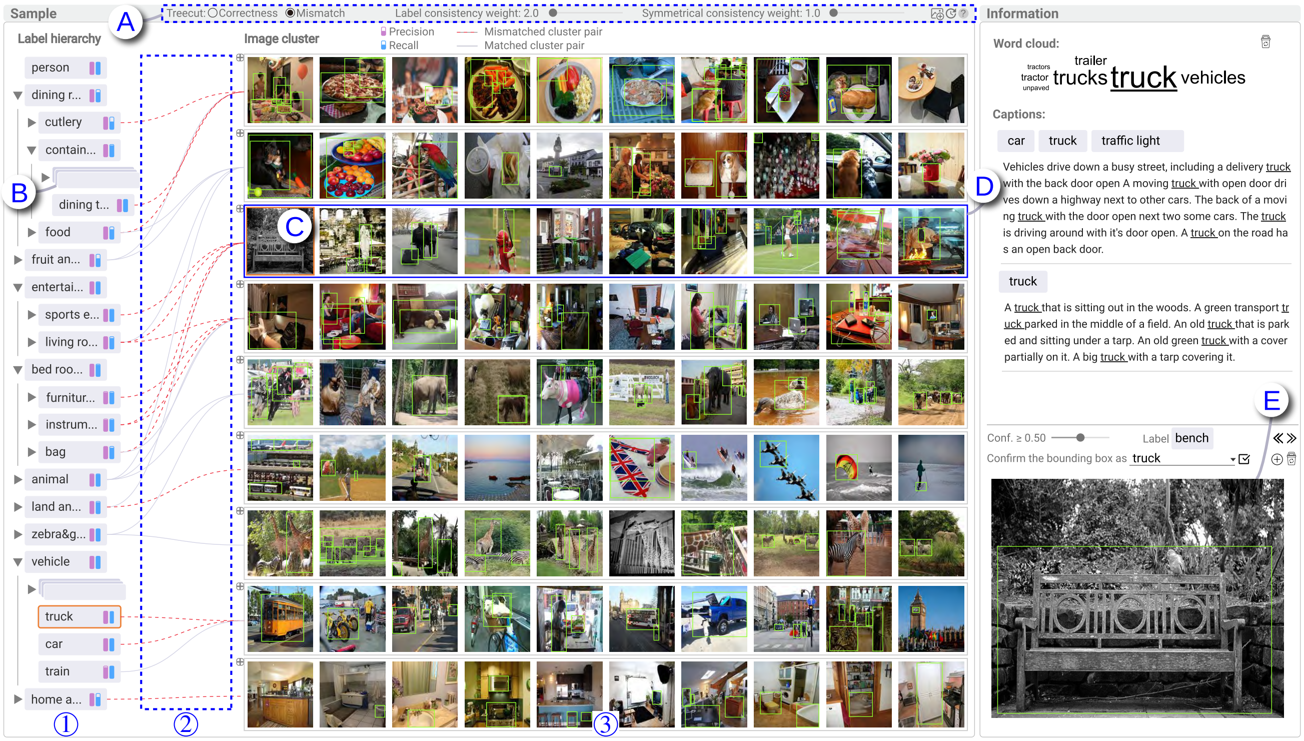
Towards Better Caption Supervision for Object Detection

Changjian Chen, Jing Wu, Xiaohan Wang, Shouxing Xiang, Song-Hai Zhang, Qifeng Tang, Shixia Liu



Treecut: ○Correctness ●Mismatch

Label consistency weight: 2.0

Symmetrical consistency weight: 1.0

Word cloud:

Captions:

car truck traffic light

Vehicles drive down a busy street, including a delivery truck with the back door open A moving truck with open door drives down a highway next to other cars. The back of a moving truck with the door open next two some cars. The truck is driving around with it's door open. A truck on the road has an open back door.

truck

A truck that is sitting out in the woods. A green transport truck parked in the middle of a field. An old truck that is parked and sitting under a tarp. An old green truck with a cover partially on it. A big truck with a tarp covering it.

Conf. ≥ 0.50

Label bench

Confirm the bounding box as truck

(a) (b)

Fig. 1. MutualDetector: (a) a node-link-based set visualization consists of a tree of labels (①), the relationships between the labels and image clusters (②), and a matrix (③) to show the representative images with the detected objects for each cluster; (b) an information panel to show important words, captions, and selected images.

Abstract—As training high-performance object detectors requires expensive bounding box annotations, recent methods resort to free-available image captions. However, detectors trained on caption supervision perform poorly because captions are usually noisy and cannot provide precise location information. To tackle this issue, we present a visual analysis method, which tightly integrates caption supervision with object detection to mutually enhance each other. In particular, object labels are first extracted from captions, which are utilized to train the detectors. Then, the label information from images is fed into caption supervision for further improvement. To effectively loop users into the object detection process, a node-link-based set visualization supported by a multi-type relational co-clustering algorithm is developed to explain the relationships between the extracted labels and the images with detected objects. The co-clustering algorithm clusters labels and images simultaneously by utilizing both their representations and their relationships. Quantitative evaluations and a case study are conducted to demonstrate the efficiency and effectiveness of the developed method in improving the performance of object detectors.

Index Terms—Machine learning, interactive visualization, object detection, caption supervision, co-clustering.

1 Introduction

Object detection is to localize and classify objects from images. It is a fundamental problem in computer vision [1], [2] and is widely used in many real-world applications [3], such as self-driving cars, augmented reality, and object tracking. With the rapid development of deep neural networks, there has been significant progress in the accuracy and efficiency of object detection methods [3]. However, the training process requires a large number of annotations, including the bounding box annotations and labels of all the objects in images [3], which are expensive to acquire. As a result, researchers turn to other cheaper or free annotations [4], [5]. For

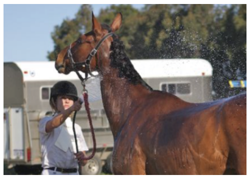
- *C. Chen, S. Xiang, S.-H. Zhang, and S. Liu are with Tsinghua University.*
- *J. Wu is with Cardiff University.*
- *X. Wang is with Zhejiang University.*
- *Q. Tang is with Shanghai Lianshu IoT Co. Ltd.*

example, web users often provide descriptions (*i.e.*, alt-texts or captions) about the images that they upload to social media. These image captions provide hints about the objects in these images and thus can be utilized for training [5].

There have been some methods to utilize image captions for training object detectors (*e.g.*, Cap2Det [5]). With such methods, the training is carried out in two steps. First, a natural language processing (NLP) model is trained to extract object labels from the captions. These labels are then used as supervision to train a detector. Although these methods can leverage image captions to train detectors, the performance is not satisfactory due to two reasons. First, the captions are usually non-exhaustive. They usually only describe the content of interest but ignore other content in the images. For example, the caption in Fig. 2 only focuses on the horse, without mentioning the people and the van behind. As a result, such non-exhaustive captions do not provide enough supervision for detecting all the objects in the images. Second, captions are image-level descriptions and do not provide object-level locations. It is usually hard to obtain high-performance detectors without object-level location annotations [6].

To improve the performance of detectors with caption supervision, an intuitive way is to introduce a small number of images with bounding box annotations into the training data. Such a small number of annotated images are more labor-efficient to collect compared to annotating all images. Moreover, humans can be involved in the analysis process to provide validation. For example, humans can provide missing labels or validate uncertain bounding boxes. The validation can then be added to the training data for further improving the detectors. However, there are still challenges. The first challenge is how to combine the small number of annotated images with captions to build an effective object detector. To the best of our knowledge, this has not been investigated by existing methods. Second, humans have to explore all the extracted labels, images with detected objects, and their relationships to identify important data for validation, especially those that can bring a relatively large gain. This process is time-consuming and labor-intensive. Thus, a tool to facilitate efficient exploration is needed. Third, existing methods only leverage the extracted labels to improve object detection, but not the other way around. Generally, label extraction and object detection are not independent but mutually influence each other. With such mutual influence, if one of them is improved via user validation, the other can also be improved. Thus, it is necessary to study how this mutual enhancement works.

In this paper, we develop MutualDetector, a visual analysis tool, to help machine learning experts and practitioners 1) explore extracted labels, images with detected objects, and their relationships; 2) provide missing labels and validate uncertain bounding boxes to improve the model performance. With MutualDetector, the users do not need to have much knowledge on the underlying model. Specifically, our tool starts from images with captions. A small part (*e.g.*, 5%) of these images are annotated with bounding boxes to balance the labeling cost and model performance. To effectively utilize both the captions and bounding box annotations, we develop a semi-supervised object detection method to train a label extractor and an object detector. The labels extracted from captions and the objects detected from images are then visualized to facilitate the exploration and validation. Since one label may appear in multiple images and one image may contain multiple labels, the relationships between the labels and images with detected objects can be modeled as a many-to-many set relationship. We thus develop a node-link-based set visualization as the core of MutualDetector. Since the numbers of labels and images are large, to address the scalability issue, we cluster the labels and images simultaneously, which is formulated as a multi-type relational co-clustering problem [7]. With the help of interactive visualization, the users can explore the relationships between the extracted labels and images with detected objects, and then provide validation for them. The validation can be utilized by the developed semi-supervised object detection method to improve the label extractor and object detector and enhance each other. The developed object detection method and the interactive visualization are tightly integrated to support a human-in-the-loop validation pipeline for improving object detection. Experiments are conducted to quantitatively evaluate the developed semi-supervised object detection method. A case study on the COCO17 dataset [8] shows the effectiveness of MutualDetector. The demo is available at: http://mutual-detector.thuvis.org/.



“One horse getting a well earned cool downafter the hot conditions on the first day of the Swan River Horse Trials.”

Fig. 2. The associated caption describes the horse but ignores the people and van in the image. Such a caption does not provide enough supervision for training an object detector.

In summary, the main contributions of this work are:

- **A semi-supervised object detection method** that utilizes both the captions and a small number of bounding box annotations to improve the detection performance.
- **A node-link-based set visualization** supported by a multi-type relational co-clustering algorithm to explain the relationships between extracted labels and images with detected objects.
- **A visual analysis tool** that tightly integrates the object detection method and interactive visualization to facilitate the exploration and validation of labels and object bounding boxes. This tool loops humans into the object detection process to improve the detection performance.

2 Related Work

Our work relates to semi-supervised object detection and visualization work for annotation quality improvement. This section reviews the related work and contrasts our contributions.

2.1 Semi-Supervised Object Detection

There are two types of semi-supervised object detection methods: self-training methods and consistency regularization methods. Self-training methods first train a detector with the annotated images, which is then used to detect objects in all images. Some detected objects are treated as annotations to retrain the detector. The key for this type of method is how to select detected objects for training. Kumar *et al.* [9] used a simple strategy that detected objects with high confidence are selected as annotations. Wang *et al.* [10] selected the detected objects that could also be detected when being patched into other images. Although self-training methods are effective, they need to be repeated many times, which is time-consuming. Unlike self-training methods, consistency regularization methods add a robustness constraint between each unlabeled image and its perturbed version, which only needs to train the detector once. For example, Jeong *et al.* [11] developed CSD-SSD based on the widely used annotation-based object detection method, SSD [12]. It ensures that the detector is robust to given perturbed inputs.

The aforementioned semi-supervised methods ignore the captions of images, which can further improve the performance. Some recent work has been proposed to train detectors with captions. For example, Cap2Det [5] trains the object detector by utilizing the labels extracted by an NLP model. Cap2Det sufficiently utilizes captions to train detectors. However, it does not utilize object bounding box annotations to improve the detection performance. Compared with it, our method tightly integrates both captions and a small number of bounding box annotations for better performance.

2.2 Visualization for Annotation Quality Improvement

Based on whether there are noisy annotations or insufficient annotations, existing visualization work for annotation quality improvement can be classified into two categories: improving the quality of noisy annotations and interactive labeling [13].

Improving the quality of noisy annotations. To improve the quality of crowdsourced labels, Park *et al.* [14] developed a visual analysis platform, C^2A, to help detect anomalies and build a consensus on crowdsourced labels. LabelInspect [15] allows experts to interactively verify uncertain labels and unreliable workers in an iterative and progressive procedure. In practice, some datasets, such as ImageNet [16], do not contain crowd information. To handle label noise in such datasets, Xiang *et al.* [17] tightly integrated a scalable trusted-item-based correction algorithm with an incremental t-SNE algorithm to support an iterative refinement procedure. Bäuerle *et al.* [18] proposed three error detection measures, class interpretation error score, instance interpretation error score, and similarity error score, and leveraged them to correct label errors.

Interactive labeling. Moehrmann *et al.* [19] used a self-organizing map to place similar images close to each other and facilitate users in labeling multiple images at the same time. Such a similarity-based strategy is also employed for labeling social spambot groups [20] and errors in electrical engines [21]. In addition to the similarity-based strategy, filtering and sorting are also used to facilitate the labeling process [22], [23]. Moreover, there are some efforts that integrate interactive visualization with active learning methods. Höferlin *et al.* [24] introduced the concept of "intra-active learning." Users can label instances recommended by an active learning algorithm or select informative instances to label with the help of visualization, which are used to further improve the underlying model. Such an integration is also substantiated by other work [25], [26], [27], [28], [29], [30], [31].

Different from the above methods that focus on classification, our work focuses on object detection. A recent work, VATLD [32], combines disentangled representation learning and semantic adversarial learning to help understand the object detection method and resolve data quality issues, such as mislabeled data. However, VATLD requires the bounding box annotations for each training data, which is not applicable for an image set with a small number of bounding box annotations. To improve the performance of object detectors trained on images and their captions, the relationships between labels extracted from captions and images with detected objects need to be analyzed. To this end, we developed a node-link-based set visualization supported by a multi-type relational co-clustering algorithm. It allows users to analyze the extracted labels, images with detected objects, and their relationships. Based on the analysis, they provide validation to improve model performance.

3 Design of MutualDetector

This section presents the requirements and system overview of MutualDetector.

3.1 Requirement Analysis

The development of MutualDetector was in collaboration with two groups of machine learning experts who are not co-authors of this work. The first group consists of two Ph.D. students (E_1 and E_2) who have won the first place in the CVPR 2020 EPIC-Kitchens Action Recognition Competition [33]. Object detection is a key component in this competition. The videos used in the competition have many frames with subtitles, and thus contain a large number of images with captions. The second group includes a professor (E_3), a post-doctor (E_4), and two Ph.D. students. They have collaborated with two hospitals and developed models for disease detection from computed tomography (CT) images. Each CT image is associated with a textual diagnosis report. Both groups used Cap2Det [5], a state-of-the-art method for training object detectors based on image captions. However, the performance was not satisfactory. There were incorrect labels extracted from the captions and imprecise detected objects. They would like to provide validation to improve the performance. However, due to the large number of labels and objects, the manual validation is tedious and time-consuming. Therefore, the experts desired a tool to efficiently validate the extracted labels and detected objects.

To identify the requirements for the tool, we conducted four semi-structured interviews with the experts from the two groups. Based on the interviews, we derived the following requirements.

R1 - Combine captions and annotations to train an object detector. In many object detection applications, images have captions, and a small number of them have bounding box annotations. The experts agreed that effectively integrating the annotations and captions provided more supervision for learning and thus could boost the detection performance. For example, E_3 said that they had asked doctors to annotate some CT images. But existing methods only utilize either annotations or captions, not the combination, which limits the detection performance. Thus, a seamless combination of them is required to build an effective object detector.

R2 - Understand the matching relationships between the extracted labels and images with detected objects. As captions are descriptions of the images, the labels extracted from captions and detected from images should be consistent [5]. However, due to the noisy nature of captions and unsatisfactory detection results, mismatches often exist. This is where user validation is required. However, when the dataset is large, exploring the mismatches is challenging. Thus, the experts expressed the need for an effective tool to understand such matching.

R3 - Examine the extracted labels in the context of captions. All the experts agreed that the non-exhaustive nature of captions is the main reason for the unsatisfactory extraction of labels, such as the failed extraction of label "van" and "person" in the example in Fig. 2. To effectively improve the extracted labels, they need to identify which labels are missing or incorrect in the context

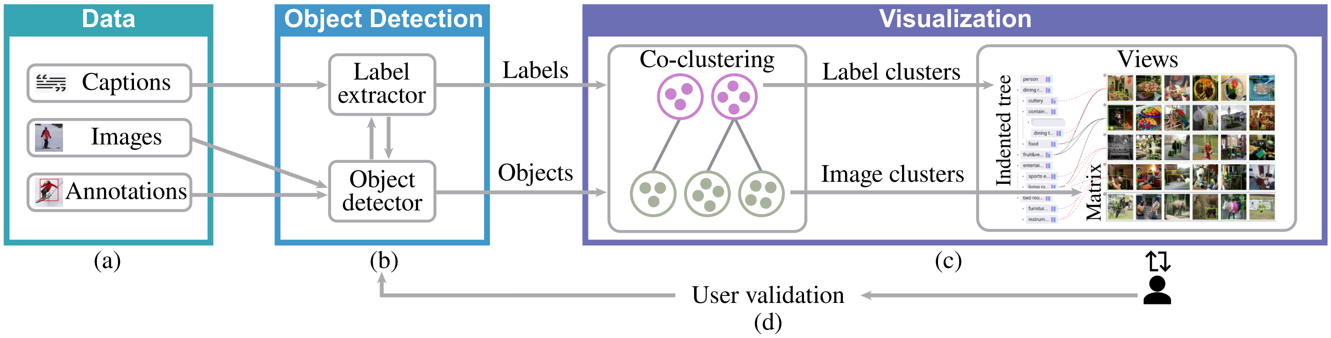


Fig. 3. MutualDetector pipeline: (a) starting from images with captions and a small number of bounding box annotations; (b) training a label extractor and an object detector; (c) simultaneously clustering extracted labels and images, and analyzing extracted labels, images with detected objects, and their relationships; (d) validating the results to improve the performance of both the label extractor and object detector.

of captions. E_1 commented, "I want to know which words are important for the extraction of a certain label. If the important words do not make sense, then I can directly remove them from the captions and retrain the label extractor. "

R4 - Explore the images and their detected objects at different levels of detail. All the experts expressed the desire to quickly explore the overall distribution of images for identifying the images with poor detection results, such as images whose detected objects are mismatched with the extracted labels. Then they wanted to examine the regions with these images at different levels of detail to understand the poor performance. For example, E_2 said, "I hope I can check only a few representative images to have an overview of the performance first and then zoom from the overview to the regions of interest at different granularities."

R5 - Mutually improve the performance of the label extractor and object detector. In current practice, the extracted labels are used as supervision to train the object detector, but are not affected by the object detector. The relationships between the label extractor and the object detector are bidirectional. On the one hand, the labels extracted by the label extractor can serve as supervision for training the object detector. On the other hand, the objects detected by the object detector can compensate for the deficiency of non-exhaustive captions and improve the label extractor. The mutual influence between the label extractor and object detector can boost their performance more effectively. E_3 commented, "Such a mutual influence is very useful to achieve better model performance, especially when a little amount of user validation is provided. For example, when some detected objects are validated to improve the object detector, the label extractor can also be improved. The improved label extractor further boosts the performance of the object detector (and vice versa)."

3.2 System Overview

Guided by these requirements, we developed MutualDetector to interactively improve the performance of object detection based on the combination of captions and annotations (**R1**). As shown in Fig. 3, MutualDetector contains two main modules: **object detection** (Sec. 4) and **visualization** (Sec. 5).

The object detection module (Fig. 3(b)) takes as input images with captions and a small number of bounding box annotations and trains a label extractor and an object detector. The extractor is used to extract labels and their representations, and the detector is employed to detect objects and learn image representations. In the visualization module (Fig. 3(c)), to facilitate the exploration of these results, a multi-type relational co-clustering algorithm [7] is applied to cluster labels and images simultaneously. The clustering results are then fed into the tree view and matrix view for exploring the extracted labels (**R3**), images with detected objects (**R4**), and the matching relationships between them (**R2**). During exploration, users can validate the extracted labels and detected objects. The validation is then used to mutually improve the label extractor and object detector (**R5**).

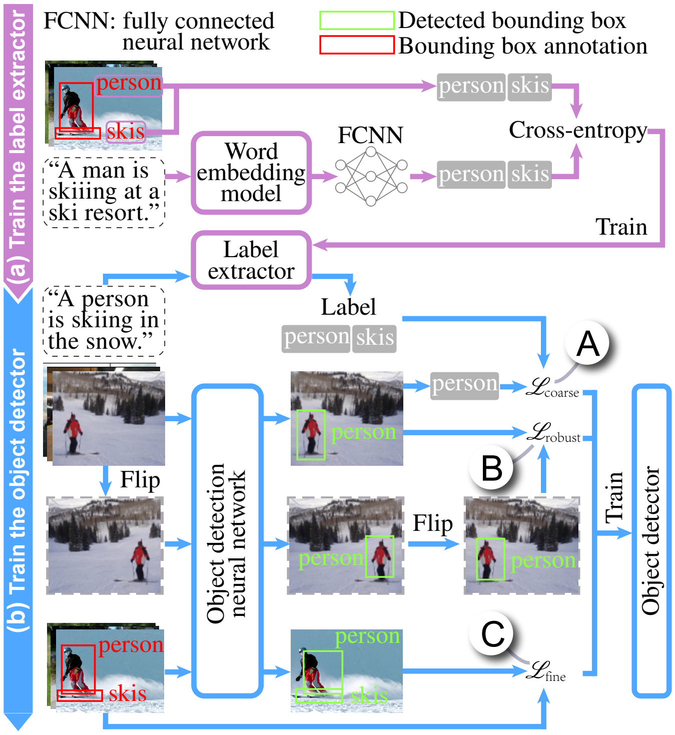

FCNN: fully connected neural network
Detected bounding box
Bounding box annotation
(a) Train the label extractor
person
skis
person skis
"A man is skiiing at a ski resort."
Word embedding model
FCNN
person skis
Cross-entropy
Train
Label extractor
(b) Train the object detector
"A person is skiing in the snow."
Label
person skis
A
person
$\mathcal{L}_{\text{coarse}}$
person
$\mathcal{L}_{\text{robust}}$
B
Flip
Object detection neural network
person
Flip
person
Train
Object detector
C
person
skis
person
skis
$\mathcal{L}_{\text{fine}}$

Fig. 4. The two steps in the developed semi-supervised object detection method: (a) training the label extractor using the labels and captions of the annotated images (**purple**); (b) training an object detector using the labels extracted from captions and the bounding box annotations from images (**blue**).

4 Semi-Supervised Object Detection

The major goal of the developed semi-supervised object detection method is to train an object detector utilizing image captions and a small number of bounding box annotations (**R1**). For annotated images, the detected objects should be consistent with their bounding box annotations, including both locations and labels, while for images with captions, labels of the detected objects should be consistent with those extracted from their captions. However, the object detector easily overfits to the noise in captions if trained with these two constraints only. The previous study [34] has shown that the robustness constraint is effective in reducing such overfitting by ensuring that the detector is robust to given perturbed inputs. Thus, we add this constraint to reduce overfitting. To satisfy these three constraints, as shown in Fig. 4, our method consists of two steps: 1) training the label extractor using the labels and captions of the annotated images (**purple**), and 2) training the object detector using both the labels extracted from captions and the bounding boxes annotated in images (**blue**).

Label extractor. Similar to Cap2Det [5], the label extractor is obtained by fine-tuning a word embedding model, GloVe [35], followed by two fully connected layers. Other NLP models, such as BERT [36], can also be employed directly. The captions of the small number of annotated images and the labels of their annotated bounding boxes are used for fine-tuning. As only the parameters of the fully connected layers will be trained, such an amount of data is enough for training. After that, for images with captions, the trained extractor is applied to extract labels from their captions.

Object detector. The developed object detector is based on SSD [12], which works well to satisfy the bounding box consistency constraint of the annotated images. Given the extracted labels and the small number of annotated images, our method extends the loss function of SSD with a second term satisfying the label consistency constraint and a third term satisfying the robustness constraint.

$$\mathscr{L}_{\text{fine}} + \alpha_1 \mathscr{L}_{\text{coarse}} + \alpha_2 \mathscr{L}_{\text{robust}}. \quad (1)$$

The **first term** $\mathscr{L}_{\text{fine}}$ is the MultiBox loss of SSD, which ensures the detected objects of the annotated images to be consistent with the bounding box annotations (Fig. 4C), including both locations and labels. If another object detection neural network, such as YOLO [1] and Faster-RCNN [2], is employed, its loss function can be directly applied here.

The **second term** $\mathscr{L}_{\text{coarse}}$ is the label consistency constraint applied to images with captions. It enforces that the detected objects in an image must be consistent with the labels extracted from its captions (Fig. 4A). Here the widely used cross-entropy loss is employed to measure their difference.

The **third term** $\mathscr{L}_{\text{robust}}$ is the robustness constraint to ensure the detector to be robust to given perturbed inputs. It can well reduce overfitting to noisy captions. As a result, we apply this constraint to the images with captions. The idea is that if one image is perturbed, the detected objects in the perturbed image and in the original image should be consistent, while the detected object locations in them should be corresponding to the perturbation. For example, the detected object locations in a horizontally flipped image and the original image should be symmetrical in the horizontal direction (Fig. 4B). The original robustness constraint proposed in [11] is applied to all detected objects in one image, which includes falsely detected ones whose labels are not described in its caption. Such falsely detected objects can be the majority of detection and dominate the calculation of the third term, especially when the detector performance is poor at the beginning of training. Therefore, we modify the original robustness constraint by only applying this constraint to the detected objects whose labels are extracted from the captions. In our implementation, the widely used horizontal flip is employed to perturb the images, which illustrates the basic idea. Other perturbations, such as rotations, can also be included to improve the performance [37]. Following [11], Jensen-Shannon divergence and L_2 loss are respectively used to measure the difference of labels and the difference of locations.

α_1 and α_2 are two weights to balance the three terms. In our implementation, α_1 is set according to the sizes of datasets. We set α_1 to be 1 for datasets with less than 50,000 images and 2 for datasets with more than 50,000 images. α_2 is set to be 1, following the same setting in [11]. The detailed formulations of the three terms can be found in supplemental material.

5 MutualDetector Visualization

The output results of the object detection method, including labels, images with detected objects, and their relationships, can be modeled as a many-to-many set relationship. To better understand such results, we first utilize the multi-type relational co-clustering algorithm for efficiently handling the large numbers of labels and images. Then with the clustering result, a set visualization is developed to illustrate the clustering results and facilitate the analysis and validation. Finally, an interactive improvement process based on user validation is introduced.

5.1 Multi-type Relational Co-clustering

To facilitate the exploration of the large numbers of labels and images, we hierarchically cluster them such that 1) similar labels (images) are clustered together; 2) labels that are matched with similar images are clustered together and vice versa. This can be formulated as *multi-type relational co-clustering* [7], which simultaneously clusters the labels and images by taking into account both their representations and relationships. Such clustering results enable users to efficiently identify label and image cluster pairs with many label-image mismatches.

Algorithm overview. Multi-type relational co-clustering utilizes both representation and relationship information for simultaneously clustering labels and images. It ensures the compactness of label and image clusters in their representation space and allows their clustering results to influence each other through their relationships, which is achieved by minimizing the following cost function:

$$||M - D^C S(D^I)^T||^2 + \beta_1 ||X^C - D^C F^C||^2 + \beta_2 ||X^I - D^I F^I||^2. \quad (2)$$

The first term ensures matched labels and images to be in the same co-clusters. The second and third terms ensure that labels (images) are close to their cluster centers. Here, X^C and X^I are the representations of labels and images, respectively. Each row of X^C (X^I) represents a representation vector of a label (image). Label representations are extracted using a word embedding method, GloVe [35], fine-tuned on the captions [38], and image representations are extracted using the detector [39], [40], [41] described in Sec. 4. M is the relationship matrix between labels and images. $M_{ij} = 1$ indicates that the i-th label is detected from the j-th image and extracted from its caption as well; otherwise $M_{ij} = 0$. D^C (D^I) denotes the clustering result of labels (images). $D^C_{ij} = 1$ ($D^I_{ij} = 1$) indicates that the i-th label (image) belongs to the j-th cluster. F^C (F^I) denotes the cluster centers of labels (images) in the representation space. S is the cluster association matrix, where S_{pq} is the mean of the co-cluster between the p-th label cluster and q-th image cluster. β_1 and β_2 are two weights to balance the three terms. In our tool, we set both of them to be 1.

To facilitate the exploration of large image sets with many labels, we repeatedly apply the above co-clustering to build the label and image hierarchies in a top-down manner. Here we take the label as an example to illustrate the basic idea. To get the sub-clustering result of a label cluster, we fix the image clustering result and apply the above co-clustering to the labels in this label cluster. This process repeats until the number of labels in a cluster is smaller than a threshold.

Determining the number of clusters. A widely used method to determine the number of clusters is to evaluate the results with different numbers of clusters, and then choose the one with the best result [42]. The key to this method is the choice of the evaluation measures. As multi-type relational co-clustering considers both representation and relationship information, we thus combine two widely used measures, the coefficient of variance (CV) [43] and sum of squared distance (SSE) [44], to evaluate the clustering results:

$$\text{CV} - \lambda_1 \text{SSE}^I - \lambda_2 \text{SSE}^C, \tag{3}$$

where CV measures the relationship difference between co-clusters. Intuitively, the larger difference between co-clusters, the better the clustering result. SSE evaluates clustering compactness in the representation space. The smaller SSE, the more compact the clustering result. SSE^I and SSE^C are the SSE of the image and label clusters, respectively. λ_1 and λ_2 are two weights to balance these three terms and are both set to be 1 in our tool. The numbers of label clusters and image clusters are then determined by a grid search.

5.2 Set Visualization

All the experts we interviewed required to simultaneously explore the labels, images with detected objects, and relationships between them. A previous study has shown that the node-link-based set visualization has the advantage to visually emphasize all of them as individual objects [45]. As a result, we employ this visualization in MutualDetector.

The node-link-based set visualization is shown in Fig. 1. Labels are placed on the left side of the set visualization as a tree (Fig. 1①). Images are placed on the right side as a matrix (Fig. 1③). Links between them represent their relationships (Fig. 1②). An information panel (Fig. 1(b)) is provided to help examine the details of the extracted labels and detected objects. To further assist exploration, a set of rich interactions, such as zooming and link highlighting [46], are provided. For example, users can select one row to zoom in for analyzing more images in a grid layout (Fig. 5(a)).

5.2.1 Visualization of Labels

To support the exploration of labels, the label hierarchy is shown as an indented tree (Fig. 1①), inspired by the Windows File Explorer. Each rectangle represents one node in the hierarchy. Intermediate nodes are with glyphs (▶ for collapsed nodes and ▼ for expanded nodes) in front, and leaf nodes are without glyphs. The heights of the violet and blue bars in each node represent the precision and recall of the corresponding label calculated on annotated images. The precision and recall of an intermediate node are the averages over its descendants. The label hierarchy can be modified in the visualization by dragging and dropping if it is not satisfactory.

Improving the readability of the label hierarchy. As intermediate nodes in the label hierarchy have no labels to summarize their content, labels are automatically extracted for them based on a knowledge graph (*e.g.*, Wikipedia [47]). For each intermediate node, all of its descendants are first matched with entities in the knowledge graph. Then we find the lowest common ancestor of these matched entities in the knowledge graph and assign its label to the intermediate node. When an intermediate node has only two descendants, we use "&" to connect labels of its descendants, which serves as the label of this intermediate node. The labels generated by this rule are more readable with a suitable length. We also allow users to edit their labels in the visualization if the automatically extracted labels are not satisfactory.

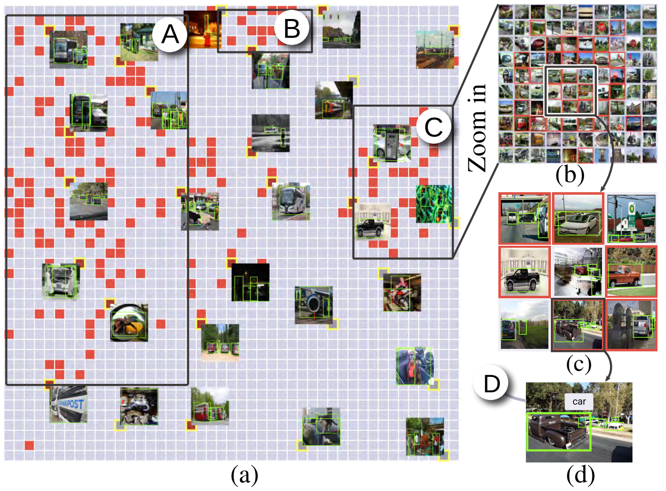


Fig. 5. (a) The grid layout of a selected image cluster; (b) region C after zooming in; (c) a sub-region in (b); (d) a selected image in (c).

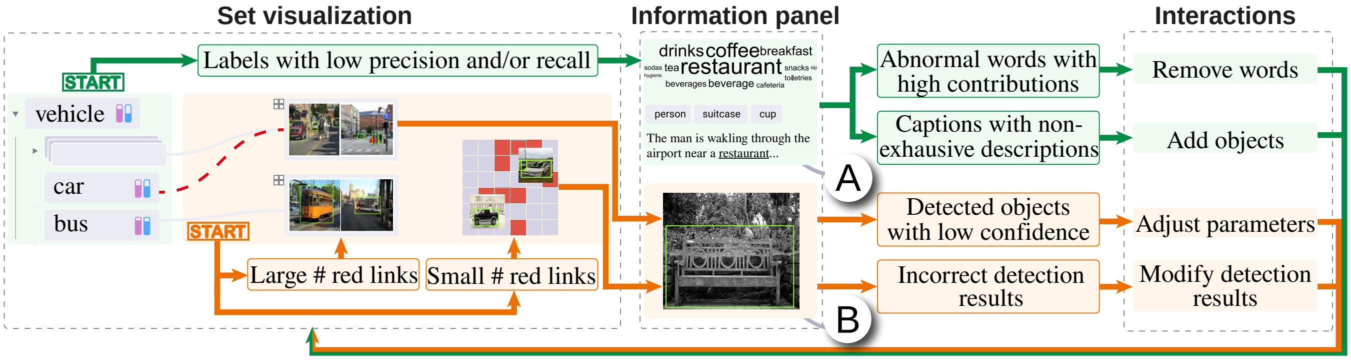
Tree cut. Due to the space limitation, it is difficult to display all labels simultaneously. Therefore, we use a tree cut algorithm [48] to select the part in the hierarchy that users are interested in. The other nodes are either collapsed or stacked (Fig. 1B). For each node x in the hierarchy, its "Degree of Interest" is calculated by

$$\text{DOI}(x|y) = \text{API}(x) - \text{Distance}(x,y) \tag{4}$$

where $\text{Distance}(x,y)$ is the tree-distance between node x and the focus node y that is currently selected by users. $\text{API}(x)$ is the apriori importance of x. It can be set using the check box in Fig. 1A, either as the correctness score or mismatch score. The correctness score is measured by the F1 measure (the harmonic mean of precision and recall), which helps identify incorrectly extracted labels. We use F1 measure rather than accuracy here because most labels are imbalanced in object detection tasks [8]. The mismatch score of a label is the sum of mismatches between all images and that label, which helps identify the labels with many mismatches. Here, one mismatch is counted when the label disagrees with the detected objects in one image.

5.2.2 Visualization of Images

To support the exploration of images and their detected objects, a matrix and a grid layout are used, which have shown their advantage to support content-level analysis [49]. The highest level in the image hierarchy is displayed as a matrix (Fig. 1③) to provide an overview, where each row with ten representative images represents one image cluster. Each row can be zoomed in for further analysis in a grid layout (Fig. 5(a)). With the help of the grid layout, users can explore all the images in one image cluster. If they are not satisfied with one of the representative images, they can replace it by the drag-and-drop operation. Links (Fig. 1②) between the indented tree and matrix represent the matching relationships between label


Set visualization
Information panel
Interactions
START
Labels with low precision and/or recall
vehicle
car
bus
START
Large # red links
Small # red links
drinks coffee breakfast
sodas tea restaurant snacks sip
hygiene toiletries
beverages beverage cafeteria
person
suitcase
cup
A
B
Abnormal words with high contributions
Captions with non-exhaustive descriptions
Detected objects with low confidence
Incorrect detection results
Remove words
Add objects
Adjust parameters
Modify detection results

The man is wakling through the airport near a restaurant...

Fig. 6. A typical improvement process: 1) improving the label extractor (**green**); 2) improving the object detector (**orange**).

clusters and image clusters. A link of red dashed line indicates the number of mismatches between an image cluster and a label cluster is larger than a given threshold.

Representative image. To improve the readability of intermediate nodes in the image hierarchy, we select several representative images for them. As analyzing mismatches is an essential requirement of the experts (**R2**), we use a sampling method motivated by the outlier-biased sampling method [50]. This method preserves the overall distribution while prioritizing the sampling of the images with larger mismatch scores. In particular, the sampling probability of an image is $1/\rho + \pi$. ρ is the density of the local region around the image. Following the outlier-biased sampling method [50], $1/\rho$ is approximated by the radius of k-nearest neighbors. π is the mismatch score of the image, which is the sum of mismatches between its detected objects and extracted labels. The representative images are sampled in a bottom-top manner. For each intermediate node, its representative images are sampled from the representative images of its children.

Image grid. When an image cluster with many mismatches is identified, users can zoom in using ⊞ in front of it to examine the mismatched images and their detected objects in a grid layout. Each cell in the grid layout represents an image. The locations of the cells are determined by the kNN-based grid layout algorithm proposed in [49], which first projects images on a 2D plane as scattered points using t-SNE and then assigns these points to grid cells by solving a linear assignment problem. This layout algorithm places similar images together. Colors are used to indicate the matched images (gray) and mismatched ones (red). If the selected image cluster contains a large number of images, several representative images are displayed first (Fig. 5(a)). These representative images are selected in descending order of their mismatch scores, and their content is displayed near the associated grid cells only when they are not too close to previously displayed images. Users can zoom in the regions with cells of interest and examine more relevant images (Fig. 5(b)). If the sizes of the grid cells are large enough, the image content is then displayed in the corresponding cells.

5.2.3 Information Panel

The information panel consists of two parts: a label panel (Fig. 6A) and an image panel (Fig. 6B).

The **label panel** aims to show important words in captions. Several word summary techniques, such as a word cloud and a list can be used here. A previous study [51] has shown that although a word cloud is not the optimal choice for judging proportions between pairs of values encoded by word sizes, it makes words with larger sizes easier to identify. Moreover, the word cloud is compact in space. Therefore, we employ the word cloud in this panel to show important words in captions, which are those with high contributions to the extraction of a selected label. The word size encodes its contribution. The contribution of a word is estimated by the information-based measure [52]. This measure adds a Gaussian noise to a word and observes the change of the label extractor's prediction on the selected label. A larger change in the prediction means a larger contribution of this word to this label. In our tool, the top 20 words whose contributions are higher than 0.5 are displayed in the word cloud. Users can click a word of interest to examine the associated captions, which are displayed as a list below. The grey rectangles above each caption display the labels extracted from this caption.

The **image panel** shows the selected image and the detected objects in it. As shown in Fig. 6B, the green boxes in the image indicate the locations of detected objects. The grey rectangle above the image shows the label of a selected box. By default, the detected objects with confidence higher than 0.5 are displayed. Users can use the confidence threshold filter above the image to examine the detected objects with different confidence. In this panel, users can modify incorrect detection results, including both the locations and labels of detected objects, or add bounding boxes for undetected objects.

5.3 Mutual Improvement via Interactive Validation

Fig. 6 shows a typical process of how MutualDetector mutually improves the extractor and detector based on user validation. First, the detected objects are used to improve the label extractor. And then, the extracted labels are used to improve the performance of the object detector. Accordingly, this process contains two parts: improving 1) the label extractor and 2) the object detector.

Improving the label extractor. The label extractor can be improved by removing abnormal words with high contributions and adding the objects that are not described in the captions (the **green** part in Fig. 6). The users first focus on identifying abnormal words with high contributions. By examining extracted labels in the tree layout, they find the labels with low precision and/or low recall. By analyzing the word contributions to these labels, the users identify abnormal words with high contributions and remove these words for the extraction of these labels. To remove a word for the extraction of a specific label, they select the word in the word cloud and click the 🗑 in the top-right corner.

After removing the abnormal words, the users continue to find the objects presented in the images but not described in the captions. Then the image representations of these objects are extracted and added to the input for retraining the label extractor.

This compensates for the missing objects in those non-exhaustive captions. Usually, the labels of uncommon objects, such as "giraffe," are unlikely to be ignored in captions. The labels that tend to be ignored often refer to small and common objects, such as "cup." The recall of these easily ignored labels is often low. With this prior knowledge, the users can select the easily ignored objects and click ⊕ in Fig. 1A to add them.

Improving the object detector. The labels extracted from the label extractor are then added to the input for training the object detector. After training, the users can improve the object detector at the global and local levels (the orange part in Fig. 6). They start by checking mismatches between the labels and images. If there is a significant number of red links (mismatches), they turn to the image matrix to check the representative images in the corresponding image clusters and find the root cause of such a large number of mismatches. A typical problem of the detected objects is that the labels are correctly extracted, but the objects are detected with low confidence. It means that the constraint enforcing the consistency between the detected objects and the extracted labels is not strong enough. The users then improve the detector by increasing the weight of the label consistency constraint in the developed semi-supervised object detection method (**global level**).

If the number of red links is small, the users turn to the grid layout to explore local regions with mismatched images (**local level**). During the local exploration, the users interactively modify the incorrect detection results. As the amount of user validation is small, we provide a validation propagation method. This method is motivated by the self-training strategy, which is widely used to augment training data [53], [54]. More specifically, if one bounding box is validated, similar bounding boxes (N nearest neighbors) whose confidence is higher than a threshold are added to the training data. These bounding boxes are then used as seeds to find more nearest bounding boxes. Empirically, we set N as 10, the threshold as 0.9. After the propagation, the model is fine-tuned. This process can be repeated several times until the detection results are satisfactory (*e.g.*, there are no red links).

6 Evaluation

We performed an experiment to quantitatively evaluate the performance of the developed semi-supervised object detection method. A case study with experts E_1 and E_2 was conducted to indicate the usefulness of MutualDetector.

6.1 Quantitative Evaluation on Object Detection

This experiment aims to evaluate the advantages of utilizing both captions and a small number of bounding box annotations.

Datasets. Three popular datasets for object detection were employed for the evaluation. **VOC07** [55] is used in the PASCAL Visual Object Classes Challenge 2007. It contains 5,011 training images and 4,952 test images, and the task is to detect 20 categories of objects. **VOC12** [55] is used in the PASCAL Visual Object Classes Challenge 2012, whose task is the same as **VOC07**. It consists of 11,540 training images. As the ground truth of its test images are not available, a common practice is to use the test images of **VOC07** [2]. **COCO17** [8] is used in the COCO 2017 Object Detection Task. 80 object categories are considered. However, we noticed that small objects (less than 2,000 pixels) were hard to be detected, even for humans. We thus removed the bounding box annotations of such small objects. Moreover, 15 categories with low occurrence frequency in the images (less than 2,000) were also removed. After cleaning, 65 categories, 112,297 training images, and 4,845 test images were obtained.

TABLE 1. Performance comparison between our method and two baselines in terms of mAP (in %) on three datasets.

# annotations	500	1,000	1,500	2,000	2,500
Ours	**51.16**	**61.25**	**63.48**	**65.90**	**66.79**
CSD-SSD	45.68	58.30	60.78	63.48	64.42
Cap2Det	48.52 (no annotations)				

(a) **VOC07**

# annotations	500	1,000	1,500	2,000	2,500
Ours	**52.27**	**58.74**	**63.41**	**65.06**	**67.68**
CSD-SSD	44.08	52.68	59.27	62.59	65.07
Cap2Det	45.11 (no annotations)				

(b) **VOC12**

# annotations	1,000	2,000	3,000	4,000	5,000
Ours	**17.24**	**26.80**	**30.22**	**33.58**	**35.32**
CSD-SSD	13.70	23.82	26.70	31.27	33.27
Cap2Det	23.44 (no annotations)				

(c) **COCO17**

TABLE 2. Computation time (in hour) comparison between our method and two baseline.

Dataset	**VOC07**		**VOC12**		**COCO17**	
# annotations	500	2,500	500	2,500	1,000	5,000
Ours	3.95	4.01	3.93	4.06	**8.44**	8.54
CSD-SSD	**3.89**	**3.93**	**3.92**	**3.94**	8.52	**8.53**
Cap2Det (no annotations)	6.19		8.97		69.84	

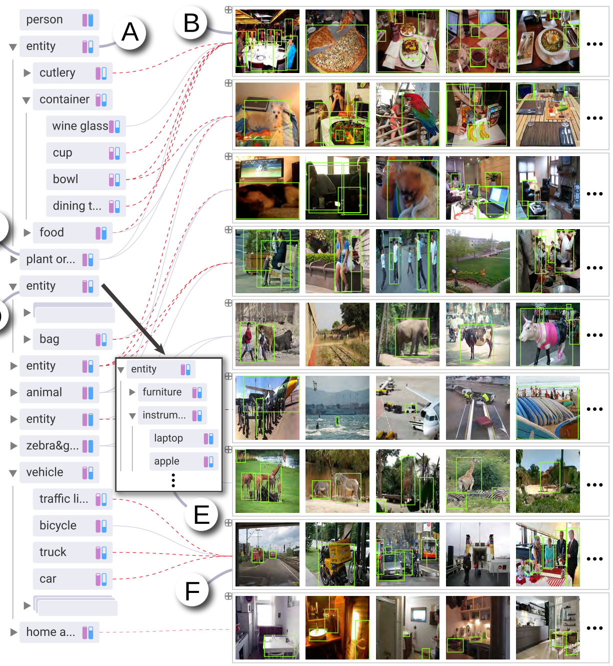
Experimental settings. We used **CSD-SSD** [11] and **Cap2Det** as the baselines, which are the state-of-the-art semi-supervised and caption-based object detection methods, respectively. **Cap2Det** uses only captions for supervision, and **CSD-SSD** uses only the annotated images, while the developed method uses both of them. To save the time and efforts of users to annotate the images for **CSD-SSD** and the developed method, we randomly sampled images and used their ground truth bounding box annotations for training. For **VOC07** and **VOC12**, we randomly sampled m images ($m \in \{500k, k = 1, \ldots, 5\}$). For **COCO17**, as there are more object categories, we randomly sampled more images ($m \in \{1,000k, k = 1, \ldots, 5\}$). For **COCO17**, labels were extracted from captions. As **VOC07** and **VOC12** do not have captions, we used the ground truth labels for supervision. For **CSD-SSD**, the parameters were set as the same in the original papers [11]. For the developed method, we followed [11], and set $\alpha_1, \alpha_2 = 1$ for **VOC07** and **VOC12**, and $\alpha_1 = 2, \alpha_2 = 1$ for **COCO17**. We did not fine-tune the parameters of our method as it has performed well with these parameter settings. We will show in the case study how to fine-tune the parameters to improve the performance with the help of visualization.

Results. We evaluated the performance of object detection by the widely used measure, mAP [55]. The results are shown in Table 1. Our method performs better than **CSD-SSD** in all three datasets. Compared to **Cap2Det**, our method can achieve better performance with only a small number of annotated images. The results show that utilizing both captions and a small number of annotated images can effectively improve the performance of object detection (**R1**). We also evaluated the complexity of our method and the two baselines. As it is hard to analyze the complexity theoretically, we compare their computation time instead. The experiments were run on a server with an Intel Xeon Silver 4214 CPU (2.20GHz) and 10 Nvidia RTX 2080Ti GPUs. The results are shown in Table 2. For each dataset, the computation time of our method and **CSD-SSD** was tested with different numbers of annotated images. We found that there was not much difference in their computation time ($\leq 0.13h$). Due to the page limit, we only show the results with the smallest and the largest numbers of annotated images. Other results can be found in supplemental material. The results show that our method is faster than **Cap2Det** and comparable with **CSD-SSD**. **Cap2Det** has a key step of region proposal which runs on CPU and is very time-consuming [2]. Both our method and **CSD-SSD** are based on SSD [12], which does not have this region proposal step, and is thus faster than **Cap2Det**.

6.2 Case Study

To evaluate how MutualDetector helps analyze and improve the object detector, we invited two experts (E_1 and E_2) who participated in the interviews to carry out a case study on the subset of **COCO17**. The subset is the same as that we have used in the quantitative evaluation. Both E_1 and E_2 are not familiar with the underlying model we used. All images have captions, of which initially $5,000$ have bounding box annotations. The mAP on the test images was 35.32%, which was unsatisfactory. Thus, E_1 and E_2 would like to use MutualDetector to improve the performance. Specifically, E_1 focused on improving the extracted labels, and E_2 focused on improving the detected objects. Both of them also participated in the analysis and discussion of each other's parts. Before the case study, we briefly introduced MutualDetector to them. As we have shown them the prototype during the interviews, they got familiar with the tool in 25 minutes. In the case study, a pair analytics protocol [56] was used, in which the experts drove the exploration and analysis, and we navigated the tool. This strategy is a natural way to capture the reasoning process and allows the experts to focus more on the analysis.

Understanding the relationships between labels and images (R2). To have an overview of the extracted labels and detected objects, E_1 first checked the label clusters, image clusters, and their relationships. There were ten label clusters and nine image clusters (Fig. 7). He noticed that some extracted labels of the label clusters did not precisely summarize their content. For example, the label of label cluster A (Fig. 7) is "entity," which is too high level to well summarize its three children: "cutlery," "container," and "food." By checking their relationships with the image clusters, E_1 found that they were connected to image cluster B (Fig. 7). The representative images of cluster B were mostly taken in dining rooms. Thus, E_1 changed the label of cluster A to "dining room" using the editing function. Similarly, he changed the labels of other four label clusters. During this process, he also found that the clustering result of label cluster D was inaccurate. As shown in Fig. 7E, label "apple" was wrongly clustered with label "laptop." To correct this, E_1 moved "apple" to label cluster C with the name "plant organ", which contained fruits and vegetables.

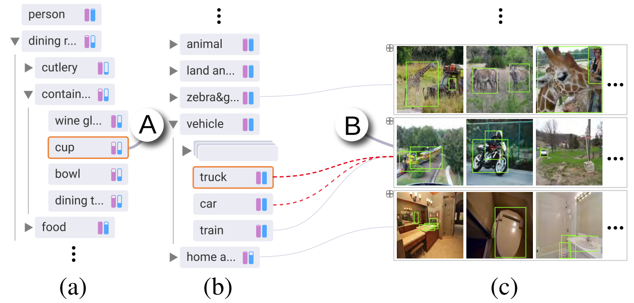


(a) (b)

Fig. 7. (a) The tree layout to show label clusters and (b) the matrix to show image clusters.

Improving extracted labels (R3, R5). E_1 began the analysis by checking the performance of extracted labels first. The prior importance of the tree cut algorithm was set as the correctness score by default to help find labels with poor performance. He found that some labels were low in precision and recall (Fig. 8(a)). Among them, label "cup" had the lowest precision (Fig. 8A). To find the root cause for the poor performance of "cup," E_1 checked the important words for its extraction. He immediately noticed the largest word "restaurant" (Fig. 9A). Cups are indeed common in restaurants. However, it is not necessary that the images of restaurants always contain cups. E_1 suspected the high contribution of "restaurant" might lead to the wrong extraction of "cup" when there were no cups in the images of restaurants.

To confirm that, E_1 clicked "restaurant" in the word cloud and checked the associated captions. He found that some of these captions did not mention cups, and their images did not contain cups (e.g., Fig. 9B). However, label "cup" was extracted from them because they contained the word "restaurant." To tackle this issue, E_1 removed "restaurant" for the extraction of label "cup" using 🗑. Similarly, E_1 also removed "breakfast" and "coffee." After that, "cup" was no longer extracted from the corresponding captions. In addition to wrongly extracted "cup," E_1 also noticed that sometimes "cup" was not extracted from captions whose corresponding images did contain cups (*e.g.*, Fig. 9C). Checking their captions, he found that these captions did not mention the cups in the images. As cups are common objects and small in size, they are easily ignored by the captions. E_1 then added the label "cup" extracted from the images to compensate for the absence of "cup" from the corresponding captions. By removing abnormal words with high contributions and adding labels extracted from images, the precision and recall for the extraction of "cup" were improved from 63.91% and 30.22% to 64.57% and 63.07%. In the similar way, E_1 processed other 15 labels with poor performance. The averaged precision and recall were improved from 88.41% and 61.67% to 90.04% and 69.94%.

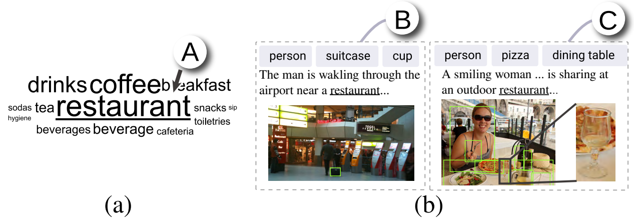


(a) (b) (c)

Fig. 8. The tree layout (a) before improving the extracted labels; (b) after reducing the mismatches at the global level and (c) the associated image matrix of (b).

The object detector was then fine-tuned with the new extracted labels. The mAP was increased from 35.32% to 36.41%.

Reducing mismatches at the global level (R2, R5). After E_1 refined the extracted labels, E_2 would like to check the detected objects and their mismatches. Setting the prior importance of the tree cut algorithm as the mismatch score, he immediately noticed that almost all the image clusters were connected with red links (Fig. 1②). This indicated many mismatches between the detected objects and extracted labels. To find the main reason for so many mismatches, E_2 turned to examine the representative images of the image clusters. He first checked the third image cluster (Fig. 1D), which had the most red links. Examining the ten representative images of this image cluster, he found that five of them had undetected objects, including benches, chairs, umbrellas, person, and cats. To figure out why the detector failed to detect these objects, E_2 selected one image with an undetected bench for detailed examination (Fig. 1C). he found that the bench occupied almost half of the image but was still not detected. He commented that there could be two reasons: the detector failed to detect the object, or it located the object with low confidence. To determine the reason, E_2 turned to the information panel and lowered the confidence threshold to check the detected objects with low confidence. The bounding box of the bench appeared when the confidence threshold was reduced from 0.5 to 0.3 (Fig. 1E). This indicated that the detector successfully located the bench, but with low confidence. Checking the corresponding caption, E_2 found that label "bench" was correctly extracted. E_2 commented, "the object bench would be more confidently detected if the detection was more strongly constrained to be consistent with the extracted label." Similar patterns were found in the other four images with undetected objects. Given the high ratio of such images, E_2 decided to make changes more globally. He increased the weight of the label consistency constraint, α_1, in the semi-supervised object detection method (Eq. (1)). This constraint ensures that the extracted labels and the detected objects are consistent with each other. E_2 tried a few α_1 values (5, 10, 30, 50, and 70) and finally set it to be 50, as the number of mismatches no longer decreased with α_1 larger than 50. The mAP was then increased from 36.41% to 38.74%. With the improved object detector, new image representations were obtained and used to improve the label extraction. The precision and recall of the label extractor were then improved from 90.04% and 69.94% to 90.06% and 69.99%. Based on the improved extracted labels, the mAP of the object detector was further improved to 38.78%. After that, the number of red links was largely reduced (Fig. 8(b)).



(a) (b)

Fig. 9. (a) Important words for the extraction of label "cup"; (b) two examples where "cup" is extracted from the caption but the object cup is not in the image, and the object cup is in the image but "cup" is not extracted from the caption.

Reducing mismatches at the local level (R4, R5). There were still a few red links (Fig. 8(b)). E_2 then utilized the grid layout to examine the remaining mismatches. The eighth image cluster (Fig. 8B) had the most red links, so it was first selected and zoomed into the grid layout. This cluster had mismatches with two labels, "truck" and "car." Selecting "truck" first, the red grids in the grid layout (Fig. 5(a)) represented the mismatched images. E_2 noticed three regions with many mismatched images (Figs. 5A, 5B, and 5C) and decided to examine these regions one by one. He first zoomed in region C and found that in many mismatched images, the locations of trucks were correctly detected (Fig. 5(c)). He examined one of such images in the information panel (Fig. 5(d)) and found that the detected truck in it was wrongly predicted as a car (Fig. 5D). Further inspection showed that both labels "car" and "truck" were extracted from the corresponding caption. Considering trucks and cars did look similar, when both labels were extracted from the caption, the model could not tell them apart. For this image, increasing the weight of the label consistency constraint in Eq. (1) did not help, which explained why mismatches still existed after the global-level adjustment. Object-level validation was needed instead. Similar patterns were also found in the neighboring images. Fig. 5(c) shows some of them. E_2 confirmed the "truck" label for eight detected objects in these images and adjusted three imprecise bounding boxes among them. Similarly, he examined other regions, as well as other labels and image clusters. In total, 351 bounding boxes were confirmed, among which 91 imprecise bounding boxes were adjusted. The mAP was increased from 38.78% to 40.01%. The object detector and label extractor further mutually enhanced each other. The precision and recall of the label extractor were then improved from 90.06% and 69.99% to 90.27% and 70.76%. The mAP of the object detector was improved to 40.12%. After that, there were no red links in the set visualization. The experts were satisfied and stopped the exploration and validation.

Table 3 summarizes the performance improvement after each step in the aforementioned expert validation process. To evaluate the efficiency of MutualDetector, we also compared the performance of MutualDetector with a human-in-the-loop method without MutualDetector (baseline). This baseline method randomly samples images for users to annotate. We simulated the annotation per image by using the associated ground truth bounding boxes. With MutualDetector, the expert only validated 351 bounding boxes and achieved 40.12% mAP. While without MutualDetector, 5,000 more annotated images were required to achieve 39.38% mAP. This comparison shows that MutualDetector largely reduced user annotation efforts and improved the annotation efficiency accordingly.

7 Expert Feedback and Discussion

In this section, we discuss the usability and limitations based on expert feedback.

TABLE 3. Performance improvement with MutualDetector.

Step	Precision	Recall	mAP
Base	88.41%	61.67%	35.32%
Improving extracted labels	90.04%	69.94%	36.41%
Globally reducing mismatches	90.06%	69.99%	38.78%
Locally reducing mismatches	90.27%	70.76%	40.12%

7.1 Usability

After the case study, we conducted seven semi-structured interviews with the four experts we worked with (E_1, E_2, E_3, E_4) and three newly invited ones (E_5, E_6, E_7). The three new experts are Ph.D. students or post-doctors from three different research institutes and have more than two years of experience in object detection research. They are not co-authors of this work. Each of the interviews took between 45 and 60 minutes. In general, all experts have positive comments on MutualDetector. We summarized their feedback into four groups.

Enhancing understanding. E_1 especially praised the simultaneous clustering of labels and images. “It puts similar labels and images together based on not only their representations but also their relationships, which helps me better understand the extracted labels and detected objects from the additional angle of their matching relationships.” Both E_1 and E_7 liked the word cloud that shows important words for the extraction of labels. “It allows me to identify the potential reason for the poor performance of the label extractor,” E_1 said.

Reducing analysis efforts. E_2 was impressed by the 4.80% gain in mAP, with only 351 bounding boxes being validated. He commented, “With this tool, I can quickly identify incorrect detection results. Validating and correcting such detection results brings in relatively large gains compared to randomly selecting images to annotate.” E_6 liked the grid layout a lot, especially the idea of exploring the images and their detected objects at different levels of detail. “It helps me quickly identify the images of interest at the global level and zoom in for details. Also, the grid layout allows me to explore multiple images and their detected objects simultaneously, which makes the exploration more efficient.”

Learning curve. The visual metaphors employed in MutualDetector are commonly used, such as node-link-based set visualization, grid layout, and indented tree. It took an average of 21.3 (STDEV=2.6) minutes for the experts to become familiar with the tool. The experts believed that the performance gain achieved with MutualDetector outweighed the learning cost. To help users get familiar with the tool quickly, we also provide a tour function to illustrate the visual encodings and interactions.

Generalization. In our implementation, the widely used SSD is employed as the base object detector to demonstrate the idea of the developed semi-supervised object detection method. Other annotation-based object detection methods, such as YOLO [1], can also be used in our method. The bounding box consistency constraint in Eq. (1) needs to be replaced with the loss function of the according object detection method, while the label consistency and robustness constraints can be directly applied. Moreover, the set visualization is also model-agnostic. It presents extracted labels and detected objects regardless of the underlying label extractor and object detector.

7.2 Limitations and Future Work

In addition to the aforementioned positive feedback, several limitations of MutualDetector are also identified, which give directions for future improvement.

Extensive evaluation. Although we developed MutualDetector in collaboration with six machine learning experts, they commented that practitioners could also use this tool to improve the performance of object detection. This is because our tool does not require the users to understand the inner workings of the underlying model. In addition, the experts mentioned that more visual cues, such as the ratio of images with large mismatch scores in each cluster, would guide them in the exploration. However, they also worried that the complexity induced by more functions might confuse users. To better investigate the usefulness of MutualDetector, we plan to share it with practitioners and collect feedback from them for further improving the usability, especially the visualization design.

Time efficiency. When user validation is provided, the model is fine-tuned with the updated training data. This process is time-consuming. For example, it takes around 5 hours to fine-tune the model for the **COCO17** dataset, which contains 112,297 training images. The training time becomes intolerable when the dataset is larger. As a result, how to update the object detector more efficiently (e.g., incremental training only with validated images) is an interesting venue for future work.

Parameter adjustment. The two balance weights (Eq. (1)) in the developed semi-supervised object detection method directly affect the performance. Thus, the experts who collaborated with us would like to adjust them for better performance. Our tool can guide the adjustment direction of these two weights (increased or decreased). However, it cannot indicate how much the weight should be increased or decreased. In the case study, the experts tried five different values for the weight of the label consistency constraint. It is still time-consuming. Thus, it would be interesting to investigate an effective parameter adjustment method. A potential solution is to infer a more suitable parameter setting from user validation as it provides hints about the deficiency of the detector.

Validation confidence. Currently, MutualDetector regards user validation as ground truth. However, it may not always be correct. Incorrect user validation usually leads to model performance degradation. In order to avoid such incorrect validation, a potential solution is to evaluate the confidence of validation. When the confidence is low, an alert for rechecking together with a visual explanation can be presented in the visualization. The visual explanation can facilitate the further improvement of the validation.

8 Conclusion

We have developed MutualDetector, a visual analysis tool to explore and improve the performance of object detectors trained on image captions and a small number of bounding box annotations. A semi-supervised object detection method is developed to utilize both captions and a small number of bounding box annotations to build an effective object detector. A node-link-based set visualization supported by a multi-type relational co-clustering algorithm is developed to help explore labels extracted from captions, images with detected objects, and their relationships. With the visualization, users can validate the extracted labels and detected objects. Based on the user validation, the label extractor and object detector can enhance each other and be further improved. The quantitative evaluation on the developed semi-supervised object detection method and the case study carried out by two experts demonstrated the effectiveness and usefulness of MutualDetector.

References

[1] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[2] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2015, pp. 1–9.

[3] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020.

[4] H. Bilen and V. Andrea, "Weakly supervised deep detection network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2846–2854.

[5] K. Ye, M. Zhang, A. Kovashka, W. Li, D. Qin, and J. Berent, "Cap2Det: Learning to amplify weak caption supervision for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9686–9695.

[6] Y. Shen, R. Ji, Z. Chen, Y. Wu, and F. Huang, "UWSOD: Toward fully-supervised-level capacity weakly supervised object detection," in *Proceedings of the Advances in Neural Information Processing Systems*, 2020, pp. 7005–7019.

[7] B. Long, Z. Zhang, X. Wu, and P. S. Yu, "Spectral clustering for multi-type relational data," in *Proceedings of the International Conference on Machine Learning*, 2006, pp. 585–592.

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 740–755.

[9] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models." in *Proceedings of the Advances in Neural Information Processing Systems*, 2010, pp. 2–9.

[10] K. Wang, X. Yan, D. Zhang, L. Zhang, and L. Lin, "Towards human-machine cooperation: Self-supervised sample mining for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1605–1613.

[11] J. Jeong, S. Lee, J. Kim, and N. Kwak, "Consistency-based semi-supervised learning for object detection," in *Proceedings of the Advances in Neural Information Processing Systems*, 2019, pp. 1–10.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 21–37.

[13] J. Yuan, C. Chen, W. Yang, M. Liu, J. Xia, and S. Liu, "A survey of visual analytics techniques for machine learning," *Computational Visual Media*, vol. 7, no. 1, pp. 1–34, 2020.

[14] J. H. Park, S. Nadeem, S. Mirhosseini, and A. Kaufman, "C^2A: Crowd consensus analytics for virtual colonoscopy," in *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*, 2016, pp. 21–30.

[15] S. Liu, C. Chen, Y. Lu, F. Ouyang, and B. Wang, "An interactive method to improve crowdsourced annotations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 235–245, 2019.

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[17] S. Xiang, X. Ye, J. Xia, J. Wu, Y. Chen, and S. Liu, "Interactive correction of mislabeled training data," in *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*, 2019, pp. 57–68.

[18] A. Bäuerle, H. Neumann, and T. Ropinski, "Classifier-guided visual correction of noisy labels for image classification tasks," *Computer Graphics Forum*, vol. 39, no. 3, pp. 195–205, 2020.

[19] J. Moehrmann, S. Bernstein, T. Schlegel, G. Werner, and G. Heidemann, "Improving the usability of hierarchical representations for interactively labeling large image data sets," in *Proceedings of the International Conference on Human-Computer Interaction*, 2011, pp. 618–627.

[20] M. Khayat, M. Karimzadeh, J. Zhao, and D. S. Ebert, "VASSL: A visual analytics toolkit for social spambot labeling," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 874–883, 2020.

[21] J. Eirich, J. Bonart, D. Jäckle, M. Sedlmair, U. Schmid, K. Fischbach, T. Schreck, and J. Bernard, "IRVINE: A design study on analyzing correlation patterns of electrical engines," *IEEE Transactions on Visualization and Computer Graphics (to be appear)*, 2021.

[22] O. Rooij, J. van Wijk, and M. Worring, "MediaTable: Interactive categorization of multimedia collections," *IEEE Computer Graphics and Applications*, vol. 30, no. 5, pp. 42–51, 2010.

[23] M. Stein, H. Janetzko, T. Breitkreutz, D. Seebacher, T. Schreck, M. Grossniklaus, I. D. Couzin, and D. A. Keim, "Director's cut: Analysis and annotation of soccer matches," *IEEE Computer Graphics and Applications*, vol. 36, no. 5, pp. 50–60, 2016.

[24] B. Höferlin, R. Netzel, M. Höferlin, D. Weiskopf, and G. Heidemann, "Inter-active learning of ad-hoc classifiers for video visual analytics," in *Proceedings of the Conference on Visual Analytics Science and Technology*, 2012, pp. 23–32.

[25] K. Kurzhals, M. Hlawatsch, C. Seeger, and D. Weiskopf, "Visual analytics for mobile eye tracking," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 301–310, 2017.

[26] F. Lekschas, B. Peterson, D. Haehn, E. Ma, N. Gehlenborg, and H. Pfister, "PEAX: Interactive visual pattern search in sequential data using unsupervised deep representation learning," *Computer Graphics Forum*, vol. 39, no. 3, pp. 167–179, 2020.

[27] L. S. Snyder, Y.-S. Lin, M. Karimzadeh, D. Goldwasser, and D. S. Ebert, "Interactive learning for identifying relevant tweets to support real-time situational awareness," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 558–568, 2020.

[28] F. Sperrle, R. Sevastjanova, R. Kehlbeck, and M. El-Assady, "VIANA: Visual interactive annotation of argumentation," in *Proceedings of the Conference on Visual Analytics Science and Technology*, 2019, pp. 11–22.

[29] C. Chen, Z. Wang, J. Wu, X. Wang, L.-Z. Guo, Y.-F. Li, and S. Liu, "Interactive graph construction for graph-based semi-supervised learning," *IEEE Transactions on Visualization and Computer Graphics*, 2021, to be published, doi: 10.1109/TVCG.2021.3084694.

[30] S. Jia, Z. Li, N. Chen, and J. Zhang, "Towards visual explainable active learning for zero-shot classification," *IEEE Transactions on Visualization and Computer Graphics*, 2021, to be published, doi: 10.1109/TVCG.2021.3114793.

[31] W. Yang, Z. Li, M. Liu, Y. Lu, K. Cao, R. Maciejewski, and S. Liu, "Diagnosing concept drift with visual analytics," in *Proceedings of IEEE Conference on Visual Analytics Science and Technology*, 2020, pp. 12–23.

[32] L. Gou, L. Zou, N. Li, M. Hofmann, A. K. Shekar, A. Wendt, and L. Ren, "VATLD: A visual analytics system to assess, understand and improve traffic light detection," *IEEE Transactions on Visualization and Vomputer Graphics*, vol. 27, no. 2, pp. 261–271, 2021.

[33] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price *et al.*, "Scaling egocentric vision: The EPIC-Kitchens dataset," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 720–736.

[34] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," in *2020 International Joint Conference on Neural Networks*, 2020, pp. 1–8.

[35] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.

[36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics*, 2019, pp. 4171–4186.

[37] Y. Chen, Y. Li, T. Kong, L. Qi, R. Chu, L. Li, and J. Jia, "Scale-aware automatic augmentation for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9563–9572.

[38] N. Dingwall and C. Potts, "Mittens: An extension of glove for learning domain-specialized representations," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, pp. 212–217.

[39] K. Cao, M. Liu, H. Su, J. Wu, J. Zhu, and S. Liu, "Analyzing the noise robustness of deep neural networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 7, pp. 3289–3304, 2021.

[40] N. Lei, D. An, Y. Guo, K. Su, S. Liu, Z. Luo, S.-T. Yau, and X. Gu, "A geometric understanding of deep learning," *Engineering*, vol. 6, no. 3, pp. 361–374, 2020.

[41] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 91–100, 2017.

[42] O. Maimon and L. Rokach, *Data mining and knowledge discovery handbook*. New York, USA: Springer, 2005.

[43] B. Long, Z. Zhang, and S. Y. Philip, *Relational data clustering: models, algorithms, and applications*. Florida, USA: CRC Press, 2010.

[44] T. Thinsungnoena, N. Kaoungkub, P. Durongdumronchaib, K. Kerdprasopb, and N. Kerdprasopb, "The clustering validity with silhouette and sum of squared errors," in *Proceedings of the International Conference on Industrial Application Engineering*, 2015, pp. 44–51.

[45] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers, "The state-of-the-art of set visualization," *Computer Graphics Forum*, vol. 35, no. 1, pp. 234–260, 2016.

[46] F. Hohman, H. Park, C. Robinson, and D. H. P. Chau, "SUMMIT: Scaling deep learning interpretability by visualizing activation and attribution summarizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 1096–1106, 2019.

[47] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

[48] G. W. Furnas, "Generalized fisheye views," *ACM Sigchi Bulletin*, vol. 17, no. 4, pp. 16–23, 1986.

[49] C. Chen, J. Yuan, Y. Lu, Y. Liu, H. Su, S. Yuan, and S. Liu, "OoDAnalyzer: Interactive analysis of out-of-distribution samples," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 7, pp. 3335–3349, 2021.

[50] J. Yuan, S. Xiang, J. Xia, L. Yu, and S. Liu, "Evaluation of sampling methods for scatterplots," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 1720–1730, 2021.

[51] C. Felix, S. Franconeri, and E. Bertini, "Taking word clouds apart: An empirical investigation of the design space for keyword summaries," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 657–666, 2017.

[52] C. Guan, X. Wang, Q. Zhang, R. Chen, D. He, and X. Xie, "Towards a deep and unified understanding of deep neural models in NLP," in *Proceedings of the International Conference on Machine Learning*, 2019, pp. 2454–2463.

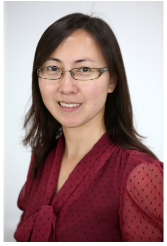
[53] Y. Li, D. Huang, D. Qin, L. Wang, and B. Gong, "Improving object detection with selective self-supervised self-training," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 589–607.

[54] Y. Wang, J. Peng, and Z. Zhang, "Uncertainty-aware pseudo label refinery for domain adaptive semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9092–9101.

[55] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[56] R. Arias-Hernandez, L. T. Kaastra, T. M. Green, and B. Fisher, "Pair analytics: Capturing reasoning processes in collaborative visual analytics," in *Proceedings of the Hawaii International Conference on System Sciences*, 2011, pp. 1–10.

Changjian Chen is now a Ph.D. student at Tsinghua University. His research interests are in interactive machine learning. He received a B.S. degree from University of Science and Technology of China.



Jing Wu is a lecturer in computer science and informatics at Cardiff University, UK. Her research interests are in computer vision and graphics including image-based 3D reconstruction, face recognition, machine learning and visual analytics. She received BSc and MSc from Nanjing University, and Ph.D. from the University of York, UK. She serves as a PC member in CGVC, BMVC, etc., and is an active reviewer for journals including Pattern Recognition, Computer Graphics Forum, etc.

Xiaohan Wang received the Ph.D. degree in computer science from University of Technology Sydney, Australia, in 2021. He received the B.E. degree from University of Science and Technology of China, China, in 2017. He is currently a postdoc researcher in the College of Computer Science and Technology, Zhejiang University, China. His research interests are video analysis and understanding.

Shouxing Xiang received the BS degree from Tsinghua University. He is currently working toward the MD degree with Tsinghua University. His research interest includes interactive data quality improvement.

Song-Hai Zhang received the PhD degree of Computer Science and Technology from Tsinghua University, Beijing, in 2007. He is currently an associate professor in the Department of Computer Science and Technology at Tsinghua University. His research interests include image/video analysis and processing as well as geometric computing.

Qifeng Tang is the CEO of Shanghai Lianshu IoT Co. Ltd. His research interests include computer vision and big data applications. He is a Ph.D. student of East China University of Science and Technology and received EMBA from Beihang University.

Shixia Liu is a professor at Tsinghua University. Her research interests include visual text analytics, visual social analytics, interactive machine learning, and text mining. She worked as a research staff member at IBM China Research Lab and a lead researcher at Microsoft Research Asia. She received a B.S. and M.S. from Harbin Institute of Technology, a Ph.D. from Tsinghua University. She is a fellow of IEEE and an associate editor-in-chief of IEEE Trans. Vis. Comput. Graph.

Supplemental Material: Towards Better Caption Supervision for Object Detection

Changjian Chen, Jing Wu, Xiaohan Wang, Shouxing Xiang, Song-Hai Zhang, Qifeng Tang, Shixia Liu

Appendix A: Semi-Supervised Object Detection

Given labels extracted from image captions and a small number of object bounding box annotations, the proposed semi-supervised object detection method trains a detector with a bounding box consistency constraint, a label consistency constraint, and a robustness constraint.

$$\mathscr{L}_{\text{fine}} + \alpha_1 \mathscr{L}_{\text{coarse}} + \alpha_2 \mathscr{L}_{\text{robust}}. \tag{1}$$

Bounding box consistency constraint. The first term $\mathscr{L}_{\text{fine}}$ ensures the detected objects of annotated images to be consistent with the bounding box annotations, including both the locations and labels. Here we use the same MultiBox loss as in SSD [1]. $\mathscr{L}_{\text{fine}}$ is the averaged loss over MultiBox losses of all annotated images. For each image, the MultiBox loss is:

$$\frac{1}{M}\left(\mathscr{L}_{conf} + \mu \mathscr{L}_{loc}\right), \tag{2}$$

where M is the number of detected bounding boxes. μ is set to 1 as in SSD [1]. $\mathscr{L}_{loc}$ is the localization loss that measures the difference between the locations of the detected bounding boxes and the ground truth bounding boxes, i.e.,

$$\mathscr{L}_{loc} = \sum_i^M \sum_j^N \sum_{m\in\{cx,cy,w,h\}} \delta_{ij} \cdot \text{smooth}_{\text{L1}}(l_i^m - g_j^m). \tag{3}$$

N is the number of ground truth bounding boxes. l_i^{cx} and l_i^{cy} are offsets for the center of the i-th detected bounding box. l_i^w and l_i^h are the width and height of the i-th detected bounding box. g_j^{cx} and g_j^{cy} are normalized offsets for the center of the j-th ground truth bounding box. g_j^w and g_j^h are the normalized width and height of the h-th ground truth bounding box. $\delta_{ij} = \{0,1\}$ is an indicator for the matching between i-th predicted bounding box and j-th ground truth bounding box. Two bounding boxes are matched if their Jaccard overlap is higher than 0.5.

$\mathscr{L}_{conf}$, the confidence loss, measures the difference between the labels of the detected bounding boxes and the ground truth bounding boxes.

$$\mathscr{L}_{conf} = -\sum_i^M \sum_p^P c_i^p \log \hat{c}_i^p. \tag{4}$$

$c_i^p = 1$ means that the i-th detected bounding box matches to the ground truth bounding box with the p-th label. otherwise $c_i^p = 0$. $\hat{c}_i^p$ is the confidence score that the i-th detected bounding box has the p-th label. P is the number of labels.

Label consistency constraint. The second term $\mathscr{L}_{\text{coarse}}$ is the label consistency loss that is applied to images with captions. It enforces that the labels of detected objects in an image must be consistent with the labels extracted from its captions. $\mathscr{L}_{\text{coarse}}$ is the averaged label consistency loss of all images with captions. For each image, the label consistency loss is:

$$-\sum_p^P e^p \log \hat{e}^p. \tag{5}$$

$e^p = 1$ means that the p-th label is extracted from the caption; otherwise $e^p = 0$. $\hat{e}^p$ is the confidence score that this image contains objects of the p-th label, which is calculated by

$$\hat{e}^p = \max_i \hat{c}_i^p. \tag{6}$$

Robustness constraint. The third term $\mathscr{L}_{\text{robust}}$ is the robustness constraint to ensure the detector to be robust to given perturbed inputs. $\mathscr{L}_{\text{robust}}$ is the averaged robustness constraint of all images with captions. For each image, the robustness constraint is:

$$\frac{1}{M}\left(\mathscr{L}_{robust-conf} + \mathscr{L}_{robust-loc}\right), \tag{7}$$

where M is the number of detected bounding boxes. $\mathscr{L}_{robust-loc}$ is the robustness constraint for locations that measures the difference between the detected bounding boxes in the original image and perturbed image.

$$\mathscr{L}_{robust-loc} = \sum_i^M e^p \sum_{m\in\{cx,cy,w,h\}} \delta_{ij} \cdot \text{smooth}_{\text{L1}}(l_i^m - \bar{l}_i^m). \tag{8}$$

l_i^{cx}, l_i^{cy}, l_i^w, l_i^h are the locations and sizes of detected bounding boxes in the original image, as described in Eq.(3). $\bar{l}_i^{cx}$, $\bar{l}_i^{cy}$, $\bar{l}_i^w$, $\bar{l}_i^h$ are the locations and sizes of detected bounding boxes in the perturbed image. e^p in the constraint ensures that the robustness constraint is only applied to detected objects whose labels are extracted from the captions.

$\mathscr{L}_{robust-conf}$ is the robustness constraint for classification that measures the difference between the predicted labels of detected objects in the original image and perturbed image.

$$\mathscr{L}_{robust-conf} = \sum_i^M JS(\mathbf{c}_i, \bar{\mathbf{c}}_i). \tag{9}$$

$\mathbf{c}_i$ ($\bar{\mathbf{c}}_i$) is the prediction vector of i-th detected bounding box in the original image (perturbed image). $\mathbf{c}_i = [c_i^1, c_i^2, ..., c_i^P]$. $\bar{\mathbf{c}}_i = [\bar{c}_i^1, \bar{c}_i^2, ..., \bar{c}_i^P]$. $JS(\cdot,\cdot)$ is Jensen-Shannon divergence.

Appendix B: Computation Time

Table 1 shows the computation time of our method and the two baselines. The experiments were run on a server with an Intel Xeon Silver 4214 CPU (2.20GHz) and 10 Nvidia RTX 2080Ti GPUs. The results show that our method is faster than **Cap2Det** and comparable with **CSD-SSD**. **Cap2Det** has a key step of region proposal which runs on CPU and is very time-consuming. Both our method and **CSD-SSD** are based on SSD, which does not have this region proposal step, and thus are faster than **Cap2Det**.

Table 1: Computation time (in hour).

# annotations	500	1,000	1,500	2,000	2,500
Ours	3.95	**3.92**	4.12	3.94	4.01
CSD-SSD	**3.89**	**3.92**	**4.05**	**3.93**	**3.93**
Cap2Det	6.19 (no annotations)				

(a) **VOC07**

# annotations	500	1,000	1,500	2,000	2,500
Ours	3.93	**3.76**	3.97	3.99	4.06
CSD-SSD	**3.92**	3.78	**3.87**	**3.85**	**3.94**
Cap2Det	8.97 (no annotations)				

(b) **VOC12**

# annotations	1,000	2,000	3,000	4,000	5,000
Ours	**8.44**	8.48	8.46	8.57	8.54
CSD-SSD	8.52	**8.45**	**8.42**	**8.51**	**8.53**
Cap2Det	69.84 (no annotations)				

(c) **COCO17**

References

[1] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot MultiBox detector,” in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 21–37.