Contrastive Proposal Extension With LSTM Network for Weakly Supervised Object Detection

Pei Lv, Suqi Hu, and Tianran Hao

Abstract—Weakly supervised object detection (WSOD) has attracted more and more attention since it only uses image-level labels and can save huge annotation costs. Most of the WSOD methods use Multiple Instance Learning (MIL) as their basic framework, which regard it as an instance classification problem. However, these methods based on MIL tend to only converge on the most discriminative regions of different instances, rather than their corresponding complete regions, that is, insufficient integrity. Inspired by the human habit of observing things, we propose a new method by comparing the initial proposals and the extended ones to optimize those initial proposals. Specifically, we propose one new strategy for WSOD by involving contrastive proposal extension (CPE), which consists of multiple directional contrastive proposal extensions (D-CPEs), and each D-CPE contains LSTM-based encoders and dual-stream decoders. Firstly, the boundary of initial proposals in MIL is extended to different positions according to well-designed sequential order. Then, the CPE compares the extended proposal and the initial one by extracting the feature semantics of them using the encoders, and calculates the integrity of the initial proposal to optimize its score. These contrastive contextual semantics will guide the basic WSOD to suppress bad proposals and improve the scores of good ones. In addition, a simple dual-stream network is designed as the decoder to constrain the temporal coding of LSTM and improve the performance of WSOD further. Experiments on PASCAL VOC 2007, VOC 2012 and MS-COCO datasets show that our method has achieved the state-of-the-art results.

Index Terms—Object detection, feature extraction, training, Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNNs), supervised learning, image sequence analysis, image annotation, object recognition, semantics, context awareness.

I. INTRODUCTION

Weakly supervised object detection (WSOD), which only uses image-level labels to train the detection model, has attracted the wide attention of researchers. Compared with the fully supervised object detection (FSOD) task [1], [2], [3], [4], which requires bounding-box annotations for massive amounts of data, WSOD is relatively time-saving, labor-saving and promising, while still has some challenges to obtain satisfactory performance.

Fig. 1. The scanning way in locating complete objects. (a) represents a certain part of the object in the observed scene, (b) represents the trend of subsequent observation based on previous column, and (c) represents the results of further observation. The instance in each row contains a certain part of the object, and its extensions in different directions.

Nowadays, most of the WSOD methods use Multiple Instance Learning (MIL) as their basic framework. They regard WSOD as an instance classification problem, and use the instance classifier to approach the goal of object detection. One of the most representative work is the Weakly Supervised Deep Detection Network (WSDDN) [5], which firstly combines MIL with WSOD. Due to its effectiveness and conciseness, plenty of following work are further carried out based on it. The work in [6] and [7] proposed an online refinement module to solve the problem that WSDDN only locates the object region with the highest salience score, rather than the whole object. However, WSOD methods are still far from achieving the competitive results generated by FSOD methods.

In the real world, we often take a scanning way to observe objects as shown in Fig. 1. When only the local part of the object is perceived, we will be in the habit of continuing the observation for further exploration and searching its remaining parts in a certain order. It is very suitable to model this sequential process using recurrent neural networks, such as LSTM network. In MIL, an image contains multiple different initial proposal regions, however, the objects in most of the proposal regions are incomplete. Therefore, when a single proposal is observed, we can regard it as a certain part of one object and make further observations of its neighboring area. Under these situations, this single instance can be involved to form a direct comparison with other further observed instances. The direct comparison will allow us to conduct an evaluation of the initial instance in two cases: 1) if the further observed region (extended proposal) is similar to the previous one (original proposal), the object in the observed region is
likely to be incomplete. 2) if the further observed region is obviously different from the original one, the original proposal prefers to containing a complete object.

Inspired by the above facts, we propose a new Contrastive Proposal Extension (CPE) module for basic WSDDN framework, which mimics the scanning way of human beings to extract the contrastive contextual semantics of different instances. In CPE, each adjacent row or adjacent column in the proposal feature is regarded as successive time series data. In order to extract the semantic feature between them, we choose the Long Short-Term Memory (LSTM) network [8] as the basic component of CPE. The CPE module contains multiple Directional Contrastive Proposal Extension (D-CPE) sub modules, and each D-CPE is composed of two encoders based on LSTM network and two decoders based on dual-stream structure network.

In detail, the initial proposal is expanded in different directions to produce multiple extended proposals, which are used to compare with the original proposal for the integrity evaluation of the original proposal. D-CPEs extract different RoI pooling features for the extended proposal and the initial one, and then encode them individually. In the meantime, the decoder is involved to constrain the hidden space of the LSTM encoder. The D-CPE further calculates the contrastive contextual semantics between the initial proposal and the extended one, where the larger the contextual semantic difference between them, the greater the probability that the initial proposal is relatively complete. Afterwards, a number of contrastive contextual semantics in different directions are combined to jointly estimate the probability of the completeness of this proposal. In the forward propagation process of the WSDDN network, it provides basic detection vectors and classification vectors, and the CPE module provides contrastive contextual semantics. The contrastive contextual semantics will guide the detection vectors and classification vectors in WSDDN to generate more reasonable confidence scores, that is, increasing the confidence scores of instances with high probability of completeness, and suppressing the confidence scores of instances with low probability. We report the experimental results on PASCAL VOC 2007, 2012 and MS-COCO datasets, and compare our method with other representative approaches.

The main contributions are summarized as follows:

- We propose one new Contrastive Proposal Extension (CPE) module in WSOD, which estimates the contextual semantic difference by exploring the comparable relationship between the original instance and its corresponding extended ones.
- We introduce the LSTM network to encode the contextual semantics into the framework of WSOD, and effectively alleviate the mismatching problem between instance score and instance integrity.
- The experiments on PASCAL VOC 2007, PASCAL VOC 2012 and MS-COCO show that our proposed method has achieved the state-of-the-art results.

II. RELATED WORK

Most of the existing work regards the task of WSOD as one MIL problem, and trains the model with image-level labels. To further improve the performance, some methods try to combine weakly supervised semantic segmentation with WSOD [9]. Moreover, in some FSOD methods, Recurrent Neural Network also played a considerable role [10]. Therefore, in this section, we mainly review above representative related work.

A. Weakly Supervised Object Detection

In order to solve the task of only using image-level labels to realize object detection, most of existing work are proposed based on MIL framework. For such kinds of methods, an image can be regarded as an instance package containing multiple instances, and these instances will be used for model training to select the most reasonable candidate instances. These WSOD methods mainly consist of two consecutive stages, object discovery and instance refinement. The object discovery stage usually combines MIL and CNNs to implicitly model the localization of potential objects using image-level labels [11], [12], [13], [14], [15], [16]. In addition, contextual information [17], attention mechanism [18], gradient map [19] and semantic segmentation [20] are also used to mine the complete object proposal. To optimize the overall framework, some approaches tried to use minimum entropy prior [21], multi-view learning [22], and generative adversarial learning [23]. Beyond that, to alleviate the problem of non-convexity, Wan et al. [24] introduced the continuous optimization algorithm into MIL, and proposed an effective continuation MIL (C-MIL) method. Arun et al. [25] designed the WSOD network based on incoherence coefficients by exploiting the difference between the distribution of unknown predictions and the label-aware conditional distribution. Chen et al. [26] proposed a spatial likelihood voting (SLV) to aggregate the process of proposal localizing, using voting results to regularize the bounding-box. To detect the complete region of objects, recent work began to focus on adopting cascaded refinements of MIL classifications [7], leveraging continuity strategies [21] and modeling the location uncertainty [25].

Contrastive learning [27], [28], [29], as a self-supervised learning method, in the absence of labels, can learn more general features from unlabeled data before performing tasks such as classification. After that, using other techniques to fine-tune the pre-trained model will effectively improve the object detection performance even better than supervised training and weakly supervised training with image-level labels.

Inspired by the idea of comparing the similarities and differences of data in contrastive learning, we use CPE to extract the contrastive contextual semantics between the initial proposal and the extended one, which is further combined with the semantic information of the backbone network to estimate a more reasonable score of the initial proposal.

B. Weakly Supervised Semantic Segmentation

Weakly supervised semantic segmentation (WSSS) task [30], [31] is closely related to the WSOD, and they are often combined together to achieve better performance for their respective goals. To solve the problem of the
lack of object border annotations in WSOD, some recent work combined the segmentation map or class activation map with the WSDDN to obtain more complete objects. Shen et al. [9] proposed a joint weakly-supervised detection and segmentation framework into a multi-task learning framework. Diba et al. [32] generated candidate regions based on the segmentation results, and used these regions for multi-instance learning. Similarly, Gao et al. [33] used Affinity Net in [34] to generate segmentation maps as an additional basis for object detection. Cheng et al. [35] used the segmentation maps generated by Grad-CAM [36] to filter the initial proposal boxes, which reduced the interference of additional boxes and obtained initial regions with higher quality. In addition, Zeng et al. [37] proposed a bottom-up and top-down object distillation framework (WSOD2) that fully combined with deep objectness representation learned from bottom-up evidences. Although the introduction of weakly-supervised semantic segmentation maps is conducive to detecting more complete objects, this kind of methods still can not extract all the contents of the image completely, and have certain limitations.

To avoid the problem caused by incomplete segmentation map and class activation map, we directly compare the features of original instances and the extended ones, which allows CPE to detect subtle differences between the extended and the initial region, further determining the integrity of the initial instance.

C. Recurrent Neural Network

Recurrent Neural Network (RNN) has a variety of forms, such as Bidirectional RNNs, LSTM, GRU, Graph LSTM, etc. They have a wide range of applications in different fields, including speech recognition, semantic segmentation, trajectory prediction, natural language processing and object detection. Yan et al. [38] adopted LSTM to construct a spatial loop layer to directly capture the global context and improve the feature representation for semantic segmentation. Bell et al. [39] used RNN to construct a spatial recurrent neural network to integrate the contextual information outside the region of interest for object detection. In [40], Stewart et al. used the LSTM network to decode the extracted features, generated time sequences, and performed end-to-end training to detect humans in dense areas. Li et al. [10] combined CNN with LSTM to propose an object detection model based on a novel attention to context convolution neural network (AC-CNN), which fused the global context with the local context to make the final detection. Liang et al. [41] proposed a Graph Long-short Memory Network for semantic object parsing. Yan et al. [42] proposed a Pixel Recurrent Neural Network to model the discrete probability of the pixel values and encode the complete set of dependencies in the image.

Inspired by the human habit of taking a scanning way to observe things in sequential order, we choose the relatively powerful and flexible LSTM as the basic structure of the CPE module among these various RNNs. In practice, although GRU is faster and simpler than LSTM, it always uses cell state indiscriminately, which is not conducive to extracting the contextual semantics between instances and their extensions.

III. PROPOSED APPROACH

Our work focuses on increasing the degree of matching of the initial proposal evaluated by the detector to improve the performance of WSOD. We build our method with online instance classifier refinement branches (OICR) as the basic framework, and the overall framework is shown in Fig. 2. The CPE consists of multiple D-CPEs, which is responsible for extracting the contrastive semantics between the initial proposals and the extended ones in different directions. In the training stage, different D-CPEs will be fused together and combined with the semantic information of the basic framework to jointly estimate the final score of initial proposals.

A. Basic MIL Network

Bilen et al. [5] firstly proposed a WSDDN framework for WSOD by combining with MIL. Given an image I, and corresponding image-level labels \( y = [y_1, y_2, \ldots, y_C]^T \), we can obtain initial proposals \( B = \{B_1, B_2, \ldots, B_N\} \) by WSDDN, where C is the total number of categories, \( y_c = 1 \) or 0 represents whether the image contains category c, and \( N \) is the total number of initial proposals generated by proposal generation methods. Firstly, I passes through the CNN backbone layer and RoI pooling layer to generate a series of proposal features for the initial proposals. And then, the proposal features are taken into two fully connected layers respectively to generate feature matrices \( x^{cls}, x^{dec} \in \mathbb{R}^{N \times C} \). We apply softmax functions for categories and proposals respectively:

\[
\begin{align*}
[\sigma^{cls}(x^{cls})]_{ij} &= \frac{x^{cls}_{ij}}{\sum_{k=1}^{C} x^{cls}_{ik}}, \\
[\sigma^{dec}(x^{dec})]_{ij} &= \frac{x^{dec}_{ij}}{\sum_{k=1}^{N} x^{dec}_{kj}},
\end{align*}
\]

(1)

where \( [\sigma^{cls}(x^{cls})]_{ij} \) represents the \( i^\text{th} \) class label for the \( i^\text{th} \) region, and \( [\sigma^{dec}(x^{dec})]_{ij} \) represents the learned weight for the \( j^\text{th} \) class of the \( i^\text{th} \) region. Afterwards, the basic MIL network generates proposal scores for all candidate regions, \( x^{cls} = \sigma^{cls}(x^{cls}) \odot \sigma^{dec}(x^{dec}) \).

In order to integrate the CPE module (referring to Sec. III-B and III-C) into the overall framework, we make some changes to the basic structure of MIL. In detail, the CPE module generates the contrastive contextual semantics \( \mathcal{N} (\mathcal{N} \in \mathbb{R}^{N \times C}) \). Then, we merge \( \mathcal{N} \) with \( x^{cls} \) and \( x^{dec} \) to generate new classification vectors and detection vectors, whose dimensions are consistent with that of \( x^{cls} \) and \( x^{dec} \). The formula is as follows:

\[
\begin{align*}
x^{cls} &= \text{Linear}([x^{cls}, \mathcal{N}]), x^{cls} \in \mathbb{R}^{N \times C}, \\
x^{dec} &= \text{Linear}([x^{dec}, \mathcal{N}]), x^{dec} \in \mathbb{R}^{N \times C}.
\end{align*}
\]

**Linear** represents a single linear layer, which is to reduce the feature loss caused by the fusion of \( \mathcal{N} \) and \( x^{cls}, x^{dec} \). We use the similar calculation method as MIL, and via \( x^{rels}, x^{rdec}, \) generate scores \( x = \sigma^{dec}(x^{rdec}) \odot \sigma^{cls}(x^{rels}) \) for all proposals. Finally, by summing the scores of all instances for category c, we can obtain the scores of category c in the image:
\[ \sigma^c = \sum_{i=1}^{N} x_i^c, \]

and we optimize the multi-class across entropy loss in Eq. 3 to train the basic WSDDN.

\[ L_W = - \sum_{c=1}^{C} y_c \log \sigma^c + (1 - y_c) \log(1 - \sigma^c). \tag{3} \]

Furthermore, we introduce multi-layer online instance classifier refinement branches in [6] to obtain a more reasonable instance classifier. The output of refinement branches \( \phi^k \in \mathbb{R}^{N \times (C+1)} \) are different from the basic instance classifier, where \((C+1)\) represents \(C\) object classes and an additional background class, \(k\) represents the \(k^{th}\) instance classification optimizer. Based on the output \( \sigma \) of the previous basic MIL instance classifier, a more reasonable instance score can be obtained by using the online classifier branch. Assuming that the image contains class \(c\), the \(k^{th}\) instance classification optimizer will select the instance \(B_i^k\) with highest score in the proposal region of class \(c\), and then calculates the IoU formed by surrounding areas \(B\) and \(B_i^k\). When the filter is larger than the threshold \(\tau\), the region is marked as class \(c\), otherwise, it will be marked as background class. The loss function of a single instance classifier is:

\[ L_r^k = - \frac{1}{|N|} \sum_{r=1}^{N} \sum_{c=1}^{C+1} y^k_{cr} \log \phi^k_{cr}, \tag{4} \]

where \(y^k_{cr}\) represents the \(c^{th}\) class label for the \(r^{th}\) proposal in the \(k^{th}\) instance classification optimizer, and \(\phi^k_{cr}\) represents the score for the \(c^{th}\) class of the \(r^{th}\) proposal in the \(k^{th}\) instance classification optimizer.

### B. Proposal Extension

In our method, D-CPE module is in charge of extracting the contrastive contextual semantics between the extended instance and the initial one. As shown in Fig. 3, each D-CPE contains two RoI pooling layers, two LSTM encoders, and corresponding decoders. The LSTM encoder encodes proposal features to obtain corresponding semantic information, and the decoder is composed of two dual-stream branch networks to constrain the coding space of the LSTM encoder.

First, the initial region \(B_i(x_i, y_i, w_i, h_i)\) will be used to obtain the extended proposal, where \(x_i, y_i\) are the initial left-top coordinates of the \(i^{th}\) proposal region in the image \(I(W^I, H^I)\), \(w_i, h_i\) are the width and height of the proposal region, \(W^I, H^I\) are the width and height of the image. We respectively expand the boundaries in the four directions \(D = \{R2L, L2R, T2B, B2T\}\) of the initial region to form extended regions \(B_L, B_R, B_B, B_T\). \(R, L, T\) and \(B\) are the abbreviations of right, left, top and bottom directions respectively. \(B_L\) represents the extended region formed by keeping the right border of the initial region unchanged and extending the left border of the initial region along the left direction. \(B_L, B_R, B_T, B_B\) have similar meanings, but extend in different directions, and eventually formulate the following:

\[
\begin{align*}
B_L & (\max(x_i - w_i', 0), y_i, w_i + \min(w_i', x_i), h_i), \\
B_R & (x_i, y_i, \min(W^I - x_i + 1, w_i' + w_i), h_i), \\
B_T & (x_i, \max(y_i - h_i', 0), w_i, h_i + \min(h_i', y_i)), \\
B_B & (x_i, y_i, w_i, \min(H^I - y_i + 1, h_i' + h_i)),
\end{align*} \tag{5}
\]
where \( w_t' = \frac{(w_t)}{(r_t)\times 2} \), \( h_t' = \frac{(h_t)}{(r_t)\times 2} \), \( t \) is a hyper-parameter and \( t > 1 \). They can be abbreviated as \( B_L = (B : B_L) \), \( B_R = (B : B_R) \), \( B_T = (B : B_T) \), \( B_B = (B : B_B) \), where \( B_L \), \( B_R \), \( B_T \), \( B_B \) are the extra areas after the extension except the initial region. In the entire training process, the extension strategy will only be performed once on the initial proposal to obtain the extended ones. \( x_i \), \( y_i \) represent the left-top coordinates of the proposal \( i \). \( w_i, h_i \) represent the width and height of this proposal. \( w'_i, h'_i \) are the extended width and height according to the aspect ratio of the proposal \( i \). \( W'_i, H'_i \) represent the width and height of the image. Taking the extension in R2L direction as an example, when the left boundary of the proposal exceeds the image, that is, when \( (x_i - w'_i) < 0 \), and the actual extension width is \( x_i \), the left boundary of the image is used as the extended border, otherwise, \( x_i - w'_i \) is used as its left border, and the actual extension width is \( w'_i \). The extended borders in other directions are calculated in a similar way.

C. Contrastive Contextual Estimation

Extended proposals in different directions are taken into different D-CPE respectively with initial proposals. Then, these proposals and the convolutional features of the image are taken into the RoI pooling layers in D-CPE to be further processed. The outputs of the RoI pooling layer are respectively sent to the corresponding LSTM encoders to obtain semantic information.

To explore along a certain direction gradually, each adjacent column or row in the pooling feature is regarded as the corresponding input according to a certain order, and is used to extract the semantic differences to judge the integrity of the initial region. Inspired by other work encoding image features using RNN [39], [43], [44], we choose the Long Short-Term Memory (LSTM) network [8] with the best performance in our experiments as the basic component of CPE.

The semantic information generated by different encoders will be fed into a linear layer with sigmoid activation function to obtain the corresponding semantic scores. The contrastive semantic score will be merged with the features generated by the basic dual-branch structure network to adjust score of the initial proposal. Incomplete proposals will gain lower score, and relatively complete proposals will obtain higher score. The scores of all the proposals will be optimized by fusing the contrastive semantic score and the score estimated by the original WSDDN. The specific process is described as follows.

We extend the initial proposal in different directions to obtain additional extended ones. The features which are input to the CPE need to be processed additionally. We perform channel average pooling on the extracted features and generate the new features \( X^F(m, n) = \frac{1}{N} \sum_{i=1}^{C} X^i_{m, n} \), where \( X^i_{m, n} \) is the feature value at the position \((m, n)\) in the feature map extracted by the \(i^{th}\) channel, \( i \in \{1, 2, \ldots, C\} \), and \( C \) is the number of feature channels.

The four D-CPEs are CPE\(_L\), CPE\(_R\), CPE\(_B\), CPE\(_T\) respectively. The composition of these four modules are exactly the same, and we take CPE\(_L\) as an example for illustration. Given the initial proposal region \( B \) and the extended region \( B_L \), the feature \( X^F \) are used as the input of two RoI Pooling layers with different parameters for the initial region proposal and the extended one.

\[
X^B = \text{RoI Pooling}^1(X^F; B), \quad X^{B_L} = \text{RoI Pooling}^2(X^F; B_L),
\]

where \( X^B \in \mathbb{R}^{N \times H^f \times W_1^f} \), \( X^{B_L} \in \mathbb{R}^{N \times H^f \times W_1^f} \), \( N \) is the number of proposals, \( H^f \) is the height of the pooling features, and \( W_1^f, W_2^f \) are the width of the pooling features, \( W_2^f = W_1^f + w \). \( W_1^f \) represents the width of pooling features of the initial region, \( w \) represents the width of pooling features of the extra extended region. Due to the inherent characteristics of RoI pooling, we can obtain the features of the extended proposal by concatenating \( X^B \) and \( X^{B_L} \):

\[
X^{B_L} = X^B : X^{B_L},
\]

where \( X^{B_L} \in \mathbb{R}^{N \times H^f \times W_2^f} \).
be regarded as the input of LSTM in specific direction $D$. The data in each row or column in the feature matrix will be regarded as the input of the two encoders. As shown in Fig. 4, L2R needs to perform a transpose operation, while R2L needs to perform a horizontal flip operation additionally.

After above operations, these vectors will be fed into LSTM encoders and perform sequential coding. For the CPE module, we still take the module CPE$_L$ as an example to demonstrate D-CPE, which is shown in Fig. 3. $X_r^{Bi}$ and $X_{r2l}^{Bi}$ are respectively fed into the two LSTM encoders in CPE$_L$, where $i = \{1, 2, \ldots, W_1\}$, $j = \{1, 2, \ldots, W_2^f\}$, $W_1 < W_2^f$. $X_r^{Bi}$ and $X_{r2l}^{Bi}$ can be regarded as time sequences with $W_1^f$, $W_2^f$ steps, respectively.

$$D^B = LSTM(X_r^{Bi}, W_1^f),$$

$$D^{BL} = LSTM(X_{r2l}^{Bi}, W_2^f),$$

(8)

where $W_1^f$, $W_2^f$ represent the different step sizes of the LSTM encoder, $W_2^f > W_1^f$. According to the characteristics of LSTM that the output of previous time step is used as the input of the next one, we further derive the vectors $D^B \in \mathbb{R}^{N \times M}$, $D^{BL} \in \mathbb{R}^{N \times M}$. $M$ is a hyper-parameter, then we can infer:

$$D^{BL} = LSTM(X_{r2l}^{Bi}, W_2^f),$$

$$= LSTM(X_{r2l}^{Bi}, X_{r2l}^{Bi}, W_1^f + w),$$

(9)

where $W_1^f$ represents the first $W_1^f$ column of $X_{r2l}^{Bi}$, and $w$ represents the last $w$ column, so $X_{r2l}^{Bi}$ can be regarded as the input of the previous $W_1^f$ step, and $X_{r2l}^{Bi}$ be regarded as the input of the next $w$ step.

According to Eq. 9, we can infer that there is a relationship between $D^B$ and $D^{BL}$. To further extract the difference, we use a separate linear layer and a sigmoid layer to extract contextual semantics of proposals for each LSTM encoder. Moreover, we simply apply an attention mechanism [45] to better encode global information and extract complete semantic information. Then, we use the initial proposal contextual semantics score $S^B$ and the extended proposal contextual semantics score $S^{BL}$ to construct $N^L = |S^B - S^{BL}|$ for each instance and the corresponding extended proposal. As shown in the Fig. 5, the three columns from left to right represent the initial proposal score matrix, extended proposal score matrix and contrastive contextual semantic score matrix with the size of $N \times C$. Each row in the matrix represents the corresponding scores of each category. After obtaining the integrity scores of the initial proposal and the extended one, we perform element-wise subtraction operations on the two score matrices and absolute value calculations to obtain a preliminary contrastive semantic score. We further improve $N^L$:

$$N^L = \frac{|S^B - S^{BL}| - \min(|S^B - S^{BL}|)}{\max(|S^B - S^{BL}|) - \min(|S^B - S^{BL}|)}.$$  

(10)

Subsequently, we combine the contrastive scores obtained by all D-CPEs by the following formula:

$$N = \alpha(N^L + N^R) + (1-\alpha)(N^B + N^T),$$

(11)

where $\alpha \in (0,1)$, $\alpha$ is a hyper-parameter. The semantic difference between the output of the two encoders in each D-CPE is mainly caused by the extension part $B_i$. When the semantic information of the initial proposal and extended one is quite different, the value of $N^L$ will be large. $B_i$ may contain objects or backgrounds that are significantly different from $B$, and may not contain other components of the object in the initial proposal. That means $B$ already contains the complete object. On the contrary, when the semantic information of the initial proposal and extended one is similar, the value of $N^L$ will be small. $B_i$ may be similar to $B$. The semantic information of the extended proposal is similar to the initial one, $B_i$ may contain other components of the object and the object in the initial proposal is incomplete.

To constrain the LSTM encoder, we add a small WSDDN dual-stream branch decoder to each LSTM encoder. Similar to...
the WSDDN network, the detection branch and the classification branch respectively include a linear layer and a softmax layer in different directions. The loss function of CPE during the training process is defined as follows:

\[ L_{\text{CPE}}(L) = L_W^1(L) + L_W^2(L) \]  

\[ L_W^1, L_W^2 \] are calculated in the same way as Eq.3. Similarly, we perform the same calculation on other modules, the loss function of the complete CPE module is

\[ L_{\text{CPE}} = (L_{\text{CPE}}(L) + L_{\text{CPE}}(R) + L_{\text{CPE}}(B) + L_{\text{CPE}}(T))/4. \]

Meanwhile, in order to constrain the weakly supervised object detector, we integrate all the loss functions, and the final loss function of the entire model is as follows:

\[ L = L_{\text{CPE}} + L_W + \sum_{k=1}^{3} L_{r}^k, \]

IV. Experiments

In this section, we first describe the datasets, evaluation metrics, and the implementation details. Then, we conduct ablation studies to evaluate the influence of different modules or parameters on the results. Finally, we compare our proposed method with other state-of-the-arts on WSOD task to verify its effectiveness.

A. Datasets and Evaluation Metrics

We evaluate our method on the PASCAL VOC 2007 and 2012 [46], and the MS-COCO 2014 [47] datasets. We only utilize image-level labels during model training. PASCAL VOC 2007 and 2012 datasets have 9,962 and 22,531 images. We choose the \textit{trainval} set (5,011 images for 2007 and 11,540 images for 2012) to train our model and choose \textit{test} set (4,951 images for 2007 and 10,991 images for 2012) to evaluate our model. For testing, we adopt two evaluation metrics, the mean of Average Precision (mAP) [48] on the \textit{test} set, and the correct localization (CorLoc) [49] on the \textit{trainval} set. Both these two metrics follow the standard PASCAL VOC protocol. IoU>0.5 between ground truth and predicted boxes. The MS-COCO dataset contains 80 categories, we choose the \textit{train}2014 set (about 80K images) for training, \textit{val}2014 set (about 40K images) and \textit{minival} set (5K images) for testing. We report AP@.50 and AP@[.50: .05: .95] on \textit{val}2014 and \textit{minival}.

B. Implementation Details

Our framework is based on VGG16 network [50] pretrained on ImageNet. In detail, following the work in [7], we apply Selective Search (SS) [51] and MCG [52] which generate about 2,000 proposals per image for the PASCAL VOC and MS-COCO datasets, respectively. We refine the instance classifier three times, i.e., \( K = 3 \) and \( \tau = 0.1 \) in Sec. III-A according to [6]. During training, the mini-batch size for SGD is set to 1, the momentum and weight decay are all set to 0.9, the learning rate is set to 5e-4 for the first 25k, 40k, 320k iterations and then decreased to 5e-5 in the following 35k, 70k, 480k iterations on VOC 2007, VOC 2012, MS-COCO respectively. We set the hyper-parameter \( t = 4, \alpha = 0.5 \). Specifically, we adopt five image scales \{480, 576, 688, 864, 1200\} and horizontal flipping randomly on the images to generate 10 groups of augmented data. NMS = 0.3 (with 30\% IoU threshold) is applied to compute the AP and CorLoc.

The experiments are implemented on the Pytorch deep learning framework, and part of the code is written in C++. All of our experiments are running on a workstation with an Intel(R) Xeon(R) CPU E5-2698 v4(2.20GHz) and 4 NVIDIA Tesla V100 GPUs.

C. Ablation Studies

We conduct some ablation studies to validate the effectiveness of our method, including the influence of different directional extensions of CPE, various components of CPE, different decoders of D-CPE, the coupling coefficient of D-CPE, and extension coefficient of the extended proposal.

1) Different Directional Extensions: We conduct ablation studies on different directional extensions in CPE module. As shown in Tab. 1, in row 1, 2, 3, 4, we discuss the effects in different directional extensions. Among these four extensions, although the result on one single direction is similar, the performance in different categories is different. The APs of \textit{person} category on L2R, R2L, T2B, B2T are 19.8\%, 15.4\%, 17.1\%, 19.4\%, respectively. The \textit{person} category is usually manifested in a standing posture in the scene, leading to the extension in the vertical direction, which is better than that in the horizontal direction. Meanwhile, there is a large difference in the extension of \textit{dog} category in two vertical directions. The reason is, compared with other animals, the limbs in the running and standing postures of \textit{dog} occupy most of the object region. In the extension along the B2T direction, the proportion of the object such as \textit{aeroplane}, \textit{chair}, \textit{dog}, \textit{motorbike}, is much smaller than that of the background, resulting in the extension along the T2B direction far better than that of B2T.

Moreover, taking the combination of T2B and B2T as an example, Tab. 1 shows that the overall result of T2B+B2T is not higher than that of B2T, but nearly equal. After analyzing 20 categories, we find that the results of \textit{bike}, \textit{boat}, \textit{bottle}, \textit{car}, \textit{table}, \textit{person}, \textit{plant}, \textit{sofa}, \textit{train}, \textit{tv} are not higher than that of B2T after combining the expansion in the opposite direction T2B+B2T. The reason is that more discriminative features of these categories are distributed in the lower part of the overall object area. When the proposal expands from the bottom to top direction (B2T), the extended proposal has covered most of the object, and its lower boundary is close to the object boundary. After combining the extensions in the opposite direction T2B, the lower boundary of the extended proposal may go beyond the object boundary and cover areas that do not belong to the object, causing to generate a lower score. Therefore, the overall result of T2B+B2T is a little lower. To be noted that, in some initial proposals, the usage of the extension in the opposite direction will conflict with the integrity evaluation of the initial proposal and cause the detector to make an unreasonable evaluation. After introducing more extension directions, this problem will be alleviated to
In this section, we report the influence of different D-CPE components. As shown in Tab. II, rows 1, 2, 3 respectively show the results with “Base”, “Base+A” and “Base+D”. Row 4 shows the results of “Base+A+D”. By comparing the results in row 1 and row 2, the mAP of “Base+A” is worse than that of “Base” because the additional attention mechanism allows the LSTM encoder to select more critical information, but it may also ignore some objects in the current scene. By increasing the decoder based on dual-stream structure network to constrain the coding space of LSTM encoder, better results can be obtained than that of “Base”. The reason is that the decoder effectively restrains the encoding space, so that it can choose as many objects as possible in the scene. When “Base+A+D” is used in our method, we have obtained the best result due to the combination of LSTM network with self-attention mechanism and decoder, because the attention mechanism will capture the key information in the scene as much as possible, and the decoder requires the encoder to capture more complete scene information to include all objects in the scene as much as possible. To more intuitively observe different results generated by these components, we show some sample results in Fig. 6. In the first four columns, we can see that in “Base” cases, the model may treat a single object as two objects (row 2) or the detector regards the background a certain extent. In general, different directional extensions of CPE module will produce various results. The experiments also show that the more extension direction involved, the more complete comparative semantics extracted by the CPE, and the evaluation of the initial instance will be more accurate. Moreover, by introducing the aspect ratio of the proposal to adjust different scales of the extension operation dynamically, we can obtain more accurate extended instances to have better results.

2) Different Components of D-CPE: In this section, we report the influence of different D-CPE components. As shown in Tab. II, rows 1, 2, 3 respectively show the results with “Base”, “Base+A” and “Base+D”. Row 4 shows the results of “Base+A+D”. By comparing the results in row 1 and row 2, the mAP of “Base+A” is worse than that of “Base” because the additional attention mechanism allows the LSTM encoder to select more critical information, but it may also ignore some objects in the current scene. By increasing the decoder based on dual-stream structure network to constrain the coding space of LSTM encoder, better results can be obtained than that of “Base”. The reason is that the decoder effectively restrains the encoding space, so that it can choose as many objects as possible in the scene. When “Base+A+D” is used in our method, we have obtained the best result due to the combination of LSTM network with self-attention mechanism and decoder, because the attention mechanism will capture the key information in the scene as much as possible, and the decoder requires the encoder to capture more complete scene information to include all objects in the scene as much as possible. To more intuitively observe different results generated by these components, we show some sample results in Fig. 6. In the first four columns, we can see that in “Base” cases, the model may treat a single object as two objects (row 1) or the detector regards the background as one object (row 2). After adding decoders on “Base” to constrain the LSTM encoder, although the case where different positions of the object are contained by multiple boxes can be reduced, there may be a phenomenon that the detector generates one box for multiple similar objects (row 4) and can not detect all objects in the scene (row 6). Similar to the case of “Base+D”, in the case of “Base+A”, some objects in the scene can not be detected in row 2, 3, but get tighter boxes in row 5. In the case of “Base+A+D”, the situations mentioned above can be alleviated, and more reasonable boxes can be obtained. However, there are still some boxes that do not match the actual object or multiple boxes that are generated for a single object. The last column shows the results generated by our model trained with Fast R-CNN.

3) Different Decoders of D-CPE: Fig. 7 shows that the effect of different decoders in D-CPEs. “S-Decoder” represents the decoder of the linear layer using sigmoid as the activation function. “T-Decoder” represents the dual-stream branch network is used with softmax as the activation function. Compared to the case which constrains the LSTM encoder using “S-Decoder”, the usage of “T-Decoder” can obtain better results in more cases. In individual categories, “T-Decoder” has also shown great advantages, especially in cat, dog and person categories, the result with “T-Decoder” is at least 13 percent higher than that with “S-Decoder”. And in aeroplane, bird, boat, bottle, cow and table categories, it is about 3 to 8 percent higher. The results by these two decoders are similar in bike, bus, plant, sheep, and train categories, with the difference of less than 3 percent. In the car, chair and sofa categories, the “S-Decoder” performs better, especially on tv, which is higher than the “T-Decoder” by 8 percent. The performance of these two decoders is quite different in the cat, dog, person, and tv categories. The reason is that cat, dog, and person often appear together with other objects, while the tv category often appears alone in the scene. In different scenarios, these two decoders have different performances, especially when the scene contains multiple detection objects. The “T-Decoder” can more widely select the instances that contain the object, therefore, it can add more comprehensive constraints on the encoder. In the monotonous object scenario, “S-Decoder” can effectively reduce the noise in the encoder to generate more reasonable contrastive semantics of initial proposals and extended proposals.

4) Coupling Coefficient of D-CPE: Tab. III shows the influence of different coefficient $\alpha$ on the different categories. On the whole, mAP is at a peak when $\alpha = 0.5$, and at the
Fig. 6. Visualization results of D-CPE with different components. “Base” represents the case only with the LSTM encoder, “D” represents the case that additional decoder is added, “A” represents the case that the attention mechanism is added to the LSTM encoder, “FR” represents the usage of pseudo-labels for box regression.

TABLE III

RESULTS WITH DIFFERENT $\alpha$ ON PASCAL VOC 2007 DATASET

| $\alpha$ | aeroplane | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv | mAP |
|----------|------------|------|------|------|--------|-----|-----|-----|-------|-----|-------|-----|-------|-------|---------|-------|-------|------|-------|----|-----|
| 0.1      | 66.4       | 78.4 | 59.3 | 34.2 | 34.1   | 74.1| 70.2| 74.2| 31.5  | 63.4| 43.8  | 60.7| 62.2 | 70.1  | 14.4    | 33.4 | 46.4 | 62.0 | 73.8  | 62.2| 55.7 |
| 0.2      | 65.7       | 76.4 | 58.8 | 35.9 | 35.2   | 73.0| 69.5| 75.8| 29.4  | 62.6| 46.6  | 63.3| 63.9 | 70.1  | 13.9    | 31.5 | 42.8 | 60.1 | 73.1  | 68.4| 55.8 |
| 0.3      | 63.9       | 76.8 | 58.2 | 33.2 | 33.4   | 73.7| 71.0| 74.4| 29.9  | 62.6| 45.1  | 62.3| 62.3 | 69.8  | 16.3    | 32.0 | 47.2 | 62.9 | 75.2  | 58.2| 55.5 |
| 0.4      | 61.9       | 78.3 | 57.1 | 36.1 | 36.4   | 74.5| 65.0| 74.6| 28.5  | 61.3| 46.8  | 58.7| 60.5 | 70.0  | 15.9    | 32.3 | 43.1 | 59.5 | 74.7  | 68.8| 55.2 |
| 0.5      | 62.4       | 76.4 | 59.7 | 33.8 | 28.7   | 71.7| 66.1| 72.2| 33.9  | 67.7| 47.6  | 67.2| 60.0 | 71.7  | 18.1    | 29.9 | 53.8 | 58.9 | 74.3  | 64.1| 55.9 |
| 0.6      | 67.0       | 76.8 | 56.9 | 34.5 | 31.3   | 73.7| 69.4| 70.9| 29.4  | 61.2| 47.7  | 64.5| 65.1 | 70.7  | 13.6    | 32.1 | 42.8 | 58.8 | 72.6  | 58.1| 54.9 |
| 0.7      | 66.5       | 76.7 | 55.6 | 32.5 | 34.5   | 75.3| 68.3| 72.4| 28.1  | 61.6| 47.3  | 67.0| 61.5 | 69.5  | 17.4    | 31.9 | 44.5 | 58.9 | 73.8  | 58.2| 55.1 |
| 0.8      | 65.1       | 77.8 | 56.7 | 33.1 | 33.4   | 74.3| 67.5| 73.9| 30.9  | 64.9| 46.6  | 66.0| 63.0 | 69.9  | 16.1    | 30.8 | 41.7 | 61.0 | 71.6  | 60.7| 55.3 |
| 0.9      | 63.3       | 78.8 | 56.5 | 33.3 | 33.4   | 73.8| 66.9| 71.1| 29.5  | 65.5| 48.2  | 65.0| 60.5 | 70.4  | 15.2    | 33.0 | 41.8 | 58.3 | 74.6  | 60.5| 55.0 |

lowest point when $\alpha = 0.6$. With different $\alpha$, aeroplane, bottle, car, cow, dog, horse, sheep and tv categories will have large changes, which can reach up to 5 percent. The biggest difference is tv, when $\alpha = 0.4$, AP = 68.8%, while $\alpha = 0.6$, AP = 58.1%. When the $\alpha$ changes, the APs with smaller changes are bike and motorbike, the AP gap between their maximum and minimum is only 2.4% and 2.2%. In most categories, the change of AP in the adjacent interval
is relatively small with the increase of $\alpha$. This proves that different categories are sensitive to the hyper-parameter $\alpha$, which is caused by the aspect ratios of different categories in the scene.

5) Extension Coefficient of Extended Proposal: We further investigate the effectiveness of extension coefficient. In Fig. 8 and 9, we show the mAP and CorLoc under different extension coefficient $t$ on PASCAL VOC 2007. Obviously, we can observe when the extension coefficient $t$ increases from 2 to 4, the mAP and CorLoc become better. When $t$ is in the range of 4 to 5, CorLoc is slowly decreasing, and mAP is also decreasing. As the extension coefficient $t$ changes, the overall results of certain categories do not change much, such as plant category. The best AP is 31.8% ($t = 6$), while the worst is 28.4% ($t = 3$). Some categories may show huge differences, the highest AP of tv category is 69.1% ($t = 6$), but the worst is 43.8% ($t = 1$), the gap between them is about 26.0%. The APs of other categories fluctuate continuously with the change of $t$. Different extension coefficients will cause large changes of the mAP on some categories. The reason is that if fixed coefficients are used to expand the proposal, different aspect ratios of the initial proposal will not be taken into account. Therefore, we further design an adaptive aspect ratio coefficient in Eq. 5 to reduce this negative impact, resulting in an increase in mAP. As shown in Fig. 10, we show the influence of the extension coefficient $t$ on different animal categories. With different $t$, the APs of dog, horse and sheep categories have significant changes, while the bird, cat and cow have not changed too much. The sensitivities of different categories to $t$ are due to the fact that the initial proposals of different categories have different aspect ratios in the image.

D. Comparison With the State-of-the-Arts

We compare the proposed method with other state-of-the-arts, including MIST [60], PG-PS [35], SLV [26] and CASD [58], as well as other approaches utilizing segmentation for WSOD. Following previous WSOD work, our proposed method is also compared in two settings. The first one is to directly use the generated results without bounding-box regression. The second setting is to use the generated boxes by our method as pseudo ground-truths to train Fast R-CNN detector 2015Fast. Tab. IV and Tab. V show the mAP and CorLoc performance respectively on Pascal VOC 2007 dataset. The top part shows the results obtained by training our model with the first setting. Among them, our method obtains the excellent results. Compared with the mAP of baseline (OICR), our method improves by 14.7 percent. In the bottom part of the table, we show the results with the second setting. Compared to these methods trained with Fast R-CNN, our model still get the best performance i.e., 57.3% mAP among all the methods. Our method outperforms other powerful methods, MIST+FRCNN, PG-PS+FRCNN, P-MIDN+MGSC+FRCNN, by 2.4 percent, 2.7 percent and 2.3 percent in mAP respectively. In Tab. VI, we show the results on Pascal VOC 2012. Our method reaches mAP$_{0.5}$ of 54.6%, which is 12.1 percent higher than that of baseline in mAP and exceeds other results of the state-of-the-arts. To further demonstrate the effectiveness of the proposed method, we conducted experiments on the more challenging MS-COCO dataset and report results in Tab. VII. Our proposed method achieves AP@[.5: .05: .95] of 13.8%, surpassing other methods by a large margin on the val2014 split. This shows our method can still achieve excellent results on large-scale datasets.

We also have to admit that there is a certain gap between the proposed method and other methods in Tab. V and Tab. VI in the whole CorLoc. In the proposed method, as the region is gradually approaching complete, the instance may contain more general features which may belong to multiple categories. Some instances are similar to certain objects but belong to other categories. However, the proposed method recognizes them as the same class of objects, which leads to wrong judgments and increases the number of false positives (FP). At the same time, since the instance already contains regions with discriminative features, it increases the number of true positives (TP) and reduces the number of false negatives (FN). For different metrics, CorLoc only considers Precision.
but the mAP considers both Precision and Recall. Therefore, our mAP is higher than that of other WSOD methods, while the CorLoc score is not the best, but still competitive.

**E. Visualization Results**

In order to better analyze the detection results, we visualize some example images from the PASCAL VOC 2007 dataset.
In Fig. 11, we show the successful detection (IOU > 0.5), our method can obtain the detection results which are close to the ground-truth, and can detect the main object in most scenes. It is noted that our method can alleviate most of the mentioned problems and improve detection performance. We further show some failed detection results in Fig. 12. For example, motorbike and tv in the second column, although the scene contains multiple similar objects, only a single detection box was generated. Similarly, in the first column, the detection box for the sheep contains multiple sheep objects. The reason is that the same object in the scene is too compact, and at the same time, the extension is relatively large, which makes the encoder ignore the background area of the extension. Besides, the detection boxes generated by the model are too small to contain the complete object. In the first column, the detection box for the boat is too small. In addition, the detection of the dog as a bird in the first row shows the problem of wrong detection, and the last column in the second row shows the problem of confusing the background as the object. Compared with the CNN network, although we introduce the decoder to constrain the encoding space of the LSTM network, it still does not perform very well. And the background of images in the training data often contain some objects similar to the detection objects, which will also affect the detection accuracy and can be seen in the last column of the second row.
want to explore the extension scales of different categories of degradation of detection performance. In addition, we will also use CPE to recognize them as one object, resulting in the detection results. MS-COCO datasets. Most advanced results on the PASCAL VOC 2007, 2012 and 2014 frameworks have obtained more accurate localization results, while our proposed network can provide a new way to optimize proposal scores. Our proposed network framework has obtained more accurate localization results, significantly improved the baseline method, and achieved the best results on the PASCAL VOC 2007, 2012 and MS-COCO datasets.

However, there are still some problems that need to be solved in the future. For example, when similar objects are enclosed together or the distance of them is too close, the CPE may recognize them as the one object, resulting in the degradation of detection performance. In addition, we also want to explore the extension scales of different categories in different directions delicately, including more expansion methods and the combination of contrastive semantics in different. Furthermore, we will explore better dynamic methods to obtain extended proposals to reduce the impact of different pooling features of these proposals on extracting contrastive semantics of the integrity.

V. CONCLUSION

In this paper, we propose an effective WSOD method by combining the sequential encoder LSTM. We regard the initial instance and the extended one as a mutual relational network, and regard the proposal features generated by the instances as time sequences. CPE extracts the contrastive contextual semantics in different directions between the initial proposal and the extended proposal by LSTM encoders, and combines the contrastive contextual semantics with the basic WSDDN network to produce more reasonable initial instances to perform the task of WSOD. By comparing the initial proposal and the extended proposal, CPE provides a new way to optimize proposal scores. Our proposed network framework has obtained more accurate localization results, significantly improved the baseline method, and achieved the most advanced results on the PASCAL VOC 2007, 2012 and MS-COCO datasets.

REFERENCES

[1] F. Sun, T. Kong, W. Huang, C. Tan, B. Fang, and H. Liu, “Feature pyramid reconfiguration with consistent loss for object detection,” IEEE Trans. Image Process., vol. 28, no. 10, pp. 5041–5051, May 2019.
[2] Y. Zhu, C. Zhao, H. Guo, J. Wang, X. Zhao, and H. Lu, “Attention couplednet: Fully convolutional attention coupling network for object detection,” IEEE Trans. Image Process., vol. 28, no. 1, pp. 113–126, Jan. 2019.
[3] S. Deng et al., “A global-local self-adaptive network for drone-view object detection,” IEEE Trans. Image Process., vol. 30, pp. 1556–1569, 2021.
[4] Z.-M. Chen, X. Jin, B.-R. Zhao, X. Zhang, and Y. Guo, “HCE: Hierarchical context embedding for region-based object detection,” IEEE Trans. Image Process., vol. 30, pp. 6917–6929, 2021.
[5] H. Bilen and A. Vedaldi, “Weakly supervised deep detection networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Dec. 2016, pp. 2846–2854.
[6] P. Tang, X. Wang, X. Bai, and W. Liu, “Multiple instance detection network with online instance classifier refinement,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 3059–3067.
[7] P. Tang et al., “PCL: Proposal cluster learning for weakly supervised object detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 1, pp. 176–191, Jan. 2020.
[8] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.
[9] Y. Shen, R. Ji, Y. Wang, Y. Wu, and L. Cao, “Cyclic guidance for weakly supervised joint detection and segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 697–707.
[10] J. Li et al., “Attentive contexts for object detection,” IEEE Trans. Multimedia, vol. 19, no. 5, pp. 944–954, May 2017.
[11] H. Bilen, M. Pedersoli, and T. Tuytelaars, “Weakly supervised object detection with convex clustering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1081–1089.
[12] C. Ge, J. Wang, Q. Qi, H. Sun, and J. Liao, “Fewer is more: Image segmentation based weakly supervised object detection with partial aggregation,” in Proc. BMVC, 2018, p. 136.
[13] R. Gokberk Cinbis, J. Verbeek, and C. Schmid, “Multi-fold MIL training for weakly supervised object localization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 2409–2416.
[14] M. Shi and V. Ferrari, “Weakly supervised object localization using size estimates,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 105–121.
[15] C. Wang, W. Ren, K. Huang, and T. Tan, “Weakly supervised object localization with latent category learning,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2014, pp. 431–445.
[16] X. Zhang, Y. Yang, and J. Feng, “Ml-LocNet: Improving object localization,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1224–1232.
[17] V. Kantorov, M. Oquab, M. Cho, and I. Laptev, “ContextLocNet: Context-aware deep network models for weakly supervised localization,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 350–365.
[18] E. W. Teh, M. Rochan, and Y. Wang, “Attention networks for weakly supervised object localization,” in Proc. BMVC, 2016, pp. 1–11.
[19] Y. Shen, R. Ji, K. Yang, C. Deng, and C. Wang, “Category-aware spatial constraint for weakly supervised detection,” IEEE Trans. Image Process., vol. 29, pp. 843–858, 2020.
[20] Q. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang, “Weakly supervised object localization with progressive domain adaptation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 3512–3520.
[21] F. Wan, P. Wei, J. Jiao, Z. Han, and Q. Ye, “Min-entropy latent model for weakly supervised object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 1297–1306.
[22] X. Zhang, Y. Yang, and J. Feng, “MI-LocNet: Improving object localization with multi-view learning network,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 240–255.
[23] Y. Shen, R. Ji, S. Zhang, W. Zuo, and Y. Wang, “Generative adversarial learning towards fast weakly supervised detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 5764–5773.
[24] F. Wan, C. Liu, W. Ke, X. Ji, J. Jiao, and Q. Ye, “C-MIL: Continuation multiple instance learning for weakly supervised object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 2194–2203.
[25] A. Arun, C. V. Jawahar, and M. Pawan Kumar, “Dissimilarity coefficient based weakly supervised object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 9424–9433.
[26] Z. Chen, Z. Fu, R. Jiang, Y. Chen, and X.-S. Hua, “SLV: Spatial likelihood voting for weakly supervised object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 12992–13001.
[27] M. Yang, Y. Li, P. Hu, J. Bai, J. C. Lv, and X. Peng, “Robust multi-view clustering with incomplete information,” IEEE Trans. Pattern Anal. Mach. Intell., early access, Mar. 1, 2022, doi: 10.1109/TPAMI.2022.3155499.

Fig. 12. Failed detection results of our method. The red boxes are the results detected by our method, and blue boxes indicate the ground-truth.
[28] M. Yang, Y. Li, Z. Huang, Z. Liu, P. Hu, and X. Peng, “Partially view-aligned representation learning with noise-robust contrastive loss,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 1134–1143.

[29] Y. Li, P. Hu, Z. Liu, D. Peng, and X. Peng, “Contrastive clustering,” in Proc. AAAI Conf. Artif. Intell., 2020, pp. 8547–8555.

[30] P.-T. Jiang, L.-H. Han, Q. Hou, M.-M. Cheng, and Y. Wei, “Online attention accumulation for weakly supervised semantic segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 10, pp. 7062–7077, Oct. 2022.

[31] B. Zhang, J. Xiao, J. Jiao, Y. Wei, and Y. Zhao, “Affinity attention graph neural network for weakly supervised semantic segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 11, pp. 8082–8096, Nov. 2022.

[32] A. Diba, V. Sharma, A. Parazandeh, H. Pirsiavash, and L. Van Gool, “Weakly supervised cascaded convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 5131–5139.

[33] M. Gao, A. Li, R. Yu, V. I. Morariu, and L. S. Davis, “C-WSL: Count-guided weakly supervised localization,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 152–168.

[34] J. Ahn and S. Kwak, “Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 4981–4990.

[35] G. Cheng, J. Yang, D. Gao, L. Guo, and J. Han, “High-quality proposals for weakly supervised object detection,” IEEE Trans. Image Process., vol. 29, pp. 5794–5804, 2020.

[36] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 618–626.

[37] Z. Yan, H. Zhang, Y. Jia, T. Breuel, and Y. Yu, “Combining the best of top-down and bottom-up attention for weakly supervised semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Nov. 2019, pp. 8291–8299.

[38] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu, “Pixel recur- rent neural networks,” in Proc. Int. Conf. Mach. Learn., 2016, pp. 1747–1756.

[39] L. V. Pato, R. Negrinho, and P. Aguiar, “Seeing without looking: Contextual rescoring of object detections for ap maximization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 14610–14618.

[40] T. Chen, S. Saxena, L. Li, D. J. Fleet, and G. Hinton, “Pixon2seq: A language modeling framework for object detection,” 2021, arXiv:2109.10852.

[41] A. Vaswani et al., “Attention is all you need,” 2017, arXiv:1706.03762.

[42] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL visual object classes challenge: A retrospective,” Int. J. Comput. Vis., vol. 111, no. 1, pp. 98–136, Jan. 2014.

[43] T. Y. Lin, M. Maire, S. Belongie, J. Hays, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in Proc. European Conference Computer Vision, 2014, pp. 740–755.

[44] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal visual object classes (VOC) challenge,” Int. J. Comput. Vis., vol. 88, no. 2, pp. 303–338, Sep. 2009.

[45] T. Deselaers, B. Alexe, and V. Ferrari, “Weakly supervised localization and learning with generic knowledge,” Int. J. Comput. Vis., vol. 100, no. 3, pp. 275–293, Dec. 2012.

[46] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

[47] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” Int. J. Comput. Vis., vol. 104, no. 2, pp. 154–171, Apr. 2013.

[48] J. Pont-Tuset, P. Arbeláez, J. T. Barron, F. Marques, and J. Malik, “Multiscale combinatorial grouping for image segmentation and object proposal generation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 1, pp. 128–140, Jan. 2017.

[49] Y. Zhang, Y. Bai, M. Ding, Y. Li, and B. Ghanem, “W2F: A weakly-supervised to fully-supervised framework for object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 928–936.

[50] S. Kosugi, T. Yamasaki, and K. Aizawa, “Object-aware instance labeling for weakly supervised object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Nov. 2019, pp. 6063–6071.

[51] G. Yan et al., “C-MIDN: Coupled multiple instance detection network with segmentation guidance for weakly supervised object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Nov. 2019, pp. 9833–9842.

[52] X. Li, M. Kan, S. Shan, and X. Chen, “Weakly supervised object detection with segmentation collaboration,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Nov. 2019, pp. 9734–9743.

[53] C. Li, S. Wang, D. Xu, Y. Liu, and W. Zhang, “Object instance mining for weakly supervised object detection,” in Proc. AAAI Conf. Artif. Intell., vol. 34, no. 7, 2020, pp. 11482–11489.

[54] Z. Huang, Y. Zou, V. Bhagavatula, and D. Huang, “Comprehensive attention self-distillation for weakly-supervised object detection,” 2020, arXiv:2010.12023.

[55] Y. Xu, C. Zhou, X. Yu, B. Xiao, and Y. Yang, “Pyramidal multiple instance detection network with mask guided self-correction for weakly supervised object detection,” IEEE Trans. Image Process., vol. 30, pp. 3029–3040, 2021.

[56] X. Dong, W. Zhang, Z. Li, W. Li, and Y. Zhao, “Weakly supervised object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10595–10604.

[57] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440–1448.