Small Object Detection via Coarse-to-fine Proposal Generation and Imitation Learning

Xiang Yuan  Gong Cheng *  Kebinng Yan  Qinghua Zeng  Junwei Han
School of Automation, Northwestern Polytechnical University, Xi’an, China
{shaunyuan, kebingyan, zengqinghua}@mail.nwpu.edu.cn, {gcheng, jhan}@nwpu.edu.cn

Abstract

The past few years have witnessed the immense success of object detection, while current excellent detectors struggle on tackling size-limited instances. Concretely, the well-known challenge of low overlaps between the priors and object regions leads to a constrained sample pool for optimization, and the paucity of discriminative information further aggravates the recognition. To alleviate the aforementioned issues, we propose CFINet, a two-stage framework tailored for small object detection based on the Coarse-to-fine pipeline and Feature Imitation learning. Firstly, we introduce Coarse-to-fine RPN (CRPN) to ensure sufficient and high-quality proposals for small objects through the dynamic anchor selection strategy and cascade regression. Then, we equip the conventional detection head with a Feature Imitation (FI) branch to facilitate the region representations of size-limited instances that perplex the model in an imitation manner. Moreover, an auxiliary imitation loss following supervised contrastive learning paradigm is devised to optimize this branch. When integrated with Faster RCNN, CFINet achieves state-of-the-art performance on the large-scale small object detection benchmarks, SODA-D and SODA-A, underscoring its superiority over baseline detector and other mainstream detection approaches. Code is available at https://github.com/shaunyuan22/CFINet.

1. Introduction

Small object detection (SOD) aims to classify and localize the instances with limited regions, which plays an important role in a wide range of scenarios, such as pedestrian detection, autonomous driving, and intelligent surveillance understanding, to name a few [42, 28, 22, 19, 36, 2, 20]. Compared to the generic object detection which has been extensively studied, SOD task receives relatively little attention and good solutions are still scarce so far. Moreover, generic detectors [29, 6, 25, 31, 7, 3, 26, 5] usually struggle on handling small objects due to two inherent challenges: the insufficient and low-quality samples for training and the uncertain prediction of RoIs (Region of Interests).

First, current prevailing detectors exploit either overlap-based [29, 26] or distance-based [31] strategies to select the positive priors of objects for training. However, small instances usually occupy an extremely limited area, therefore the region overlaps between densely arranged anchors and ground truth boxes are significantly small and far from the commonly used positive IoU (Intersection-over-Union) threshold, as in Figure 1. In other words, the existing positive sample criterion is overly stringent when applied to small/tiny objects, resulting in a restricted number of samples available for optimization. An intuitive approach involves reducing the threshold for defining a positive sample [47]. However, while this can lead to an increase in the number of positive samples, it often at the expense of overall sample quality, in which low-quality samples disrupt optimization and incur a trivial regression solution. Worse still, this is actually contradictory to the purpose of proposal

Figure 1. Distribution of maximum IoU of anchors matched to each ground-truth instance in SODA-D [9] train-set, where extremely Small (eS), relatively Small (rS), and generally Small (gS) correspond to three area subsets in SODA [9] with the ranges (0, 144), (144, 400) and (400, 1024). The smaller the objects are, the lower IoU the matched anchors have, hence the commonly used positive IoU threshold (0.7) is too rigorous for small objects.

\*Corresponding author: gcheng@nwpu.edu.cn

1This paper focuses on the detection of “pure” small objects, where the scales of all the objects are distributed within a relatively tight range [45, 34, 9], which is distinct from the known small objects in multi-scale object detection [24].
The results are tested on the SODA-D [9] test-set and with a ResNet-50 [17] as the backbone.

network, i.e., guaranteeing the recall and ease the burden of subsequent work.

To sum up, current prevailing priors-to-proposals paradigms that heavily depend on the overlap or distance metric have inherent limitations in detecting small objects, and nowadays devised assignment or sampling schemes contribute minimally to this problem [9, 40]. Since the proposals play such a crucial role in two-stage detectors, how about the improved Region Proposal Network (RPN) variants [13, 32, 33] meet small object detection? Following this line, we take Cascade RPN [32], one of the most superior proposal network for generic object detection, to perform preliminary experiments and the results are shown in Table 1. While the auxiliary regression phase provides refined priors with better initialization for subsequent regression, the final results remain somewhat unsatisfactory. Specifically, the improvement mainly comes from the larger objects and the \( AP_{r_S} \) as well as \( AP_{r_S} \) actually decrease significantly instead, indicating that the region-based sampling strategy is inclined to large instances which further dominate the proposal network. Meanwhile, enlarging the sample region contributes little (even negative) to this condition (see the bottom row in Table 1). Therefore, the coarse-to-fine pipeline has the potential to surmount the barrier of conventional prior-to-proposal paradigm, but the crux lies in dedicating sufficient attention to small instances.

Second, small objects usually lack discriminative information and distorted structures, leading the inclination of model to give ambiguous even incorrect predictions [1, 11]. Meanwhile, there are a certain amount of large instances embodying clear visual cues and better discrimination. Building upon this observation, several works proposed to bridge the representation gap between small objects and large ones, and most of them [2, 22, 28, 1] rely on Generative Adversarial Network (GAN) [14] or similarity learning [19, 36] to super-resolve/restore the features of size-limited instances under the guidance of large ones that are deemed to be visually authentic. However, these approaches overlook the fact: high quality \( \neq \) large size meanwhile small size \( \neq \) low quality. In other words, the criterion for humans and the model to decide whether a sample is competent to be a good example is distinct. For the latter, it is dynamic and should be adjusted according to the current optimization of the detector. Moreover, efforts in this line have to resort to sophisticated training strategies or additional models, which is time-consuming and break the conventional end-to-end paradigm.

Putting the above parts together, we propose a two-stage small object detector CFINet based on the coarse-to-fine pipeline and feature imitation learning. Concretely, enlightened by the multi-stage proposal generation scheme in Cascade RPN, we devise Coarse-to-fine RPN (CRPN). It firstly employs a dynamic anchor selection strategy to mine potential priors to conduct coarse regression, and henceforth, these refined anchors will be classified and regressed by the region proposal network. In addition, we extend the conventional classification-and-regression setting with an auxiliary Feature Imitation (FI) branch, which can leverage the regional features of high-quality instances to guide the learning of those objects with uncertain/mistaken predictions, and a loss function based on the Supervised Contrastive Learning (SCL) [18] is designed to optimize the whole process. The main contributions of this paper are summarized as follows:

- A coarse-to-fine proposal generation pipeline named CRPN was built to perform anchor-to-proposal procedure, where an area-based anchor mining strategy and cascade regression empower the high-quality proposals for small instances.

- An auxiliary Feature Imitation (FI) branch was introduced to enrich the representations of low-quality instances perplexing the model under the supervision of high-quality instances, and this novel branch is optimized by a tailored loss function based on SCL.

- The experiment results on the SODA-D and SODA-A datasets exhibit the superiority of our CFINet to detect these instances with extremely limited sizes.

### 2. Related Works

**Anchor Refinement and Region Proposals.** Two-stage anchor-based approaches heavily depend on the high-quality proposals [33, 32]. Towards this goal, RPN was first introduced in Faster RCNN [29] to produce proposals in a fully convolutional network, and this simple yet effective design facilitates end-to-end model optimization. Following RPN, [13] proposed to iteratively regress the predefined anchors. GA-RPN [33] discards the uniform anchoring strategy and formulates the anchor generation in two steps: first determining the locations which may contain objects and on which the anchor scales are predicted then. By installing a multi-stage anchor-to-proposal strategy and alleviating the misalignment between the refined anchors and image features, Cascade RPN [32] enables high-quality proposal generation. Unfortunately, current proposal-oriented
Figure 2. The overall architecture of CFINet. In Coarse-to-fine RPN (CRPN), the area-based Anchor Mining strategy ensures sufficient candidates for instances of various sizes (small: orange boxes, large: green boxes) based on the dynamic pos threshold, which will be then used to obtain the coarse proposals. After that, the alignment between coarse proposals and corresponding features is enabled by the Adaptive Convolution before feeding into the RPN to produce high-quality proposals (blue boxes). The Feature Imitation (FI) branch is devised to facilitate the representations of small instances, in which the RoI features of uncertain/mistaken predictions will be pulled to their counterparts of exemplar feature set in embedding space (throughout the Feat2Embed module), while pushed apart from the exemplar features of other categories and background. And the exemplar features are collected based on the model predictions using the proposed quality indicator, i.e., Instance Quality (IQ). We tailor a Feature Imitation loss function $L_{FI}$ to optimize this auxiliary branch. Note that we only exhibit single-level Feature Pyramid Network (FPN) [25] feature for clear illustration.

frameworks fail to produce high-quality proposals for instances with limited regions, and the root cause lies in the notoriously low overlaps between the objects and priors [40]. Different from the above methods, our coarse-to-fine proposal pipeline could exploit the potential of multi-stage refinement paradigm, thereby guaranteeing both the quantity and quality of proposals for instances with extremely limited sizes.

**Feature Imitation for Small Object Detection.** One of the major challenges to detect small objects is the low-quality representation [45, 40, 9], while large instances often with clear structures and discriminative features. Hence, a series of efforts have been made to boost the semantic representations of small/tiny instances by mining the intrinsic correlations between small and large objects. Based on the generative adversarial paradigm, Perceptual GAN [22] designs a generator that is optimized to produce high-quality representations of small instances to fool the subsequent discriminator. Bai et al. [1] devised a novel pipeline to restore a clear face from the inputting blurry one. Noh et al. [28] further introduced precise supervision for the super-resolution process of small objects. Moreover, Wu et al. [36] and Kim et al. [19] both exploited similarity learning to force the features of small-scale pedestrians close to that of the large-scale ones which are obtained by an additional model. The existence of super-resolution branch or offline feature bank hampers the end-to-end optimization while our method updates the exemplar features in an online fashion, which guarantees the diversity of high-quality feature sets thereby getting rid of the collapse issue.

**Contrastive Learning for Object Detection.** The recent explosion of self-supervised learning mainly comes from its Contrastive Learning fashion, and several works have extended this paradigm into detection fields. Detco [38] is an effective self-supervised framework for object detection which utilizes the image and its local patches to conduct contrastive learning. Wu et al. [37] applied contrastive learning to object detection under smoky conditions. Though contrastive learning has recently received considerable interests [4, 16, 35], the potential of utilizing contrastive learning for better representation of small objects has not yet been investigated to date.

3. **Our Method**

This section presents the details about CFINet. We start with a discussion about the inherent limitations of Cascade RPN when confronting small objects, then our coarse-to-fine high-quality proposal generation pipeline tailored for size-limited instances is introduced. Afterwards, we elucidate the architecture of newly designed Feature Imitation branch, also with the optimization and training procedure.
The overall architecture of CFINet is shown in Figure 2.

3.1. Towards Better Proposals

Limitations of Cascade RPN. High-quality proposals play a pivotal role in two-stage detectors, but need heuristic anchor settings. Cascade RPN [32] discards this conventional setup by placing one single anchor on each feature point and conducting multi-stage refinement. Though exhibiting superior performance on objects of general scales, Cascade RPN fails to tackle extremely small objects well due to its inherent limitations. Concretely, the distance metric used in first-stage regression cannot guarantee sufficient potential anchors for small objects who have significantly small center region. Moreover, Cascade RPN only marks eligible anchors on a single pyramid level as positive, while this heuristic scheme simply discards those possible anchors at other levels which can still convey the existence and rough location information of small objects [42].

Coarse-to-fine RPN. To remedy the aforementioned issues of Cascade RPN when handling small instances, we propose Coarse-to-fine RPN and the detailed structure is in Figure 2. First, we design an area-based anchor selection strategy to enable the instances of various sizes could have (relatively) adequate potential anchors. Concretely, for an object box with the width $w$ and the height $h$, any anchors who have an IoU larger than $T_a$ will be regarded as positive for the coarse regression, and $T_a$ is formulated as follow:

$$T_a = \max(0.25, 0.20 + \gamma \cdot \log \frac{\sqrt{w \cdot h}}{12}),$$  

where $\gamma$ denotes a scale factor and is set to default 0.15 in our experiments, and the term 12 actually corresponds to the minimal area definition of SODA dataset [9], which enables adequate samples for extreme-size objects and can be tuned for different datasets. Moreover, $\gamma$ and $\max$ operation keep the optimization from being overwhelmed by the low-quality priors. Taking IoU as the criterion to mine potential anchors, the optimization inconsistency in multi-stage regression of Cascade RPN can be averted. Meanwhile, the model determines the positive sample in a more smooth way on top of the proposed continuous threshold.

Distinct from Cascade RPN, we preserve anchors of all Feature Pyramid Network (FPN) [25] levels $\{P_2, P_3, P_4, P_5\}$ to perform first-stage regression. In this way, we could mine sufficient potential anchors for extremely small instances and meanwhile, larger instances still can obtain proper attention since the anchors matched to them have naturally higher IoUs, as discussed in Figure 1. After the first-stage regression, we then capture the offsets inside the regressed boxes and input them with the feature maps to RPN, in which the Adaptive Convolution [32] will be exploited to align the features and conduct second-stage regression and foreground-background classification.

Loss Function. The training objective of our CRPN is:

$$L_{CRPN} = \alpha_1 (L_{\text{reg}}^c + L_{\text{reg}}^f) + \alpha_2 L_{\text{cls}},$$  

where we use cross-entropy loss and IoU Loss [44] as $L_{\text{cls}}$ and $L_{\text{reg}}$, respectively. The $c$ and $f$ in Eq. (2) indicate the coarse-stage and fine-stage in our CRPN, and noting that we only do classification in the latter stage. The loss weights $\alpha_1$ and $\alpha_2$ are set to 9.0 and 0.9, respectively.

3.2. Feature Imitation for Small Object Detection

Efforts on exploiting the intrinsic correlations between objects of different scales to boost the representations of small objects have been made, but most of them fail to the effectiveness and diversity. Specifically, the majority of previous methods [22, 2, 1, 28] resort to GAN to super-resolve the representations of small instances. This calls for the sophisticated training schemes and is prone to fabricate fake textures and artifacts [11]. Another line of efforts turn to the similarity learning which either has to construct offline feature bank in a cumbersome way [19], or directly leverages $L_2$ norm to similarity measurement between different RoI features [36], potentially leading the feature collapse issue: the region features after amending could have high-similarity but lost their own characteristics. This homogenization in feature space actually impairs the generality and robustness of the model.

To mitigate the collapse risks and avoid the memory burden as well as enable the end-to-end optimization, we devise a Feature Imitation (FI) head (see Figure 2). Most importantly, instead of solely taking large-scale objects as the guidance of this procedure, we consider the model response in current state for each instance, thereby constructing a dynamic and currently optimized feature bank of proper exemplars in an online fashion. The FI branch mainly composes an Exemplar Feature Set and a Feature-to-Embedding (Feat2Embed) module, where the former reserves the RoI features of high-quality exemplars and the latter projects the input to the embedding space. Next we elucidate the details about our Feature Imitation branch.

What is a proper exemplar? As we discussed above, an exemplar is vital in the imitation learning. To determine the most representative/proper/high-quality examples which can deliver authentic guidance/supervision for small objects confusing the model at this moment, we first introduce a simple quality indicator for an instance. Given a ground-truth (GT) object $g = (c^*, b^*)$, where $c^*$ and $b^*$ denote its label and bounding box coordinates. Assuming the detection head outputs a prediction set $\mathcal{S} = \{C_i, IoU_i\}_{i=1,\ldots,M}$ for $g$, in which $C_i \in \mathbb{R}^{N+l}$ indicates the predicted classification vector and $IoU_i$ stands for the IoU of predicted box to GT, and $N$ is the number of foreground classes. Then we can obtain the potential high-quality set $\mathcal{S'} = \{(C_j, IoU_j)\} \arg \max C_j = \ldots$
Algorithm 1 Training of Feature Imitation branch.

**Input:**

- The set of GT boxes $G = \{c_i^*, b_i^*\}_{i=1,2,...,T}$ and corresponding RoI features $\{x_i^*\}_{i=1,2,...,T}$ in current batch;
- The set of exemplar RoI features $E = \{E_i\}_{i=1,2,...,N}$;
- The set of background RoI features in current batch $X_{bg}$;
- The threshold of high-quality $T_{hq}$;
- The number of pos/neg samples $N_{pos}$ and $N_{neg}$;
- The transformation function $\Gamma$;

**Output:**

- The set of positive embeddings $P_{pos}$ and negative embeddings $P_{neg}$

1: Initialize the set of positive features $X_{pos}$ and negative features $X_{neg}$ with $\emptyset$;
2: for $g$ in $G$ do
3: Compute the IQ of current $g$ according to Eq. (3)
4: $X_{neg}^g \leftarrow$ sample $N_{neg}$ features from $X_{bg} \cup E \setminus E_c^g$;
5: if $IQ \geq T_{hq}$ then
6: $E_c \leftarrow x_i^g$
7: $X_{pos}^g \leftarrow \Gamma(x_i^g)$
8: else
9: $X_{pos}^g \leftarrow$ sample $N_{pos}$ features from $E_c^g$
10: end if
11: $X_{pos} = X_{pos} \cup X_{pos}^g, X_{neg} = X_{neg} \cup X_{neg}^g$
12: Apply Eq. (4) to $X_{pos}$ and $X_{neg}^g$ to obtain $P_{pos}$ and $P_{neg}$
13: end for
14: return $P_{pos}$ and $P_{neg}$

$c_i^* \in \{1,2,...,M\}^* \times M$. Now the Instance Quality of an object $g$ is defined as:

$$IQ = \frac{1}{|S^g|} \sum_{j=1}^{M^*} C_{j,c^*} \cdot IoU_j$$  \hspace{1cm} (3)

The IQ of a GT serves as an indicator of the current model’s detection capability, enabling us to capture the high-quality exemplars who have precise localization and high-confidence classification scores, and the instances confusing the model often fail to either of them. By setting appropriate threshold, we can select proper instances to build the teacher feature-set and perform the imitation process.

**Feat2Embed Module.** Instead of directly measuring the similarity between different RoI features [36], we first embed these features with the simple Feat2Embed module. The input of FI branch is the region feature $x_i \in \mathbb{R}^{H \times W \times C}$ obtained by the RoI-wise operation, e.g., RoI Align, which will be first processed by three consecutive $3 \times 3$ convolutional layers (with no padding operation) to abstract compact representations. It is worth noting that we update parameters during the extraction of current regional features and freeze parameters during the extraction of exemplar ones (see Feat2Embed module in Figure 2), resulting in improved stability in performance. Subsequently, the intermediate features will be mapped to the embedding space on top of a two-layer perceptron and the embedding layer with the dimension of 128, in which the dimension of hidden layers is set to 512. We have also investigated various design choices and structures for our Feat2Embed module, and detailed information can be found in the Supplementary Materials. Finally, the output of the Feature Imitation branch is defined as:

$$v_i = \Theta_{FI}(x_i),$$ \hspace{1cm} (4)

where $\Theta_{FI}$ denotes the parameters of Feature Imitation branch to be optimized.

**Loss Function.** The objective of our FI head is simple: calculating the similarity between the RoI feature of proposal and that of the stored high-quality instances in embedding space, thereby pulling the features of those instances that confuse the model close to the exemplar ones of belonging category, while pushing apart from that of other categories and backgrounds. To this end, we propose a loss function based on Supervised Contrastive Learning [18] which extends the contrastive learning setup and allows multiple positive samples for an anchor object by exploiting the accessible label information. The loss function tailored for our FI branch is as follows:

$$L_{FI} = \frac{-1}{|P_{pos}|} \sum_{j} \sum_{v_p \in P_{pos}} \log \frac{\exp (v_j \cdot v_p / \tau)}{\sum_{v_n \in P} \exp (v_j \cdot v_n / \tau)},$$ \hspace{1cm} (5)

where $P = P_{pos} \cup P_{neg}$ denotes the sample set, while $P_{pos}$ and $P_{neg}$ represent the positive and negative set respectively and they have the same cardinality ideally, and $v_p$ and $v_n$ are the positive and negative sample from $P_{pos}$ and $P_{neg}$. Moreover, $j$ indexes the current proposal and $\tau$ indicates the
temperature which plays a crucial part in contrastive learning and needs to be well designed, and we conduct ablation studies (see Table 9) to determine the optimal setting in our framework. The total loss function is presented:

\[ \mathcal{L} = \mathcal{L}_{CRPN} + \mathcal{L}_{cls} + \mathcal{L}_{reg} + \alpha_3 \mathcal{L}_{FI}, \]

in which \( \mathcal{L}_{cls} \) and \( \mathcal{L}_{reg} \) are the original losses of detection head, and the \( \alpha_3 \) is utilized to scale the weight of Feature Imitation part. With contrastive learning setups, not only can we fulfill the imitation learning but also prevent the collapse issue, thereby boosting the representations of small instances effectively. Moreover, the imitation process is only installed in training phase and will not slow the pace of inference.

Training. Next we elucidate the training details of FI branch. The exemplar set \( \mathcal{E} = \{ \mathcal{E}_i \}^{i=1,2,\ldots,N} \) containing high-quality features of \( N \) foreground categories and \( \mathcal{E}_i = \{ \mathbf{x}_{i,j} \}^{j=1,2,\ldots,N_i} \) corresponding to the exemplar features of the \( i \)-th class, and \( N_i \) represents its size. We use \( T_{hq} \) to pick out those high-quality instances which are compatible to be a good exemplar, and in practice, we set a bound value to the number of high-quality predictions of an instance to filter the effect of fluctuation of the network. The function \( \mathcal{G} \) is used to augment features for high-quality instances, \( i.e., \) the positive features for a high-quality instance are the transformations of itself. The overall training procedure of FI branch is shown in Alg. 1 and more details please refer to the Supplementary Materials.

### 4. Experiments

#### 4.1. Dataset

To evaluate the effectiveness of our method, we perform extensive experiments on the recently released large-scale benchmark tailored for small object detection: SODA [9], including SODA-D and SODA-A.

**SODA-D.** Focusing on the driving scenario, SODA-D comprises 24828 high-quality images and 278433 instances distributed across nine categories: **people, rider, bicycle, motor, vehicle, traffic-sign, traffic-light, traffic-camera,** and **warning-cone.** One of the most distinctive strengths of SODA-D is its diversity in terms of period, geographical locations, weather conditions, camera viewpoints, etc.

**SODA-A.** SODA-A contains 872069 objects with oriented box annotations in 2513 aerial images and encompassing nine classes: **airplane, helicopter, small-vehicle, large-vehicle, ship, container, storage-tank, swimming-pool,** and **windmill.** The instances in SODA-A can appear in arbitrary orientations and are with significant density variations. To be specific, the average number of instances per image in SODA-A is about 350.

As a specialized benchmark for small object detection, the instances in SODA are tiny, with most of them having an average size ranging from 10 to 30 pixels (see Figure 3). In contrast to conventional datasets for object detection, SODA includes extensive ignore annotations, aimed at filtering out instances that are either too large or are challenging to be identified deterministically due to heavy occlusion or lens flare. This procedure helps the model focus on valu-

<table>
<thead>
<tr>
<th>Method</th>
<th>Publication</th>
<th>Schedule</th>
<th>AP</th>
<th>AP$_{50}$</th>
<th>AP$_{75}$</th>
<th>AP$_{S}$</th>
<th>AP$_{S}$</th>
<th>AP$_{G}$</th>
<th>AP$_{N}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>One-stage</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RetinaNet [26]</td>
<td>ICCV’17</td>
<td>1×</td>
<td>28.2</td>
<td>57.6</td>
<td>23.7</td>
<td>11.9</td>
<td>25.2</td>
<td>34.1</td>
<td>44.2</td>
</tr>
<tr>
<td>FCOS [31]</td>
<td>ICCV’19</td>
<td>1×</td>
<td>23.9</td>
<td>49.5</td>
<td>19.9</td>
<td>6.9</td>
<td>19.4</td>
<td>30.9</td>
<td>40.9</td>
</tr>
<tr>
<td>ATSS [46]</td>
<td>CVPR’20</td>
<td>1×</td>
<td>26.8</td>
<td>55.6</td>
<td>22.1</td>
<td>11.7</td>
<td>23.9</td>
<td>32.2</td>
<td>41.3</td>
</tr>
<tr>
<td>YOLOX [12]</td>
<td>ArXiv’21</td>
<td>70e</td>
<td>26.7</td>
<td>53.4</td>
<td>23.0</td>
<td>13.6</td>
<td>25.1</td>
<td>30.9</td>
<td>30.4</td>
</tr>
<tr>
<td>DyHead [10]</td>
<td>CVPR’21</td>
<td>1×</td>
<td>27.5</td>
<td>56.1</td>
<td>23.2</td>
<td>12.4</td>
<td>24.4</td>
<td>33.0</td>
<td>41.9</td>
</tr>
<tr>
<td><strong>Query-based</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CornerNet [21]</td>
<td>ECCV’18</td>
<td>2×</td>
<td>24.6</td>
<td>49.5</td>
<td>21.7</td>
<td>6.5</td>
<td>20.5</td>
<td>32.2</td>
<td>43.8</td>
</tr>
<tr>
<td>CenterNet [48]</td>
<td>ArXiv’19</td>
<td>70e</td>
<td>21.5</td>
<td>48.8</td>
<td>15.6</td>
<td>5.1</td>
<td>16.2</td>
<td>29.6</td>
<td>42.4</td>
</tr>
<tr>
<td>RepPoints [43]</td>
<td>ICCV’19</td>
<td>1×</td>
<td>28.0</td>
<td>55.6</td>
<td>24.7</td>
<td>10.1</td>
<td>23.8</td>
<td>35.1</td>
<td>45.3</td>
</tr>
<tr>
<td><strong>Two-stage</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline [29]</td>
<td>NeurIPS’15</td>
<td>50e</td>
<td>19.2</td>
<td>44.8</td>
<td>13.7</td>
<td>6.3</td>
<td>15.4</td>
<td>24.9</td>
<td>34.2</td>
</tr>
<tr>
<td>Cascade RPN [32]</td>
<td>NeurIPS’19</td>
<td>1×</td>
<td>28.9</td>
<td>59.4</td>
<td>24.1</td>
<td>13.8</td>
<td>25.7</td>
<td>34.5</td>
<td>43.0</td>
</tr>
<tr>
<td>RFLA [40]</td>
<td>ECCV’22</td>
<td>1×</td>
<td>29.7</td>
<td>60.5</td>
<td>25.9</td>
<td>12.5</td>
<td>25.5</td>
<td>33.4</td>
<td>44.7</td>
</tr>
<tr>
<td>CFINet (ours)</td>
<td>-</td>
<td>1×</td>
<td>30.7</td>
<td>60.8</td>
<td>26.7</td>
<td>14.7</td>
<td>27.8</td>
<td>36.4</td>
<td>44.6</td>
</tr>
</tbody>
</table>

Table 2. Comparison with state-of-the-art detection approaches on the SODA-D test-set, where ‘Baseline’ refers to Faster RCNN [29], serving as the baseline for the two-stage methods in the table. All the methods are trained on a ResNet-50 [17], except YOLOX (CSP-Darknet) [12] and CornerNet (HourglassNet-104) [21]. ‘Schedule’ denotes the number of epochs for training, in which ‘1×’ corresponds to 12 epochs and ‘50e’ indicates 50 epochs.
Table 3. Comparison with state-of-the-art detection approaches on the SODA-A test-set, where 'Baseline' refers to Rotated Faster RCNN [29], serving as the baseline for the two-stage methods in the table. Other settings are consistent with Table 2.

Figure 4. The average number of high-quality proposals generated by (a) RPN, (b) Cascade RPN and (c) CRPN for instances in extremely Small (eS), relatively Small (rS), generally Small (gS), and Normal (N) subsets, respectively. Noting that a proposal who has an IoU larger than 0.5 to any ground-truth boxes will be registered as a high-quality proposal.

Table 4. Average Recall (AR) performances of our CRPN and its counterparts on the SODA-D test-set. All the methods are with Faster RCNN (ResNet-50) as the baseline and trained for a 1× schedule, in which RPN denotes the vanilla version of Faster RCNN whose positive threshold in RPN stage is set 0.7, and RPN-0.5 represents the version with 0.5 as its positive threshold of RPN. The results are tested with 300 proposals per image.

Table 5. Ablation analysis of our method, in which 'Baseline' denotes the vanilla Faster RCNN, CRPN and FI indicate Coarse-to-fine RPN and Feature Imitation branch, respectively.

4.2. Implementation Details

In the following experiments, unless specified, we use train-set to conduct the training and leave test-set to performance comparisons and ablation studies. Considering that the images in SODA enjoy a very high resolution (∼4000 × 3000), we first split the original images into a series of 800 × 800 patches with a stride of 650, and similar to [9] these patches will be resized to 1200 × 1200 during training and testing. All the experiments in this paper are conducted on a single RTX 3090 with the batch size of 4. Only random flip involved in data augmentation. We train all the models with a 1× schedule (a bunch of 12 epochs), and the learning rate is set to 0.01 which decays after epoch 8 and epoch 11 by 0.1. The default optimizer is SGD with
the momentum of 0.9 and the weight decay of 0.0001. We use ResNet-50 [17] with FPN [25] for all models.

4.3. Main Results
To exhibit the effectiveness of our method, we conduct a thorough comparison with current representative approaches on the SODA-D and SODA-A.

Table 2 represents the results of our method and several mainstream approaches on the SODA-D test-set. Integrating with Faster RCNN [29], our CFINet achieves state-of-the-art performance with an overall AP of 30.7%, and outperforms the baseline model with 1.8% points. When delving into specific metrics, our method exhibits clear predominance, particularly on the most challenging metrics AP$_{eS}$ and AP$_{rS}$. Moreover, CFINet exceeds the tailored small object detection method RFLA [40] by a significant margin (1.0% on AP, 1.5% on AP$_{eS}$, and 0.9% on AP$_{rS}$) when both taking Faster RCNN as the baseline. Actually, RFLA sacrifices the performance on extremely Small instances (with a decrease of 0.6% points when compared to Faster RCNN).

On the SODA-A test-set, CFINet also achieves the best result and shows great advantage in comparison to other solutions especially on AP$_{eS}$ (see Table 3), indicating its superiority and generality. Furthermore, albeit exhibiting advantage on AP$_{50}$ metric and larger instances, Oriented RCNN [39] lags largely behind our approach on AP$_{50}$ (73.1% vs. 70.7%) and AP$_{eS}$ (13.5% vs. 12.5%).

4.4. Effectiveness of CRPN
One of the main designs in this paper is the Coarse-to-fine RPN, which is based on the observation that current fixed overlaps-based sampling paradigm is inappropriate for small instances due to the inherent contradictions, while the refined designs of RPN could partially reduce this barrier but still fail to satisfactory results. Here, we conduct thorough analyses to demonstrate the capability of our CRPN to generate high-quality proposals for size-limited instances.

We first exhibit the recall performances of our CRPN and its counterparts in Table 4, from which we can see that lower analyses to demonstrate the capability of our CRPN to but still fail to satisfactory results. Here, we conduct thorough analyses to demonstrate the capability of our CRPN to generate high-quality proposals for size-limited instances.

We first exhibit the recall performances of our CRPN and its counterparts in Table 4, from which we can see that lower analyses to demonstrate the capability of our CRPN to generate high-quality proposals for size-limited instances.

We first exhibit the recall performances of our CRPN and its counterparts in Table 4, from which we can see that lower analyses to demonstrate the capability of our CRPN to generate high-quality proposals for size-limited instances.

4.5. Ablation and Discussion
In this part, we conduct ablation studies as well as comprehensive discussions to attest the importance of the CRPN and FI branch, and moreover, determine the appropriate settings of our approach. All the experiments of this section are conducted on the SODA-D test-set. Investigation of Designed Components. We first perform ablation experiments to verify the effectiveness of two modules. As in Table 5, our CRPN and FI can both improve the performances steadily, while the introduction of feature imitation branch is conducive for the recognition of size-
limited instances. Consistently, the integration of FI and CRPN achieves the best result, as CRPN is capable of generating sufficient high-quality proposals (as shown in Figure 4) thereby offering a more accurate indication of instance quality and the potential to act as an exemplar. 

**Fixed or Dynamic.** A natural idea is directly setting fixed positive threshold to obtain more anchors for one-stage regression of CRPN. Here we show that our simple-yet-effective area-based anchor mining strategy can achieve the best performance. In Table 6, when the IoU threshold of a positive potential anchor for first-stage regression drops to 0.20, the $AP_{5}$ is only 13.7% and this could be attributed to low-quality samples and the predominance of large instances as discussed before. The proposed area-based anchor mining strategy could mitigate this problem and obtain the best overall accuracy.

**Weights of Feature Imitation Loss.** In this section, we analyze the impact of the weight parameter of FI branch (namely the hyper-parameter $\alpha_3$) on the model. As shown in Table 7, paying too little or excessive attention both deteriorates the final performance, hence we set $\alpha_3$ to 0.5 in our experiments to ensure overall accuracy.

**The Criterion of Being An Exemplar.** The quality of the exemplar plays a pivotal role in the imitation process [19, 36]. Next we discuss the choices for capturing a high-quality exemplar to build the feature set. In Table 8, the lower $T_{ihq}$ involves more exemplars and updates the teacher set more frequently while the higher $T_{ihq}$ does exactly the opposite. It can be seen that increasing the $T_{ihq}$ from 0.5 to 0.65 could facilitate the imitation process and when the $T_{ihq}$ reaches 0.70, the overall AP drops instead. This may originates that the earliest samples stored in the feature set are inappropriate for current state, because the optimization is dynamic and the model is evolving hence the criterion of being an exemplar has already changed.

**The Temperature.** The temperature is paramount for contrastive learning [18] and we conduct a series of experiments to verify the best choice for $\tau$. From Table 9, when $\tau$ ranges from 0.10 to 0.80, the overall performance increases first and then decreases to 30.2%, therefore we choose 0.6 for our method.

**Visualization.** We demonstrate the visualization results of example images from SODA-D test-set in Figure 5 to intuitively show the capability of our detector when detecting the small instances.

5. Conclusion

In this paper, we proposed CFINet, a two-stage detector based on the Coarse-to-fine Region Proposal Network and Feature Imitation setups, in which the former can produce sufficient high-quality proposals for small instances particularly for those with extremely limited sizes. Then the novel detection head on top of the feature imitation branch facilitates the representations of small objects posing challenges to the model under the contrastive learning paradigm. The experiments results show that our method achieves state-of-the-art performance on the large-scale small object detection datasets SODA-D and SODA-A. In the future, a more flexible and general indicator of instance quality is worth investigating.

Acknowledgments

This work was supported in part by the National Science Foundation of China under Grant 62136007 and Grant U20B2068, and in part by the Natural Science Basic Research Program of Shaanxi under Grants 2021JC-16 and 2023-JC-ZD-36.
References


