MutDet: Mutually Optimizing Pre-training for Remote Sensing Object Detection

Ziyue Huang[1], Yongchao Feng[1] Qingjie Liu[1,2*], and Yunhong Wang[1,2]

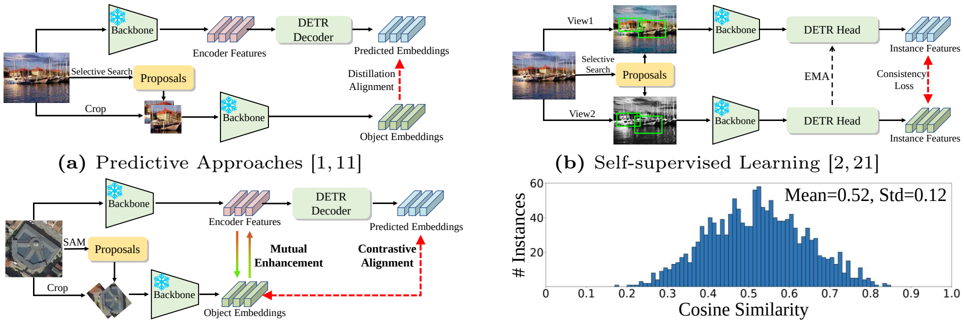
[1] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
[2] Hangzhou Innovation Institute, Beihang University
{ziyuehuang, fengyongchao, qingjie.liu, yhwang}@buaa.edu.cn

Abstract. Detection pre-training methods for the DETR series detector have been extensively studied in natural scenes, *e.g.*, DETReg. However, the detection pre-training remains unexplored in remote sensing scenes. In existing pre-training methods, alignment between object embeddings extracted from a pre-trained backbone and detector features is significant. However, due to differences in feature extraction methods, a pronounced feature discrepancy still exists and hinders the pre-training performance. The remote sensing images with complex environments and more densely distributed objects exacerbate the discrepancy. In this work, we propose a novel **Mut**ually optimizing pre-training framework for remote sensing object **Det**ection, dubbed as **MutDet**. In MutDet, we propose a systemic solution against this challenge. Firstly, we propose a mutual enhancement module, which fuses the object embeddings and detector features bidirectionally in the last encoder layer, enhancing their information interaction. Secondly, contrastive alignment loss is employed to guide this alignment process softly and simultaneously enhances detector features' discriminativity. Finally, we design an auxiliary siamese head to mitigate the task gap arising from the introduction of enhancement module. Comprehensive experiments on various settings show new state-of-the-art transfer performance. The improvement is particularly pronounced when data quantity is limited. When using 10 % of the DIOR-R data, MutDet improves DetReg by 6.1% in AP_{50}. *Codes and models are available at: https://github.com/floatingstarZ/MutDet.*

Keywords: Detection Pre-training, Oriented Object Detection, Remote Sensing

1 Introduction

DETR-based methods [15, 19, 25, 41] have recently been successfully applied to oriented object detection [33] in remote sensing images. However, DETR comes with training and optimization challenges, which need a large-scale training dataset due to the increased parameters in detection modules. In remote sensing images, the objects are densely distributed in the overhead view with



(a) Predictive Approaches [1, 11]

(b) Self-supervised Learning [2, 21]

(c) Mutually Optimizing (Ours)

(d) Feature Discripancy

Fig. 1: Motivation of our method. (a) The predictive approaches [1,11] utilize the embedding alignment task to learn visual knowledge from the pre-trained backbone. The feature discrepancy [2] between object embeddings and detector features impedes the effectiveness of pre-training. (b) Methods based on self-supervised learning [2, 21] circumvent feature discrepancy but can not sufficiently leverage the knowledge from pre-trained backbone. (c) Our approach employs contrastive alignment to achieve mutual learning between object embeddings and predictions, alleviating feature discrepancy. Simultaneously, we enhance the learning of visual knowledge by deeply fusing object embeddings with encoder features. (d) We use cosine similarity to measure the distance between object embeddings and predictions. The detector fails to fit the object embeddings, indicating the feature discrepancy problem.

arbitrary orientation, which requires more time and expert knowledge for annotation. The high annotation cost makes it difficult to obtain large-scale annotated datasets. We aim to address these challenges through detection pre-training [1, 2, 11, 17, 21, 32], wherein detection modules are unsupervised pre-trained using generated pseudo-labels.

Detection pre-training can broadly be categorized into predictive and self-supervised learning approaches [16]. Predictive approaches (Figure 1a) such as UP-DETR [11] and DETReg [1] achieve pre-training by making the detector to align the object embeddings of cropped images. The alignment is achieved by distillation-like alignment loss, enabling the model to learn fine-grained local features required for detection. However, there still exists a significant feature discrepancy [2] (Figure 1d) between the detector features and the object embeddings, hindering the pre-training performance. The feature discrepancy mainly arises from different feature extraction ways. The detector utilizes the image feature and a DETR decoder to predict embeddings, while object embeddings are extracted from cropped images through an entire backbone. The object embeddings contain deeper visual features unaffected by contextual interference. The complex and dense distribution of objects in remote sensing images also exacerbates the discrepancy. Self-supervised learning methods (Figure 1b), such as AlignDet [21] and PreSoco [2], constrain the consistency of instance features across different views to achieve self-training, which misses valuable visual knowledge inherent in pre-trained backbone.

To address these limitations, we propose a **Mut**ually optimizing pre-training framework for remote sensing object **Det**ection, dubbed as **MutDet** (Figure 1c). We introduce a mutual enhancement module to alleviate feature discrepancy. Concretely, we utilize bidirectional cross attention layers to deeply fuse the object embeddings and the encoder feature of the detector. The enhanced encoder feature is employed for subsequent training, prompting the DETR head to acquire visual knowledge from the pre-trained backbone more effectively. A contrastive alignment loss is employed to achieve collaborative optimization between object embeddings and predictions, enhancing detector features' discriminability. During the fine-tuning stage, since object embeddings are not accessible, the addition of mutual enhancement module will lead to a task gap [11] between the pre-training and fine-tuning. To address this issue, we design a calibration mechanism by adding an auxiliary siamese head.

In summary, our contributions are listed as follows: (1) We investigate pre-training methods for oriented object detection in remote sensing and propose a novel pre-training framework, *i.e.*, MutDet. To the best of our knowledge, this is the first detection pre-training method in the field of remote sensing. (2) To mitigate the feature discrepancy issue, we present a mutually optimizing pre-training strategy, which includes the mutual enhancement module and contrastive learning. We introduce an auxiliary siamese head to address the task gap between pre-training and fine-tuning. (3) We compare MutDet with three detection pre-training methods on three datasets. Compared to DETReg [1] baseline, MutDet improved by 2.8% on DIOR-R, 1.9% on DOTA-v1.0, and 2.99% on OHD-SJTU-L, respectively. The improvement is more significant in conditions with limited data quantity or training time in fine-tuning. On DIOR-R, when using 10% of the data, MutDet outperforms DETReg by 6.1% in AP_{50} and 4.5% in AP_{75}; when using 1/3 of the training time (12 epochs), MutDet outperforms DETReg by 4.8% in AP_{50} and 3.9% in AP_{75}.

2 Related Work

2.1 Oriented Object Detection

In the past decade, remarkable progresses [12, 14, 34, 38–41] have been made in the field of oriented object detection in remote sensing images. RoI Transformer [12] first adapts the two-stage detection framework to oriented detection task. GWD [38] and several related works [39,40,44] optimize the localization loss with respect to rotation annotations. ReDet [14] proposes a rotation-equivariant network and a rotation-invariant feature extractor. Oriented R-CNN [34] designs oriented region proposal network for efficient detection. Recently, DEtection TRansformer (DETR) [3, 42, 47] methods have been applied to this field. These methods [10, 19, 25, 41] introduce detection modules tailored for rotation object perception. ARS-DETR [41], built upon Deformable-DETR [47], proposes an angle classification method and a rotated deformable attention module, effectively enhancing DETR's high-precision detection performance. Previous research focuses on improvement in model structure, with a limited exploration of training paradigms. Our work indicates that introducing detection pre-training before finetuning on downstream datasets can accelerate convergence and effectively improve detection performance.

2.2 Pre-training for detection

Using detection pre-training with fine-grained pretext tasks has been proven to effectively enhance the fine-tuning performance in natural scenes, especially for DETR-based detectors [1,2,11,17,21]. Detection pre-training commonly utilizes unsupervised region proposal algorithms (*e.g.* Selective Search [28]) to generate object regions for learning object localization, with various approaches to learning object representations. SoCo [32] introduces instance-level multi-scale contrastive learning to train all modules of convolutional detectors from scratch. However, training all modules, as with self-supervised image representation learning [13], requires expensive training resources. Subsequent research [1, 11, 21] usually freezes the well-pre-trained backbone and trains only the detection-related modules, which reduces costs and preserves the generalization of the backbone. UP-DETR [11] and DETReg [1] use pre-trained backbone to generate object embeddings and learn object representations through alignment with embeddings. ProSeCo [2] points out that discrepancies between object embeddings and detector features may hinder pre-training performance and proposes a self-supervised pre-training method based on a student-teacher architecture. Recent research has employed a multi-view contrastive learning framework for detection pre-training [17,21]. Unfortunately, few works explored the detection pre-training in remote sensing images. In this paper, we design a novel detection pre-training framework for remote sensing scenes, in which we propose a systemic solution to address the feature discrepancy issue.

3 Method

3.1 Detector and Preparation

Detector. We build our methods upon ARS-DETR [41], a strong DETR-detector for remote sensing images. ARS-DETR adopts a two-stage detection paradigm [47], comprising a backbone, a transformer encoder, a transformer decoder containing 6 decoder layers, and multiple prediction heads. The encoder integrates multi-scale features from the backbone and predicts 300 rough proposals, which are further refined by the decoder to obtain detection results through prediction heads.

Preparation. We employ a pipeline similar to DETReg [1] to generate the pseudo-labels containing boxes, classes, and object embeddings. The well-trained Segment Anything Model (SAM) [18] is utilized to generate the oriented bounding boxes. Then, we crop the image patches according to the boxes and utilize a pre-trained backbone to extract patch features. Inspired by AlignDet [21], we collect the features and apply principal component analysis to reduce the dimension, resulting in 256-dimensional object embeddings. Simultaneously, we employ

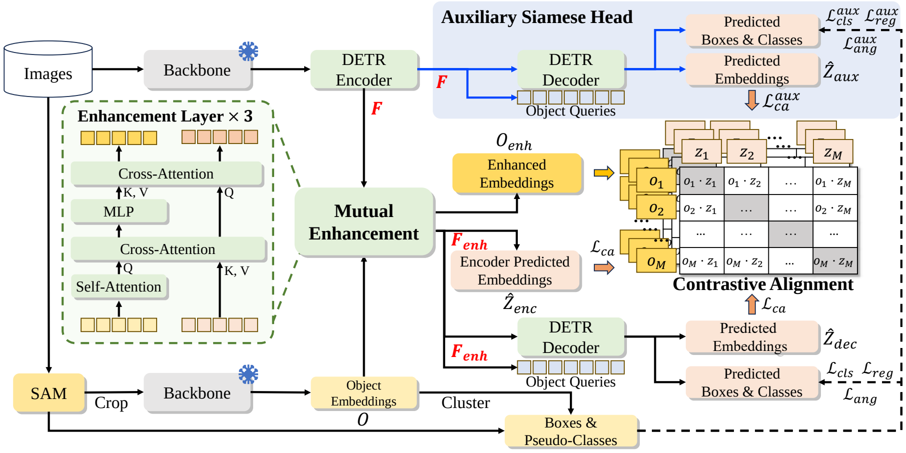
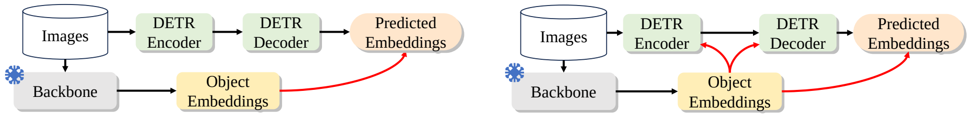


Fig. 2: Overall architecture of the proposed MutDet. MutDet optimizes DETReg [1] and introduces SAM [18] to generate proposals. It utilizes **mutual enhancement module** to cross-fuse the object embeddings and encoder features. Then, it uses **contrastive alignment loss** to optimize the enhanced object embeddings and predicted embeddings mutually. The **auxiliary siamese head** is proposed to alleviate the task gap between pre-training and fine-tuning, which shares parameters with the DETR decoder.

k-means to cluster the normalized object embeddings into 256 classes. To reduce pre-training costs, we perform bounding box generation, object embedding extraction, and clustering in an offline manner, *i.e.*, the generated pseudo-labels remain unchanged throughout the pre-training process.

3.2 Overview

The overview of our method is shown in Figure 2. Given an input image I, we first obtain its pseudo-labels, including object embeddings O, boxes, and pseudo-classes as described in the Preparation. For the detector, the encoder features F are produced by passing I through a frozen backbone and the DETR encoder sequentially. Subsequently, F along with O is fed into **mutual enhancement module** (Sec. 3.3), yielding enhanced encoder feature F_{enh} and enhanced object embeddings O_{enh}. F_{enh} is further utilized to obtain encoder predicted embeddings $\hat{Z}_{enc}$. In conjunction with object queries, F_{enh} is fed into the DETR decoder to predict the embeddings $\hat{Z}_{dec}$, boxes, and classes. To solve the task gap arising from the mutual enhancement module, we design an **auxiliary siamese head** (Sec. 3.5), which shares parameters with the DETR decoder. This auxiliary head takes F as input and predicts embeddings $\hat{Z}_{aux}$, boxes, and classes, similar to the DETR decoder. Finally, $\hat{Z}_{dec}$, $\hat{Z}_{enc}$, $\hat{Z}_{aux}$ and O_{enh} are each paired as inputs of the **contrastive alignment loss** (Sec. 3.4) to calculate the alignment loss. Next, we describe each module in detail.



(a) DETReg [1] **(b)** MutDet (Ours)

Fig. 3: A diagram illustrates the differences between the DETReg and MutDet with respect to supervisions. DETReg is only supervised by object embeddings via one pathway in the last decoder layer. In contrast, our MutDet is supervised through multiple pathways. The red line represents the supervision signal.

3.3 Mutual Enhancement Module

We utilize a mutual enhancement module to alleviate the feature discrepancy. Let $F \in \mathbb{R}^{K\times C}$ be the flattened multi-scale feature output by the DETR encoder, and $O \in \mathbb{R}^{M\times C}$ be the object embeddings, where K denotes the number of sampling points, M denotes the number of object embeddings, and $C = 256$ denotes the feature dimension. The mutual enhancement module employs three enhancement layers to achieve bidirectional feature interaction. The i-th enhancement layer could be formulated as follows:

$$\begin{aligned} O' &= \mathrm{LN}(\mathrm{MHSA}(O_i) + O_i), & O'' &= \mathrm{LN}(\mathrm{MHCA}(O', F_i) + O') \\ O_{i+1} &= \mathrm{LN}(\mathrm{MLP}(O'') + O''), & F_{i+1} &= \mathrm{LN}(\mathrm{MHCA}(F_i, O_{i+1}) + F_i) \end{aligned} \tag{1}$$

where multi-head self-attention (MHSA) layer, multi-head cross-attention (MHCA) layer, layer normalization (LN), and multi-layer perception (MLP) are the typical modules in Transformer [29]. O_{i+1} and F_{i+1} are the obtained fused object embeddings and encoder features of the $i+1$ layer, respectively. The final enhanced encoder feature F_{enh} and enhanced object embeddings O_{enh} will be employed for pre-training. F_{enh} are fed into the decoder to predict embeddings, which are then aligned with O_{enh}, as shown in Figure 2. In this process, we also introduce the DeNoising [42] strategy to obtain more accurate supervision signals.

From the perspective of backpropagation, DETReg [1] can only receive supervised signals from object embeddings through predictions, which cannot be directly propagated to each module in the detector, as shown in Figure 3a. In MutDet, the mutual enhancement module integrates object embeddings with encoder features sufficiently during the forward propagation, allowing the supervised signals from object embeddings to influence modules in the detector through multiple pathways, as illustrated in Figure 3b. Therefore, MutDet can receive diversified supervision from object embeddings and effectively learn visual knowledge from the pre-trained backbone.

3.4 Alignment via Contrastive Learning

We adopt contrastive alignment loss on enhanced embeddings O_{enh} to accomplish alignment, driven by two main reasons. Firstly, instance-level contrastive alignment is equivalent to maximizing the mutual information between the distribution of object embeddings and predicted embeddings [36], thereby facilitating the learning of shared knowledge. Contrastive learning also prevents overfitting to semantics that might hinder generalization [31]. Secondly, since enhanced object embeddings become learnable, directly applying distillation loss will lead to feature collapse. In contrast, negative samples in contrastive learning solve this issue. The contrastive alignment loss between M normalized object embeddings $O = \{\mathbf{o}_1, ..., \mathbf{o}_M\}$ and normalized predicted embeddings $Z = \{\mathbf{z}_1, ..., \mathbf{z}_M\}$ could be formulated as:

$$\mathcal{L}_{ca}(Z, O) = -\frac{2\tau}{M}\sum_{i=1}^{M}\left[\log\frac{\exp(\mathbf{z}_i \cdot \mathbf{o}_i/\tau)}{\sum_{k=1}^{M}\exp(\mathbf{z}_i \cdot \mathbf{o}_k/\tau)} + \log\frac{\exp(\mathbf{o}_i \cdot \mathbf{z}_i/\tau)}{\sum_{k=1}^{M}\exp(\mathbf{o}_i \cdot \mathbf{z}_k/\tau)}\right] \tag{2}$$

where the temperature coefficient τ default set to 0.2 [8], and we assume O and Z are one-to-one matched according to the index.

Here, we describe contrastive alignment loss in detection pre-training in detail. For an input image I with M annotations $\{\mathbf{b}_i, \mathbf{c}_i, \mathbf{a}_i, \mathbf{o}_i\}_{i=1}^{M}$, where $\mathbf{b} = (x, y, w, h)$ denotes the 4D bounding box, $\mathbf{c} \in \{1, 2, ..., 256\}$ denotes the category, $\mathbf{a} \in [-\pi/2, \pi/2)$ denotes the rotation angle, and $\mathbf{o} \in \mathbb{R}^{256}$ denotes the object embedding. In MutDet, the DETR decoder outputs four predictions at each layer: 4D bounding boxes $\{\hat{\mathbf{b}}_i = (x_i, y_i, w_i, h_i)\}_{i=1}^{N}$, classification scores $\{\hat{\mathbf{p}}_i \in \mathbb{R}^{256}\}_{i=1}^{N}$, angle classification scores $\{\hat{\mathbf{a}}_i \in \mathbb{R}^{180}\}_{i=1}^{N}$ [41], and predicted embeddings $\{\hat{\mathbf{z}}_i \in \mathbb{R}^{256}\}_{i=1}^{N}$, where N denotes the number of object queries. Following DETReg [1], we apply contrastive alignment loss only to the final layer predictions. We perform Hungarian bipartite matching [3] to one-to-one match predictions $\{\hat{\mathbf{b}}_i, \hat{\mathbf{p}}_i, \hat{\mathbf{a}}_i\}_{i=1}^{N}$ and annotations $\{\mathbf{b}_i, \mathbf{c}_i, \mathbf{a}_i\}_{i=1}^{M}$ as the same in ARS-DETR [41], resulting in the optimal permutation σ. Note that only the positive predicted embeddings $\hat{Z}^{+}_{dec} = \{\hat{\mathbf{z}}_{\sigma(i)}\}_{i=1}^{M}$ in matching are participate in the contrastive alignment loss. Besides, to further enhance supervision, we add extra contrastive alignment constraint on the enhanced encoder features F_{enh}. We feed F_{enh} into the prediction heads of the encoder and select the top N predictions according to the classification scores. Then, we also perform Hungarian bipartite matching to match the selected predictions and annotations $\{\mathbf{b}_i, \mathbf{c}_i, \mathbf{a}_i\}_{i=1}^{M}$ to get positive encoder predicted embeddings $\hat{Z}^{+}_{enc}$. The alignment loss $\mathcal{L}^{det}_{ca}$ of the detector is as follows:

$$\mathcal{L}^{det}_{ca} = \mathcal{L}_{ca}(\hat{Z}^{+}_{enc}, O_{enh}) + \mathcal{L}_{ca}(\hat{Z}^{+}_{dec}, O_{enh}) \tag{3}$$

In addition to the embedding alignment task, detection pre-training includes localization and classification tasks [1, 11, 21]. Here, the classification loss $\mathcal{L}_{cls}$ utilizes the focal loss [22] with the pseudo-labels obtained through clustering. The regression loss $\mathcal{L}_{reg}$ consists of the GIoU [26] loss and L_1 loss computed between the spatial coordinates (excluding angles) of predicted boxes and SAM-generated boxes. The angle loss $\mathcal{L}_{ang}$ uses the angle classification loss for oriented object detection [41]. Therefore, the overall loss of the detector is as follows:

$$\mathcal{L}_{det} = \mathcal{L}^{det}_{ca} + \mathcal{L}_{cls} + \mathcal{L}_{reg} + \mathcal{L}_{ang} \tag{4}$$

Table 1: The comparison of different calibration mechanisms. Pre-training is conducted using the DOTA-v1.0 dataset, and results on DIOR-R are reported, detailed settings are described to Sec. 4.1. Superscript denotes the improvement compared to without using calibration.

	Calibration Mechanism		AP$_{50}$	AP$_{75}$
	Encoder	Decoder		
w/o calibration	-	-	69.4	50.0
Encoder Feature Distillation	Distillation	-	$69.6^{+0.2}$	$50.4^{+0.4}$
Decoder Cross Distillation	-	Distillation	$69.9^{+0.5}$	$50.8^{+0.8}$
Auxilary Siamese Head	-	Training	$\mathbf{70.7}^{+1.3}$	$\mathbf{51.2}^{+1.2}$

3.5 Auxiliary Siamese Head

Introducing the mutual enhancement module allows the detector to better fit the pre-training dataset. However, since object embeddings are not accessible during fine-tuning, feature enhancement cannot be performed. The enhancement module brings the task gap that affects the transferability of the pre-training.

Therefore, we consider a calibration mechanism to alleviate the gap. We aim that the original encoder feature F to learn visual knowledge as effectively as the enhanced feature F_{enh}. We also expect the decoder to adapt to the distribution of F. Fortunately, self-distillation [6, 30] can achieve this goal. Based on this clue, we attempt to introduce knowledge distillation for object detection [5, 35, 45]. Three strategies are tested: encoder feature distillation, decoder cross distillation, and auxiliary siamese head, as shown in Table 1. The encoder feature distillation is inspired by simple knowledge distillation [6], which employs L_2 feature distillation loss to align F and F_{enh}. However, this approach fails to enable the decoder to adapt synchronously to the distribution of F. The decoder cross-distillation [30] feeds both F and F_{enh} into a shared decoder and uses distillation loss to align the outputs of the decoder. The distillation losses include knowledge distillation quality focal loss [30] for classification and L_2 loss for embedding alignment. However, since F_{enh} and the shared decoder continuously change during pre-training, obtaining accurate predicted embeddings and labels for effective distillation is challenging. Compared with the above two strategies, the simple siamese head yields the best performance. Different from the decoder cross-distillation, the auxiliary siamese head utilizes pseudo-labels to supervise the decoder output corresponding to F, as shown in Figure 2. This approach allows the decoder to receive more precise and stable supervision signals. Furthermore, the shared decoder acts as an implicit constraint, guiding F towards F_{enh}.

We only apply contrastive alignment loss in Eq. 2 to the shared decoder:

$$\mathcal{L}_{ca}^{aux} = \mathcal{L}_{ca}(\hat{Z}_{aux}^{+}, O_{enh}) \tag{5}$$

where $\hat{Z}_{aux}^{+}$ denotes the positive predicted embeddings output by auxiliary siamese head. Meanwhile, we still utilize detection-related losses: $\mathcal{L}_{cls}^{aux}$, $\mathcal{L}_{reg}^{aux}$, and $\mathcal{L}_{ang}^{aux}$.

And the loss of auxiliary siamese head is as follows:

$$\mathcal{L}_{det}^{aux} = \mathcal{L}_{ca}^{aux} + \mathcal{L}_{cls}^{aux} + \mathcal{L}_{reg}^{aux} + \mathcal{L}_{ang}^{aux} \tag{6}$$

Ultimately, the overall loss for MutDet is:

$$\mathcal{L}_{mut} = \mathcal{L}_{det} + \mathcal{L}_{det}^{aux} \tag{7}$$

4 Experiments

4.1 Dataset and Implementation Details

Pre-training Datasets. We only use the training and validation sets for pre-training to avoid data leakage. In this work, we choose DOTA-v1.0 [33] as the pre-training dataset and evaluate pre-training methods on multiple remote sensing detection datasets. Before the pre-training, images are divided into 800×800 patches with an overlap of 200 pixels, resulting in 28,249 images.

In addition, we perform large-scale pre-training by incorporating more high-quality data. We collect four large-scale remote sensing datasets and also divide the images into 800×800 patches with an overlap of 200 pixels: DOTA [33] (28,249 images, 2,952,632 boxes), DIOR-R [9] (11,725 images, 1,165,600 boxes), FAIR-1M-2.0 [27] (20,627 images, 2,519,081 boxes), and HRRSD [43] (29,916 images, 2,885,530 boxes), denoted as RSDet4. RSDet4 contains 90,518 images and nearly 10 million oriented bounding boxes, showcasing rich diversity and aligning well with remote sensing object detection tasks.

Pre-training Details. We compare three detection pre-training methods: UP-DETR [11], DETReg [1], and AlignDet [21]. Based on ARS-DETR, these methods are re-implemented in the MMRotate framework [46] to adapt to oriented object detection tasks. We employ SAM to generate proposals to better handle directional, dense, and small objects in remote sensing. Firstly, the SAM's automatic pipeline is employed to generate instance masks. We choose the ViT-H version of SAM [18], utilizing a configuration with a 64×64 point grid and a Non-Maximum Suppression (NMS) threshold of 0.8, producing about 200 masks per image on average. The instance masks are converted into oriented bounding boxes through the minimum bounding box algorithms. All boxes are utilized for pre-training to cover as many objects as possible. We find that the self-supervised pre-training (*e.g.*, SwAV [4]) demonstrates inferior performance in remote sensing detection compared to supervised pre-training on ImageNet. Hence, we initialize the backbone with ImageNet pre-training. The backbone remains fixed during detection pre-training [1, 11, 21]. We use AdamW optimizer [24] with the initial learning rate of 1×10^{-4} and train models on 4 NVIDIA GeForce RTX 3090 GPU with a total batch size of 8 for 36 epochs. We adopt a learning rate warm-up for 500 iterations, and the learning rate is reduced by a factor of 0.1 at the 32^{nd} epoch.

Fine-tuning Datasets. Three datasets are selected to validate the transferability of pre-trained models. We use training set, validation set for fine-tuning

Table 2: Comparison results on DIOR-R [9]. All methods adopt ARS-DETR [41] as a detector and use ResNet-50 as the backbone. Models are trained on the trainval set and evaluated on the test set. '-' indicates pre-training free, *i.e.*, not using detection pre-training. **Red**: optimal results. **Blue**: sub-optimal results.

Method	APL	APO	BF	BC	BR	CH	DAM	ESA	ETS	GF	GTF	HA	OP	SH	STA	STO	TC	TS	VE	WM	AP_{50}	AP_{75}
-	67.3	52.0	75.7	81.7	41.6	77.1	36.2	80.8	71.7	73.8	78.2	35.5	56.7	84.5	66.0	72.9	81.3	59.0	50.2	71.8	65.7	45.7
UP-DETR [11]	68.1	54.7	77.6	81.9	42.5	78.5	33.9	82.9	73.8	78.5	78.3	43.4	55.4	85.6	69.1	73.3	81.0	61.0	50.5	71.7	67.1	48.4
AlignDet [21]	67.6	53.4	73.4	81.2	40.4	77.1	38.4	81.1	71.6	73.4	78.1	35.8	54.7	84.6	69.7	73.2	80.7	59.7	49.6	71.8	65.8	45.8
DETReg [1]	69.0	55.5	75.7	82.4	45.4	78.6	35.6	81.7	72.4	78.8	78.2	46.1	57.7	87.6	71.5	76.0	81.3	60.6	52.9	72.1	**67.9**	**49.1**
MutDet (Ours)	75.1	60.5	78.8	84.7	45.4	81.2	39.6	85.7	77.0	78.0	81.9	52.1	57.8	88.2	78.8	77.4	84.7	60.2	54.2	72.4	**70.7**	**51.2**

Table 3: Comparison results on DOTA-v1.0 [33]. All adopt ARS-DETR [41] as detector and use ResNet-50 as backbone. The results of the test set are reported. '-' indicates pre-training free. **Red**: optimal results. **Blue**: sub-optimal results.

Method	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	AP_{50}	AP_{75}
-	87.1	71.9	47.0	68.3	72.9	75.5	87.2	90.4	83.8	82.6	52.2	60.5	74.9	71.7	67.5	**72.9**	49.6
UP-DETR [11]	80.1	77.0	48.9	70.2	74.6	76.3	87.7	90.6	78.4	82.9	53.3	66.0	75.3	70.9	59.1	72.7	**50.0**
AlignDet [21]	87.0	75.8	47.6	65.6	73.8	75.5	87.3	90.6	76.9	82.7	53.9	61.0	74.2	71.3	65.3	72.6	49.1
DETReg [1]	87.7	75.9	46.7	66.8	74.3	76.9	87.6	90.5	78.2	82.2	49.2	66.0	75.4	71.7	59.3	72.6	49.4
MutDet (Ours)	87.3	78.7	51.3	68.5	78.9	81.6	88.1	90.7	79.9	83.7	58.0	61.8	76.5	72.1	60.8	**74.5**	**51.6**

and test set for evaluation. DIOR-R [9] is an aerial image dataset with images from the DIOR [20] dataset, annotated with oriented bounding boxes. It contains 23,463 images and 192,518 instances in 20 common categories. All images in the dataset are in 800×800. DOTA-v1.0 [33] contains 2,806 high-resolution remote sensing images with spatial resolutions ranging from 800 to 4,000 pixels, totaling 188,282 instances. Images in DOTA-v1.0 are divided into 1024 × 1024 patches with an overlap of 200 pixels without extra scaling. OHD-SJTU [37] dataset consists of two subsets, namely OHD-SJTU-S and OHD-SJTU-L, containing 4,125 instances and 113,435 instances, respectively. Following ARS-DETR, we divide the images into 600×600 patches with an overlap of 150 pixels and scale them to 800×800.

Fine-tuning Details. Models are initialized with detection pre-trained weights before fine-tuning, maintaining the same training hyper-parameters with pre-training, except that batch size is set to 4. In addition to the 3× schedule for 36 epoch training, we test the 1× schedule for 12 epoch training on OHD-SJTU, with the learning rate decay set at the 10^{th} epoch. The AP_{50} and AP_{75} under the DOTA evaluation protocol are reported. Furthermore, we separately report the AP_{50} at 12, 24, and 36 epochs in the experiments.

4.2 Quantitative Results

Main Results. We compare the performance of detection pre-training methods on three datasets: DIOR-R, DOTA-v1.0, and OHD-SJTU, as shown in Table 2, Table 3, and Table 4, respectively. We also compare with the method that does not utilize detection pre-training, denoted as 'pre-training free.' All pre-training

Table 4: Comparison results on OHD-SJTU-S [37] and OHD-SJTU-L [37]. Results for two training schedules (*i.e.*, 1× and 3×) at various IoU thresholds are reported. '-' indicates pre-training free. **Red**: optimal results. **Blue**: sub-optimal results.

Method	OHD-SJTU-S [37]						OHD-SJTU-L [37]					
	1×			3×			1×			3×		
	AP_{50}	AP_{75}	$AP_{50:95}$	AP_{50}	AP_{75}	$AP_{50:95}$	AP_{50}	AP_{75}	$AP_{50:95}$	AP_{50}	AP_{75}	$AP_{50:95}$
-	89.89	76.29	58.79	90.40	82.81	63.90	68.28	34.33	36.92	**69.50**	39.64	39.47
UP-DETR [11]	90.27	82.85	67.79	89.73	82.97	66.72	70.69	42.15	40.88	69.48	43.23	40.44
AlignDet [21]	89.80	70.71	58.06	90.40	80.84	64.09	68.39	37.81	37.82	69.02	40.43	39.52
DETReg [1]	**90.56**	**83.23**	**68.40**	**90.49**	**83.31**	**68.83**	**70.87**	**44.79**	**41.87**	68.61	**45.23**	**41.03**
MutDet (Ours)	**90.67**	**88.09**	**70.56**	**90.41**	**84.04**	**69.82**	**73.46**	**45.06**	**43.34**	**71.60**	**44.01**	**41.68**

Table 5: Object detection using k% of the labeled data on DIOR-R. The models are trained on k% trainval set and then evaluated on test set. '-' indicates pre-training free. Superscript denotes the improvement compared to pre-training free. **Red**: optimal results. **Blue**: sub-optimal results.

Method	AP_{50}				AP_{75}			
	10%	25%	50%	100%	10%	25%	50%	100%
-	37.9	51.1	58.8	65.7	22.7	33.2	39.7	45.7
UP-DETR [11]	$49.9^{+12.0}$	$57.5^{+6.4}$	$62.0^{+3.2}$	$67.1^{+1.4}$	$35.0^{+12.3}$	$40.6^{+7.6}$	$44.4^{+4.7}$	$48.4^{+2.7}$
AlignDet [21]	$37.9^{+0.0}$	$50.6^{-0.5}$	$58.3^{-0.5}$	$65.8^{-0.1}$	$23.2^{+0.4}$	$32.2^{-1.0}$	$39.3^{-0.4}$	$45.6^{-0.1}$
DETReg [1]	$\mathbf{50.8}^{+12.9}$	$\mathbf{58.4}^{+7.3}$	$\mathbf{63.1}^{+4.3}$	$\mathbf{67.9}^{+2.2}$	$\mathbf{35.8}^{+13.1}$	$\mathbf{41.5}^{+8.3}$	$\mathbf{45.4}^{+5.7}$	$\mathbf{49.1}^{+3.4}$
MutDet (Ours)	$\mathbf{56.9}^{+19.0}$	$\mathbf{62.9}^{+11.8}$	$\mathbf{66.7}^{+7.9}$	$\mathbf{70.7}^{+5.0}$	$\mathbf{40.3}^{+17.6}$	$\mathbf{45.8}^{+12.6}$	$\mathbf{48.4}^{+8.7}$	$\mathbf{51.2}^{+5.5}$

methods use the training set of DOTA-v1.0 as the pre-training dataset and improve fine-tuning performance to some extent across all datasets. Our MutDet demonstrates consistent improvements over existing methods. As shown in Table 2, MutDet achieves a significant improvement of 5.0% in AP_{50} compared to pre-training free. Moreover, MutDet improves by 2.8% in AP_{50} and by 2.1% in AP_{75} compared to the baseline DETReg. Similarly, our method achieves consistent improvement on DOTA-v1.0, as shown in Table 3. Note that existing methods have all demonstrated negative impacts on fine-tuning, whereas MutDet still achieves a 1.6 % improvement, demonstrating excellent stability. The pre-training failure may be attributed to using the same dataset for both pre-training and fine-tuning. Consistency in data implies that it is challenging to acquire diverse object visual features distinct from the downstream task. Despite this limitation, MutDet still allows the detector to benefit from pre-training.

According to the data scale, the OHD-SJTU dataset includes two subsets, OHD-SJTU-S and OHD-SJTU-L, containing 2 and 6 categories, respectively. Experiments are conducted on respective subsets with 1× and 3× training schedules. As shown in Table 4, whether in 1× or 3× schedules, the improvement brought by detection pre-training on AP_{75} is more pronounced than one on AP_{50}. Under the 1× schedule on OHD-SJTU-S, compared to DetReg, MutDet improves by 0.11%, 4.86%, and 2.16% in AP_{50}, AP_{75}, and $AP_{50:95}$, respectively. Under the 1× schedule on OHD-SJTU-L, MutDet improves by 2.59%, 0.27%, and 1.47% in AP_{50}, AP_{75}, and $AP_{50:95}$, respectively. MutDet achieves either the optimal or sub-optimal results in all settings.

Table 6: Comparison results at different epochs during training on DIOR-R. All models are evaluated on test set at 12, 24, and 36 epochs. '-' indicates pre-training free. Superscript denotes the improvement compared to pre-training free. **Red**: optimal results. **Blue**: sub-optimal results.

Method	12 Epoch		24 Epoch		36 Epoch	
	AP_{50}	AP_{75}	AP_{50}	AP_{75}	AP_{50}	AP_{75}
-	55.4	37.4	61.5	42.2	65.7	45.7
UP-DETR [11]	**62.5**$^{+7.1}$	**44.5**$^{+7.1}$	64.7$^{+3.2}$	46.8$^{+4.6}$	67.1$^{+1.4}$	48.4$^{+2.7}$
AlignDet [21]	54.3$^{-1.1}$	35.9$^{-1.5}$	60.7$^{-0.8}$	41.1$^{-1.1}$	65.8$^{+0.1}$	45.6$^{-0.1}$
DETReg [1]	62.1$^{+6.7}$	44.2$^{+6.8}$	**65.1**$^{+3.6}$	**47.5**$^{+5.3}$	**67.9**$^{+2.2}$	**49.1**$^{+3.4}$
MutDet (Ours)	**66.9**$^{+11.5}$	**48.1**$^{+10.7}$	**69.8**$^{+8.3}$	**49.9**$^{+7.7}$	**70.7**$^{+5.0}$	**51.2**$^{+5.5}$

Low Training Resources. These experiments assess how detection pre-training methods perform when data quantity or training time is limited in the fine-tuning stage. Table 5 compares the performance of various methods when using k% annotated data in fine-tuning. The less data available, the more improvement detection pre-training could be achieved compared with pre-training free, with our MutDet achieving the best in all settings. When using 10% of the data, MutDet outperforms DETReg by 6.1% in AP_{50} and 4.5% in AP_{75}. Our MutDet leverages only 50% of the data to achieve comparable performance to pre-training free that uses 100% of the data. These results indicate that MutDet can more effectively acquire visual knowledge from the pre-training backbone and dataset, improving the model's detection performance under data scarcity. Table 6 shows the detection performance at different epochs during fine-tuning. MutDet exhibits the most significant improvement at 12^{nd} epoch, increasing by 11.5% over pre-training free and by 4.8% over DETReg in AP_{50}. At 24^{th} epoch, MutDet still improves by 8.3% over pre-training free and by 4.7% over DETReg in AP_{50}.

4.3 Ablation Study

Effectiveness of individual component. MutDet incorporates three designs: mutual enhancement module (Sec. 3.3), contrastive alignment loss (Sec. 3.4), and auxiliary siamese head (Sec. 3.5). We analyze the impact of each component in these three designs on the fine-tuning performance as shown in Table 7. Upon the original DETReg, we introduce contrastive alignment loss and the enhanced embeddings from the mutual enhancement module. These two components improve the baseline performance to varying degrees. Subsequently, we feed the enhanced feature to DETR decoder, which significantly enhances fine-tuning performance, mainly improving by 3.4% at the 12^{nd} epoch in AP_{50}. Next, we apply additional contrastive alignment loss to the encoder. Although it does not directly improve performance, it synergizes with the subsequent auxiliary siamese head to further improve fine-tuning performance. Table 7 reveals that siamese head also contributes to the improvement, possibly due to introducing multiple supervision [7]. Ultimately, the combination of all designs in MutDet achieves optimal performance.

Table 7: Effect of different components. Models are evaluated on DIOR-R dataset, using DOTA-v1.0 as pre-training dataset, and ARS-DETR as detector. The first line represents the DETReg baseline. Superscript denotes the improvement compared to pre-training free.

Contrastive Loss	Enhanced Embedding	Enhanced Feature	Encoder Loss	Siamase Head	#Epoch 12	24	36
					62.1	65.1	67.9
✓					$62.6^{+0.5}$	$65.8^{+0.7}$	$68.5^{+0.6}$
✓	✓				$62.8^{+0.7}$	$65.9^{+0.8}$	$68.7^{+0.8}$
✓	✓	✓			$65.5^{+3.4}$	$67.0^{+1.9}$	$69.5^{+1.6}$
✓	✓	✓	✓		$65.5^{+3.4}$	$67.4^{+2.3}$	$69.4^{+1.5}$
✓	✓			✓	$64.4^{+2.3}$	$66.7^{+1.6}$	$69.1^{+1.2}$
✓	✓		✓	✓	$66.3^{+4.2}$	$68.4^{+3.3}$	$69.9^{+2.0}$
✓	✓	✓	✓	✓	**66.9**$^{+4.8}$	**69.8**$^{+4.7}$	**70.7**$^{+2.8}$

Table 8: Comparison results with different backbones and pre-training datasets. The performances on the test sets are reported. Swin-Tiny denotes the smallest version of the Swin Transformer [23]. RsDet4 is the collected large-scale pre-training dataset, including DOTA, DIOR, FAIR-1M-2.0, and HRRSD. '-' indicates pre-training free. **Red**: optimal results. **Blue**: sub-optimal results.

Method	Backbone	Pre-training Dataset	Fine-tuning Dataset	#Epoch 12	24	36
-	ResNet-50	-	DIOR	55.4	61.5	65.7
DETReg [1]	ResNet-50	DOTA-v1.0		62.1	65.1	67.9
MutDet (Ours)	ResNet-50	DOTA-v1.0		66.9	69.8	70.7
-	Swin-Tiny	-		58.1	64.9	70.0
DETReg [1]	Swin-Tiny	RSDet4		**68.7**	**70.5**	**73.2**
MutDet (Ours)	Swin-Tiny	RSDet4		**71.5**	**70.5**	**73.7**
-	ResNet-50	-	DOTA-v1.0	69.0	71.1	72.9
DETReg [1]	ResNet-50	DOTA-v1.0		70.0	72.5	72.6
MutDet (Ours)	ResNet-50	DOTA-v1.0		73.8	73.3	74.5
-	Swin-Tiny	-		70.8	74.0	75.6
DETReg [1]	Swin-Tiny	RSDet4		**74.7**	**76.3**	**76.7**
MutDet (Ours)	Swin-Tiny	RSDet4		**75.5**	**76.0**	**76.9**

Advanced Backbone and Larger Pre-training Dataset. We replace the detector backbone ResNet-50 with Swin-Tiny [23] and use the large-scale RSDet4 dataset for pre-training. Table 8 demonstrates that pre-training methods remain effective even with a stronger backbone and larger pre-training dataset. Different detection pre-training methods achieve similar performance, *e.g.*, at 36^{th} epoch, MutDet outperforms DETReg by only 0.5% on DIOR-R and by 0.2% on DOTA-v1.0. The differences between pre-training methods are more pronounced when training time is limited, *e.g.*, at 12^{nd} epoch, MutDet surpasses DETReg by 2.8% on DIOR-R and by 0.8% on DOTA.

Different Detection methods. In addition to ARS-DETR, we also test three variations of Deformable-DETR (D-DETR): Rotated D-DETR [47], CSL D-DETR [41], and AR-CSL D-DETR [41]. We adopt the same training settings

Table 9: Comparison results with different detection methods on DIOR-R. '-' indicates pre-training free. **Red**: optimal results. **Blue**: sub-optimal results.

Methods	Rotated D-DETR		CSL D-DETR		AR-CSL D-DETR	
	AP_{50}	AP_{75}	AP_{50}	AP_{75}	AP_{50}	AP_{75}
-	38.3	23.7	62.2	41.2	63.6	42.1
UP-DETR [11]	38.8	24.5	64.7	45.1	65.0	**45.6**
AlignDet [21]	40.3	24.9	61.4	41.3	63.1	41.7
DETReg [1]	**41.4**	**26.8**	**65.6**	**45.6**	**65.2**	44.9
MutDet (Ours)	**45.8**	**28.0**	**68.0**	**47.3**	**68.3**	**47.6**

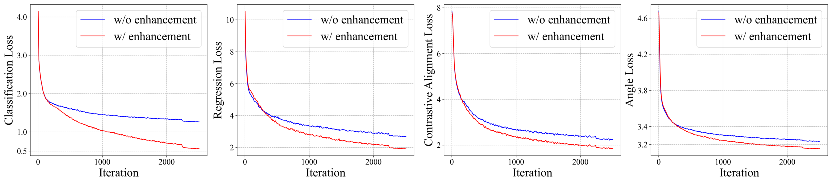


Fig. 4: Loss curves during the pre-training on DOTA-v1.0 dataset. The red curves are the losses when the mutual enhancement module is employed in pre-training, and the blue curves are the losses when not employed. Four losses are included: classification, regression, contrastive alignment, and angle prediction.

as ARS-DETR, and the results are shown in Table 9. Our MutDet outperforms other pre-training methods on all detection methods. MutDet surpasses DETReg by 4.4% on Rotated D-DETR, 2.4% on CSL D-DETR, and 3.1% on AR-CSL D-DETR in AP_{50}, demonstrates its adaptability across different detectors.

Effect of Mutual Enhancement Module. We compare the convergence of classification, regression, embedding alignment, and angle prediction tasks between using mutual enhancement module or not, as shown in Figure 4. After several training epochs, it is observed that the detector with the enhancement module converges more rapidly across all tasks. The phenomenon suggests that the detector can effectively acquire visual knowledge from the object embeddings through the enhancement module.

5 Conclusion

In this work, we propose a novel pre-training framework for remote sensing detection. Our MutDet addresses the feature discrepancy issue in previous methods through mutual optimization, effectively improving the performance of downstream detection tasks. In MutDet, we introduce SAM to generate pseudo labels to enhance the recall of remote sensing objects and achieve rotation annotation. However, MutDet only unidirectionally utilizes SAM and fails to exploit its potential fully. Our future work will consider establishing a more effective correlation between the detector and the underlying visual model. Our work lays a solid foundation for future pre-training research in remote sensing detection.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant 62176017, and the Fundamental Research Funds for the Central Universities.

References

1. Bar, A., Wang, X., Kantorov, V., Reed, C.J., Herzig, R., Chechik, G., Rohrbach, A., Darrell, T., Globerson, A.: Detreg: Unsupervised pretraining with region priors for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14605–14615 (2022)
2. Bouniot, Q., Audigier, R., Loesch, A., Habrard, A.: Proposal-contrastive pretraining for object detection from fewer data. In: International Conference on Learning Representations (2023)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
4. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Advances in neural information processing systems **33**, 9912–9924 (2020)
5. Chang, J., Wang, S., Xu, H.M., Chen, Z., Yang, C., Zhao, F.: Detrdistill: A universal knowledge distillation framework for detr-families. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6898–6908 (2023)
6. Chen, D., Mei, J.P., Zhang, H., Wang, C., Feng, Y., Chen, C.: Knowledge distillation with the reused teacher classifier. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11933–11942 (2022)
7. Chen, Q., Chen, X., Wang, J., Zhang, S., Yao, K., Feng, H., Han, J., Ding, E., Zeng, G., Wang, J.: Group detr: Fast detr training with group-wise one-to-many assignment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6633–6642 (2023)
8. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: 2021 IEEE/CVF International Conference on Computer Vision. pp. 9620–9629 (2021)
9. Cheng, G., Wang, J., Li, K., Xie, X., Lang, C., Yao, Y., Han, J.: Anchor-free oriented proposal generator for object detection. IEEE Transactions on Geoscience and Remote Sensing **60**, 1–11 (2022)
10. Dai, L., Liu, H., Tang, H., Wu, Z., Song, P.: Ao2-detr: Arbitrary-oriented object detection transformer. IEEE Transactions on Circuits and Systems for Video Technology **33**(5), 2342–2356 (2023)
11. Dai, Z., Cai, B., Lin, Y., Chen, J.: Up-detr: Unsupervised pre-training for object detection with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1601–1610 (2021)
12. Ding, J., Xue, N., Long, Y., Xia, G.S., Lu, Q.: Learning roi transformer for oriented object detection in aerial images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2849–2858 (2019)
13. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020)

14. Han, J., Ding, J., Xue, N., Xia, G.S.: Redet: A rotation-equivariant detector for aerial object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2786–2795 (2021)
15. Hu, Z., Gao, K., Zhang, X., Wang, J., Wang, H., Yang, Z., Li, C., Li, W.: Emo2-detr: Efficient-matching oriented object detection with transformers. IEEE Transactions on Geoscience and Remote Sensing (2023)
16. Huang, G., Laradji, I., Vazquez, D., Lacoste-Julien, S., Rodriguez, P.: A survey of self-supervised and few-shot object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(4), 4071–4089 (2022)
17. Huang, G., Li, W., Teng, J., Wang, K., Chen, Z., Shao, J., Loy, C.C., Sheng, L.: Siamese detr. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15722–15731 (2023)
18. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollar, P., Girshick, R.: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4015–4026 (October 2023)
19. Lee, H., Song, M., Koo, J., Seo, J.: Rhino: Rotated detr with dynamic denoising via hungarian matching for oriented object detection. arXiv preprint arXiv:2305.07598 (2023)
20. Li, K., Wan, G., Cheng, G., Meng, L., Han, J.: Object detection in optical remote sensing images: A survey and a new benchmark. ISPRS journal of photogrammetry and remote sensing **159**, 296–307 (2020)
21. Li, M., Wu, J., Wang, X., Chen, C., Qin, J., Xiao, X., Wang, R., Zheng, M., Pan, X.: Aligndet: Aligning pre-training and fine-tuning in object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6866–6876 (2023)
22. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
23. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
24. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019)
25. Ma, T., Mao, M., Zheng, H., Gao, P., Wang, X., Han, S., Ding, E., Zhang, B., Doermann, D.: Oriented object detection with transformer. arXiv preprint arXiv:2106.03146 (2021)
26. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 658–666 (2019)
27. Sun, X., Wang, P., Yan, Z., Xu, F., Wang, R., Diao, W., Chen, J., Li, J., Feng, Y., Xu, T., et al.: Fair1m: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. ISPRS Journal of Photogrammetry and Remote Sensing **184**, 116–130 (2022)
28. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International journal of computer vision **104**, 154–171 (2013)
29. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

30. Wang, J., Chen, Y., Zheng, Z., Li, X., Cheng, M.M., Hou, Q.: Crosskd: Cross-head knowledge distillation for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16520–16530 (2024)
31. Wang, Y., Tang, S., Zhu, F., Bai, L., Zhao, R., Qi, D., Ouyang, W.: Revisiting the transferability of supervised pretraining: an mlp perspective. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9183–9193 (2022)
32. Wei, F., Gao, Y., Wu, Z., Hu, H., Lin, S.: Aligning pretraining for detection via object-level contrastive learning. Advances in Neural Information Processing Systems **34**, 22682–22694 (2021)
33. Xia, G.S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L.: Dota: A large-scale dataset for object detection in aerial images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3974–3983 (2018)
34. Xie, X., Cheng, G., Wang, J., Yao, X., Han, J.: Oriented r-cnn for object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3520–3529 (2021)
35. Yang, C., Ochal, M., Storkey, A., Crowley, E.J.: Prediction-guided distillation for dense object detection. In: European Conference on Computer Vision. pp. 123–138. Springer (2022)
36. Yang, C., An, Z., Cai, L., Xu, Y.: Mutual contrastive learning for visual representation learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 3045–3053 (2022)
37. Yang, X., Yan, J.: On the arbitrary-oriented object detection: Classification based approaches revisited. International Journal of Computer Vision **130**(5), 1340–1365 (2022)
38. Yang, X., Zhang, G., Yang, X., Zhou, Y., Wang, W., Tang, J., He, T., Yan, J.: Detecting rotated objects as gaussian distributions and its 3-d generalization. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(4), 4335–4354 (2023)
39. Yang, X., Zhou, Y., Zhang, G., Yang, J., Wang, W., Yan, J., Zhang, X., Tian, Q.: The kfiou loss for rotated object detection. arXiv preprint arXiv:2201.12558 (2022)
40. Yu, Y., Da, F.: Phase-shifting coder: Predicting accurate orientation in oriented object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13354–13363 (2023)
41. Zeng, Y., Chen, Y., Yang, X., Li, Q., Yan, J.: Ars-detr: Aspect ratio-sensitive detection transformer for aerial oriented object detection. IEEE Transactions on Geoscience and Remote Sensing **62**, 1–15 (2024)
42. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L., Shum, H.Y.: DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In: The Eleventh International Conference on Learning Representations (2023)
43. Zhang, Y., Yuan, Y., Feng, Y., Lu, X.: Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection. IEEE Transactions on Geoscience and Remote Sensing **57**(8), 5535–5548 (2019)
44. Zhao, Z., Li, S.: Abfl: Angular boundary discontinuity free loss for arbitrary oriented object detection in aerial images. IEEE Transactions on Geoscience and Remote Sensing (2024)
45. Zheng, Z., Ye, R., Wang, P., Ren, D., Zuo, W., Hou, Q., Cheng, M.M.: Localization distillation for dense object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9407–9416 (2022)

46. Zhou, Y., Yang, X., Zhang, G., Wang, J., Liu, Y., Hou, L., Jiang, X., Liu, X., Yan, J., Lyu, C., et al.: Mmrotate: A rotated object detection benchmark using pytorch. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 7331–7334 (2022)
47. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: International Conference on Learning Representations (2021)