SET: Spectral Enhancement for Tiny Object Detection

Huixin Sun[1], **Runqi Wang**[2], **Yanjing Li**[1],
Linlin Yang[5], **Shaohui Lin**[6], **Xianbin Cao**[1*], **Baochang Zhang**[3,4*]
[1]School of Electronic Information Engineering, Beihang University, Beijing, China
[2]School of Computer Science and Technology, Beijing Jiaotong University
[3]School of Artificial Intelligence, Beihang University, Beijing, China
[4]Zhongguancun Laboratory, Beijing, China
[5]State Key Laboratory of Media Convergence and Communication,
Communication University of China, Beijing, China
[6]School of Computer Science and Technology, East China Normal University, Shanghai, China

Abstract

Deep learning has significantly advanced the object detection field. However, tiny object detection (TOD) remains a challenging problem. We provide a new analysis method to examine the TOD challenge through occlusion-based attribution analysis in the frequency domain. We observe that tiny objects become less distinct after feature encoding and can benefit from the removal of high-frequency information. In this paper, we propose a novel approach named ***S****pectral* ***E****nhancement for* ***T****iny object detection (SET), which amplifies the frequency signatures of tiny objects in a heterogeneous architecture. SET includes two modules. The Hierarchical Background Smoothing (HBS) module suppresses high-frequency noise in the background through adaptive smoothing operations. The Adversarial Perturbation Injection (API) module leverages adversarial perturbations to increase feature saliency in critical regions and prompt the refinement of object features during training. Extensive experiments on four datasets demonstrate the effectiveness of our method. Especially, SET boosts the prior art RFLA by 3.2% AP on the AI-TOD dataset.*

1. Introduction

Recent advances in Deep Neural Networks (DNNs) [14, 60] have significantly improved the object detection field [11, 22]. Despite the progress, Tiny Object Detection (TOD) remains a challenging problem. Tiny objects, characterized by their very limited pixel input, occupy areas equal to or less than 16×16 pixels [44]. Compared to traditional object detection [10, 21], generic object detectors often fail to manifest effectiveness on the TOD task. For instance, DINO [60], one of the most representative transformer-based object detector, achieves 37.6% AP on medium-sized objects but only 9.9% AP on very tiny objects on the AI-TOD [44] benchmark, Moreover, the vanilla FCOS [39] achieves only 2.5% AP on very tiny objects on the AI-TOD [44], which is far from sufficient to meet the demands of real-world applications, such as autonomous driving, maritime rescue, and traffic management.

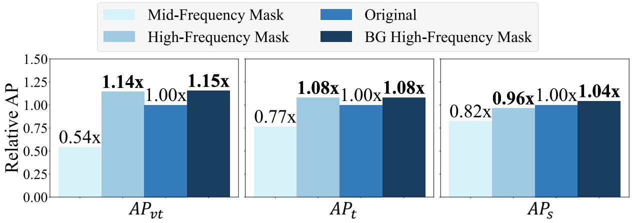

Figure 1. A feature-level occlusion-based attribution analysis in the frequency domain. Mid-frequency features ranges from 40% to 60% of the total energy, high-frequency features are above 90% of the total energy. The model is trained on the AI-TOD [44] `trainval` and validated on the AI-TOD `test`. Detection performance is evaluated by inferencing a 1× FCOS detector [39] with the masked features. Results exhibit that tiny objects are more likely to be affected by high-frequency noise and can benefit from the direct removal of high-frequency information.

Constrained by the inherently low resolution, one key challenge in TOD is to extract discriminative foreground features [7], especially in the high-level features of detection architecture after down-samplings. In the meantime, it is difficult to learn about tiny objects from the noisy and dominant background clutter [54]. Most existing works [1, 3, 30, 54] focus on enhancing feature representations of tiny objects in the spatial domain. We provide a new analysis method and address the challenge from a frequency spectrum perspective.

[*]Corresponding authors. Email: `sunhuixin@buaa.edu.cn`

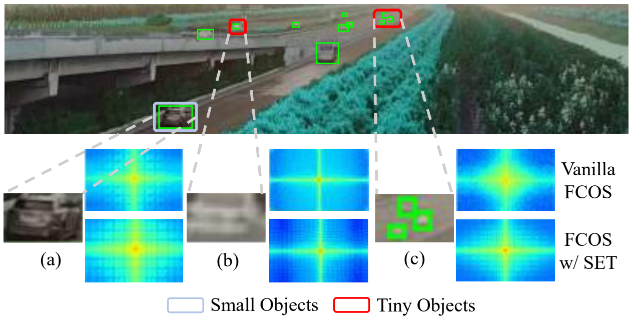

Figure 2. The analysis examines different objects after feature encoding in the frequency domain. The frequency spectrum is calculated using the Fast Fourier Transform (FFT) [31] on the cropped P_3 feature and selecting the channel with the highest energy. The frequency signatures of tiny objects become more pronounced after applying SET, manifesting distinct frequency response on the edges of the spectrum compared to vanilla FCOS.

Motivated by occlusion-based attribution analysis [33, 45], we perform an occlusion-based analysis on the intermediate Feature Pyramid Network (FPN) features to examine the impact of masking specific frequency bands on the detection performance across object scales. Specifically, we transform the features into the frequency domain using the n-dimensional Fast Fourier Transform (FFT) [31] and apply static filters to the amplitude spectrums, which are then fed to Inverse Fast Fourier Transform (IFFT) with phase spectrums to generate the masked spatial features. **High-frequency mask** results in Fig. 1 reveal that directly removing high-frequency features has a positive effect on very tiny and tiny objects, while negatively affecting the detection of larger objects. The disparity can be attributed to that tiny objects become less distinct after feature encoding, exhibiting weak high-frequency components (example (b), Fig. 2) and extremely vague frequency signatures in the cluttered background (example (c), Fig. 2). Therefore, they are more likely to be affected by high-frequency noise and benefit from the direct removal of high-frequency information. In contrast, larger objects exhibit distinct high-frequency signatures (example (a), Fig. 2) and have a higher dependency on high-frequency information. Based on the observations, we investigate an explicit measure to enhance the frequency signatures for tiny objects by removing the high-frequency information in the background. **Background (BG) high-frequency mask** results show that the approach enhances detection performance across all three object scales, resulting in improvements of 15%, 8%, and 4%, bringing significant gains for very tiny and tiny objects without compromising larger objects.

In light of the analysis above, we propose a generic **S**pectral **E**nhancement (SET) method to enhance the frequency signatures of **T**iny objects using a heterogeneous architecture for foreground and background feature refinement. The method includes two modules. As illustrated in Fig. 3, the first Hierarchical Background Smoothing (HBS) module suppresses the high-frequency information in the background through adaptive smoothing operations while preserving foreground details, thereby accentuating the frequency signatures of tiny objects. The second Adversarial Perturbation Injection (API) module leverages adversarial perturbations to increase feature saliency in critical regions and prompt the refinement of object features during training. In addition, API facilitates robust feature representations through adversarial training. SET is simple yet effective and can be easily mounted on existing detectors during the training process while bringing no extra burden to the inference procedure.

Our major contributions in this paper are summarized as:

- A feature-level occlusion-based attribution analysis from the frequency spectrum perspective is conducted to investigate the Tiny Object Detection (TOD) challenge, which shows that tiny objects are more likely to be affected by high-frequency noise. We introduce a spectral enhancement scheme for tiny object detection (SET) by designing a heterogeneous architecture for foreground and background feature refinement.
- Two new modules are designed for the TOD task. The HBS module is used to suppress the high-frequency noise in the background through adaptive smoothing operations. The API module leverages adversarial perturbations to increase feature saliency in critical regions and prompt the refinement of object features during training.
- Extensive results reveal that our SET boosts the well-known baselines and the prior art by a large margin.

2. Related Work

Tiny Object Detection. Most current approaches for tiny object detection (TOD) can be grouped into four main categories: data augmentation, multi-scale feature learning, label assignment, and feature enhancement strategies for tiny objects. Several data augmentation strategies [6, 37] target tiny objects by prompting the detector to focus on instances of specific scales during training. Conventional approaches also utilize multi-scale feature learning [23, 26, 62]. Another method line focuses on improving tiny object detection from the label assignment and proposal refinement perspectives [52, 58]. Various works aim to enhance the feature representations of tiny objects. Yuan *et al.* [58] propose a Feature Imitation (FI) mechanism, aligning RoI features with their counterparts in an exemplar feature set within the embedding space. Recent state-of-the-art TOD methods have proposed auxiliary self-reconstruction branches to enhance the weak representations of tiny objects [3, 54]. Differently, our proposed method enhances the discrimination of tiny objects from a frequency perspective, prompting the refinement of object features during training without adding any extra burden to the inference process.

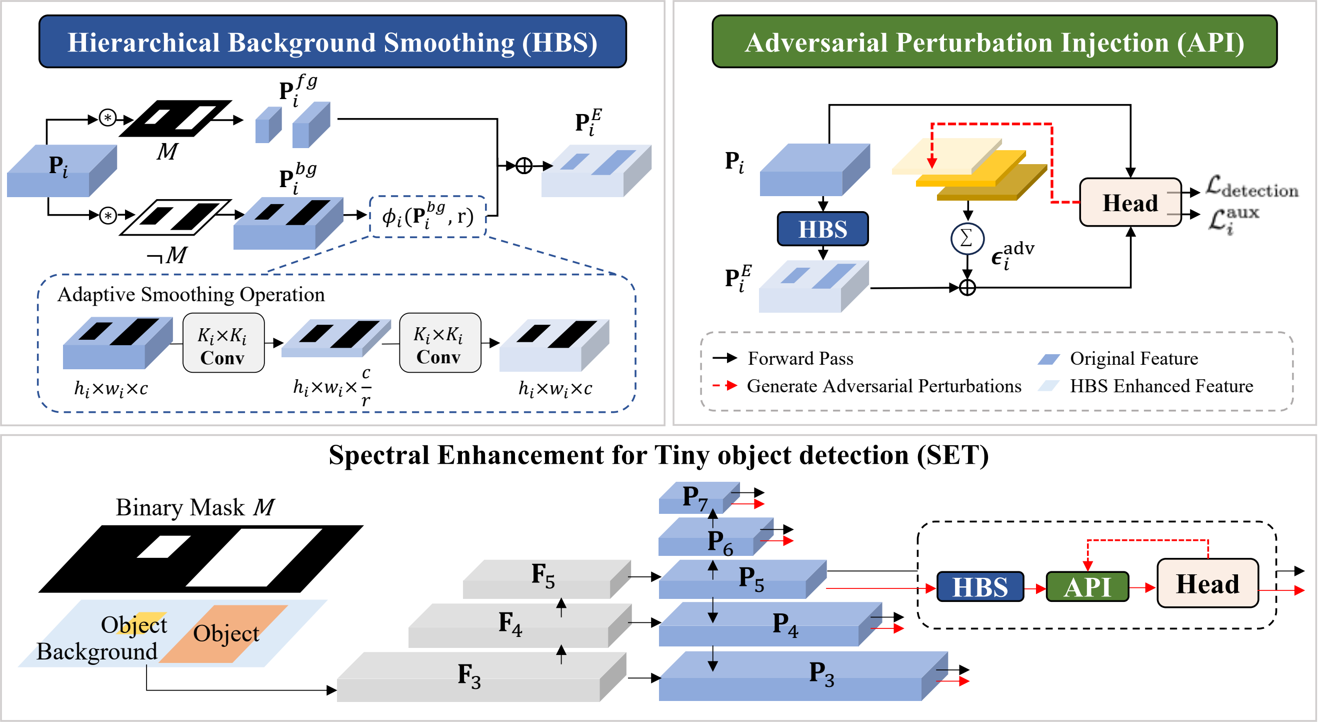
Hierarchical Background Smoothing (HBS)
$\mathbf{P}_i$
M
$\neg M$
$\mathbf{P}_i^{fg}$
$\mathbf{P}_i^{bg}$
$\phi_i(\mathbf{P}_i^{bg}, r)$
$\mathbf{P}_i^E$
Adaptive Smoothing Operation
$K_i \times K_i$ Conv
$K_i \times K_i$ Conv
$h_i \times w_i \times c$
$h_i \times w_i \times \frac{c}{r}$
$h_i \times w_i \times c$
Adversarial Perturbation Injection (API)
$\mathbf{P}_i$
HBS
$\mathbf{P}_i^E$
Σ
ϵ_i^{adv}
Head
$\mathcal{L}_{\text{detection}}$
$\mathcal{L}_i^{\text{aux}}$
Forward Pass
Generate Adversarial Perturbations
Original Feature
HBS Enhanced Feature
Spectral Enhancement for Tiny object detection (SET)
Binary Mask M
Object
Object
Background
$\mathbf{F}_5$
$\mathbf{F}_4$
$\mathbf{F}_3$
$\mathbf{P}_7$
$\mathbf{P}_6$
$\mathbf{P}_5$
$\mathbf{P}_4$
$\mathbf{P}_3$
HBS
API
Head

Figure 3. Overview of the proposed SET method with FCOS framework. The red dashed line denotes generating the adversarial perturbations ϵ_i^{adv}. The HBS module can suppress the high-frequency noise in the background through scale-wise smoothing operations. The API module leverages adversarial perturbations to increase feature saliency in critical regions and facilitate robust feature representations through adversarial training.

Adversarial Training for Object Detection. Many effective attacks crafted for object detectors have been proposed, with most generating adversarial examples at the image level [28, 38, 48]. The most common method to defend against the attacks is adversarial training [20, 29, 40], which involves augmenting training datasets with adversarial examples. For object detection, [59] extend adversarial training to the scenario of object detection by leveraging the attacks sourced from both classification and localization domains. Different from the aforementioned methods, we introduce feature-level adversarial perturbations that balance the detection semantics across branches for enhanced tiny object detection, resulting in more salient foreground features and improved model robustness.

3. Methods

This section presents the Spectral Enhancement (SET) method for Tiny object detection, depicted in Fig. 3. SET employs a heterogeneous architecture. This includes the Hierarchical Background Smoothing (HBS) module, which suppresses high-frequency information in the background, and the Adversarial Perturbation Injection (API) module, which enhances the feature saliency of critical regions and prompts the refinement of object features during training.

3.1. Hierarchical Background Smoothing

Based on the analysis in Sec. 1, we design a Hierarchical Background Smoothing (HBS) module to suppress the high-frequency information in the background feature while preserving foreground details, thereby accentuating the frequency signatures of tiny objects. This is achieved through background feature decoupling and adaptive smoothing, as described below.

Given an input image $X \in \mathbb{R}^{3\times W\times H}$, a binary mask M is generated according to the ground truth box B:

$$M_{i,j} = \mathbf{1}[(i,j) \in B], \tag{1}$$

where $M \in \{0,1\}^{W\times H}$. The indicative function $\mathbf{1}$ denotes the value of location (i,j), which is 1 if it belongs to an object and 0 otherwise. Then we utilize the generated binary mask to decouple image features and perform background smoothing, as shown in Fig. 3. In a Feature Pyramid Network (FPN) comprising N layers, the feature of the i-th layer is denoted as $\mathbf{P}_i, (i \in [1, N])$. The process of HBS on the i-th layer is as follows:

$$\begin{aligned} \mathbf{P}_i^E &= \mathbf{P}_i^{fg} + \phi_i(\mathbf{P}_i^{bg}, r) \\ &= \mathbf{P}_i \circledast M + \phi_i(\mathbf{P}_i \circledast \neg M, r), \end{aligned} \tag{2}$$

where $\mathbf{P}_i^E$, $\mathbf{P}_i^{fg}$ and $\mathbf{P}_i^{bg}$ denote the enhanced feature, foreground feature and background feature, $\neg M$ represents the

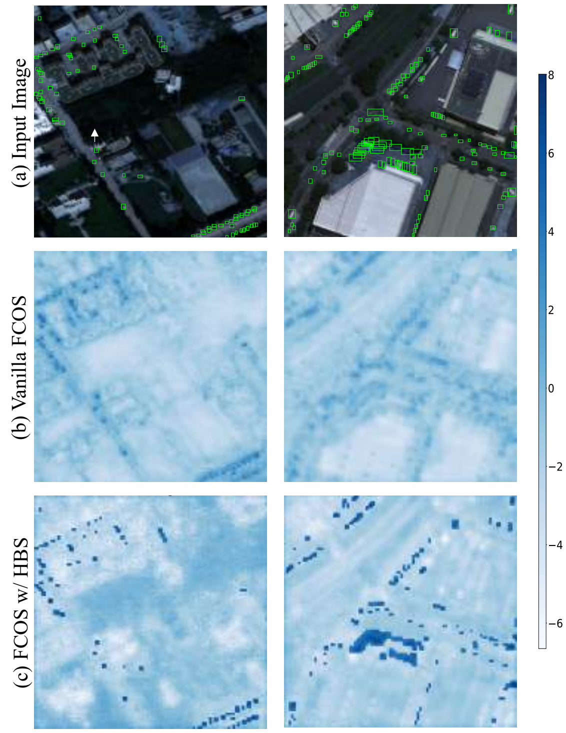

Figure 4. We visualize (a) the input image with ground-truths shown in green boxes, (b) P_3 feature in the vanilla FCOS after Principal Component Analysis (PCA) [46], and (c) P_3 feature in FCOS enhanced with HBS after PCA. Darker colors in the PCA maps indicate more critical components in the feature. As shown, HBS smooths the background regions and significantly enhances the contrast between the foreground and background.

complement of the binary mask M, and $\circledast$ is the Hadamard product with broadcasting to accommodate the dimensions. Previous works [18, 41] reveal that feature dimensionality reduction methods can lead to smoother and more efficient feature spaces. Based on this idea, we design an adaptive smoothing operation $\phi_i(\cdot, r)$ with channel reduction rate r, formulated as:

$$\phi_i(\mathbf{P}_i^{\mathrm{bg}}, r) = \sigma(\mathbf{w}_i^{\mathrm{e}} \otimes \sigma(\mathbf{w}_i^{\mathrm{r}} \otimes \mathbf{P}_i^{\mathrm{bg}})) + \mathbf{P}_i^{\mathrm{bg}}, \tag{3}$$

where $\otimes$ represents the convolution operation and $\mathbf{w}_i^{\mathrm{r}} \in \mathbb{R}^{C_i \times C_{i/r} \times K_i \times K_i}$ and $\mathbf{w}_i^{\mathrm{e}} \in \mathbb{R}^{C_{i/r} \times C_i \times K_i \times K_i}$ are the weights of the channel reducing and expanding kernels, respectively. σ denotes the non-linear function ReLU. The reduction step compresses information while retaining salient features. When features are expanded, the high-frequency information are difficult to be recreated, resulting in a smoother background. The effectiveness of smoothing is exhibited through statistical analysis in Sec. 4.7. The smoothing effect is further visualized through PCA analysis. As shown in Fig. 4, feature maps enhanced with HBS exhibit a reduced intensity range and smoother transitions in the background regions compared to the vanilla FCOS, effectively suppressing noise without losing essential background context.

Kernel size K_i at each scale level is determined based on the FPN stride S_i, where smaller kernels are utilized to remove the fine-grained details in lower feature layers and larger kernels for more coarse-grained noise in higher layers, determined as:

$$K_i = g(S_i) = \left(\left\lfloor \frac{\log_2(S_i)}{2} \right\rfloor \times 2 \right) + 1, \tag{4}$$

rounding up S_i to an odd integer and ensuring central pixel alignment in the convolutional kernel. The padding is determined as $(K_i - 1)/2$ to preserve the spatial dimensions of the feature map. In addition, we also tested other $g(S_i)$, and the results are shown in the Tab. 5. The HBS architecture allows for the suppression of high-frequency information in the background while preserving foreground details, significantly enhancing the frequency signatures of tiny objects amid high-frequency noise.

3.2. Adversarial Perturbation Injection

Analysis in Sec. 1 reveals that tiny objects exhibit weak high-frequency signatures after feature encoding and low discrimination against the cluttered background. To address this, we design an Adversarial Perturbation Injection (API) module, which leverages adversarial perturbations to enhance the saliency in critical regions. In addition, API promotes robust feature representations for tiny objects through adversarial training.

Inspired by prior works in adversarial training [16, 17, 42], we exploit feature-level perturbations that adversarially change the model output to enhance feature representations. The objective of API for each FPN layer is formulated as follows:

$$\min_{\mathbf{P}_i, \theta_i} \left(\max_{\|\epsilon_{i,\mathrm{cls}}\| \leq \rho} \mathcal{L}_{\mathrm{cls}}(\mathbf{P}_i + \epsilon_{i,\mathrm{cls}}) + \gamma \|\mathbf{P}_i\|_2^2 \right), \tag{5}$$

where $\mathcal{L}_{\mathrm{cls}}$ denotes the classification loss. The inner optimization injects adversarial perturbation $\epsilon_{i,\mathrm{cls}}$ into the feature space of $\mathbf{P}_i$, where ρ defines the perturbation size, and γ is a hyperparameter that controls the strength of regularization. θ_i denotes the model parameters at the i-th layer. We derive a closed-form solution for the adversarial perturbation under the L_2 norm following [12]:

$$\epsilon_{i,\mathrm{cls}}^{*} \approx \rho \cdot \frac{\nabla_{\mathbf{P}_i} \mathcal{L}_{\mathrm{cls}}(\mathbf{P}_i)}{\|\nabla_{\mathbf{P}_i} \mathcal{L}_{\mathrm{cls}}(\mathbf{P}_i)\|_2}. \tag{6}$$

The perturbation can elevate the gradient of critical regions, thereby increasing their saliency during training. The gradient change induced by the perturbation on the classification branch can be calculated as:

$$\Delta(\nabla_{\mathbf{P}_i} \mathcal{L}_{\mathrm{cls}}) = \left\| \nabla_{\mathbf{P}_i} \left(\mathcal{L}_{\mathrm{cls}}(\mathbf{P}_i + \epsilon_{i,\mathrm{cls}}^{*}) - \mathcal{L}_{\mathrm{cls}}(\mathbf{P}_i) \right) \right\|. \tag{7}$$

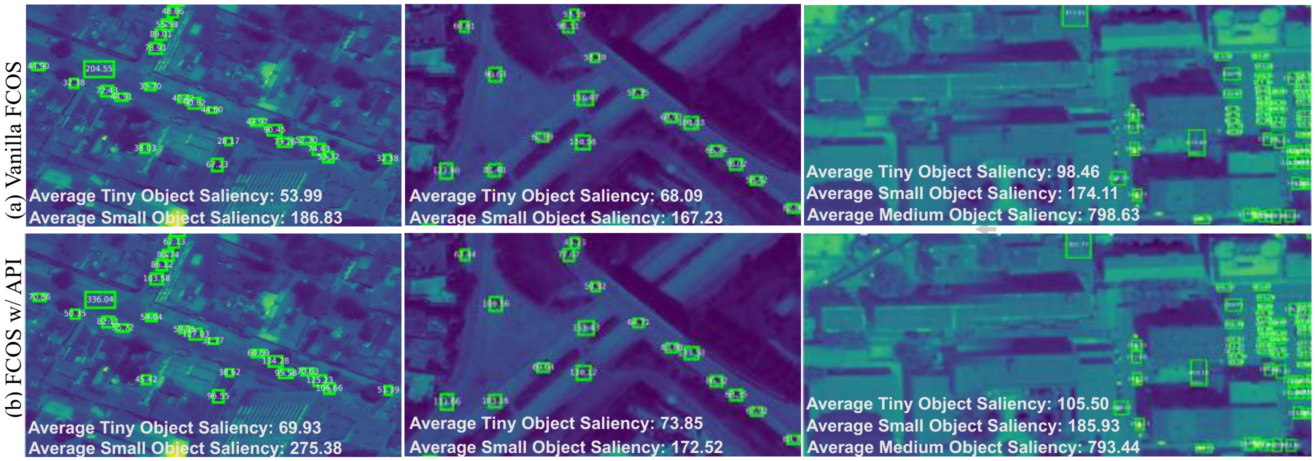
(a) Vanilla FCOS
Average Tiny Object Saliency: 53.99
Average Small Object Saliency: 186.83
Average Tiny Object Saliency: 68.09
Average Small Object Saliency: 167.23
Average Tiny Object Saliency: 98.46
Average Small Object Saliency: 174.11
Average Medium Object Saliency: 798.63
(b) FCOS w/ API
Average Tiny Object Saliency: 69.93
Average Small Object Saliency: 275.38
Average Tiny Object Saliency: 73.85
Average Small Object Saliency: 172.52
Average Tiny Object Saliency: 105.50
Average Small Object Saliency: 185.93
Average Medium Object Saliency: 793.44

Figure 5. Average saliency visualization in (a) FCOS and (b) FCOS w/ API. The saliency maps are first computed based on the ℓ_2-norm of the gradient [13] at neck feature P_3. The saliency of each object is then summed and annotated within their respective bounding box, which is different between (a) and (b). The average saliency of each object scale is calculated as the mean saliency across objects in the corresponding size category, displayed in the bottom-left corner of each image.

High $\Delta(\nabla_{\mathbf{P}_i}\mathcal{L}_{\text{cls}})$ corresponds to regions where model outputs are highly sensitive to minor changes in activations, indicating the presence of semantic information about the image [49]. This elevated gradient enhances the feature saliency of these critical regions and prompts the refinement of object features during training. The effect particularly benefits tiny objects as they exhibit weak high-frequency signatures and face a training deficiency problem. The enhanced object saliency phenomenon is discussed in Sec. 4.6 through Fig. 5. Furthermore, the adversarial training facilitates robust feature representations and reduces the sensitivity to high-frequency noise [63], thereby enhancing the discrimination of tiny objects.

We formulate the adversarial perturbation ϵ_i^{adv} for each FPN layer as an expectation over M object detection branches (e.g., the classification, regression and center-ness branch in a FCOS detector [39]), each scaled by a corresponding balancing parameter λ_m. This is generated as follows:

$$\epsilon_i^{\text{adv}} = \sum_{m=1}^{M} \lambda_m \cdot \epsilon_{i,m}^{*}, \tag{8}$$

where $\epsilon_{i,m}^{*}$ represents the adversarial perturbation for the m-th branch. The design allows for a balanced incorporation of detection semantics across multiple branches, ensuring that the perturbations are distributed for more effective training.

3.3. Auxiliary Optimization

The optimization leverages a total loss function that integrates both the original detection loss $\mathcal{L}_{\text{detection}}$ and an auxiliary loss derived from the HBS and API enhanced features. This joint loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{detection}} + \underbrace{\lambda \cdot \sum_{i=1}^{N} \mathcal{L}_i^{\text{aux}}(\mathbf{P}_i^E + \epsilon_i^{\text{adv}})}_{\text{auxiliary loss}}, \tag{9}$$

where $\mathcal{L}_i^{\text{aux}}$ denotes the auxiliary loss component for the i-th layer, which is summed and scaled by a balancing hyperparameter λ. The perturbation ϵ_i^{adv} is applied to the HBS-enhanced feature $\mathbf{P}_i^E$. The auxiliary optimization process updates the model's parameters in concert with the original loss, leading to more robust and discriminative features for tiny objects. We visualize the effectiveness of the method in Fig. 2. The frequency signatures of tiny objects become more pronounced after applying SET, manifesting distinct frequency response on the edges of the spectrum compared to vanilla FCOS, benefiting their detection amid high-frequency noise and in the presence of larger objects.

4. Experiments

4.1. Datasets and Implementation Details

Datasets. We evaluate the proposed method on **AI-TOD** [44], **VisDrone2019** [65], **DOTA-v2.0** [47] and **COCO** 2017 [22]. The main experiments were conducted using the AI-TOD dataset, with an average object size of 12.8 pixels, substantially smaller than other object detection datasets such as PASCAL VOC (156.6 pixels) [10] and MS COCO (99.5 pixels) [21]. Furthermore, we test our method on the VisDrone2019 [65] and DOTA-v2.0 [47], which comprises high-resolution drone-shot images with a substantial proportion of tiny instances. We also conduct experiments on the general object detection benchmark COCO 2017 [22], incorporating AP$_{vt}$, AP$_t$, AP$_s$ metrics to evaluate objects of very tiny (2-8 pixels), tiny (8-16 pixels) and small (16-32 pixels), following [44].

Table 1. Main results with various frameworks on AI-TOD [44]. Models are trained on the AI-TOD `trainval` and validated on the AI-TOD `test`. We report APs (%) with different IoU thresholds and APs (%) for objects of various sizes based on the AI-TOD criterion. RFLA is based on Cascade R-CNN. The * denotes using P2$\sim$P6 FPN features. The **bold** indicates the best result.

Framework	AP	AP$_{0.5}$	AP$_{0.75}$	AP$_{vt}$	AP$_{t}$	AP$_{s}$
PAA [19]	10.0	26.5	6.7	3.5	10.5	13.1
ATSS [61]	11.6	28.5	7.6	2.5	11.9	15.9
Centernet [9]	13.4	39.2	5.0	3.8	12.1	17.7
DetectoRS [34]	14.8	32.8	11.5	0.0	10.8	18.3
DotD [50]	16.1	39.2	10.6	8.3	17.6	18.1
DetectoRS w/NWD [43]	20.8	49.3	14.3	6.4	19.7	29.6
DetectoRS w/ SR-TOD [3]	24.0	54.6	17.1	**10.1**	24.8	29.3
One-stage						
RetinaNet [24]	7.2	20.5	3.6	2.5	6.8	11.0
RetinaNet w/ SET	**9.1**	**24.6**	**5.2**	**2.9**	**8.2**	**12.8**
RetinaNet* [24]	13.3	31.7	8.6	3.9	14.4	17.7
RetinaNet* w/ QueryDet [55]	12.2	29.3	7.3	2.4	10.5	18.5
RetinaNet* w/ SET	**16.5**	**38.6**	**10.9**	**5.0**	**15.7**	**21.8**
FCOS [39]	12.0	29.0	8.0	2.5	11.9	17.1
FCOS w/ SET	**14.2**	**34.9**	**9.8**	**2.9**	**13.0**	**20.2**
FCOS* [39]	15.1	35.8	10.2	5.9	16.6	18.8
FCOS* w/ SET	**18.0**	**42.7**	**12.4**	**7.2**	**18.5**	**23.0**
Multi-stage						
Faster R-CNN [35]	11.1	26.3	8.1	0.0	7.2	23.3
Faster R-CNN w/ SET	**12.2**	**28.2**	**9.7**	**0.1**	**9.1**	**24.3**
Cascade R-CNN [2]	13.6	30.3	10.6	0.0	9.9	25.5
Cascade R-CNN w/ SET	**14.8**	**32.4**	**11.2**	**0.4**	**10.8**	**26.7**
RFLA [52]	21.7	50.5	15.3	8.3	21.8	24.5
RFLA w/ SR-TOD [3]	21.8	50.8	15.4	9.7	21.8	27.4
RFLA w/ SET	**24.9**	**55.6**	**17.9**	**9.8**	**25.2**	**31.0**
Transformer-based						
DINO-5scale [60]	23.2	56.6	15.4	9.9	23.1	29.3
DINO-5scale w/ DNTR [25]	26.2	56.7	20.2	12.8	26.4	31.0
DINO-5scale w/ SET	**26.6**	**57.1**	**20.8**	**13.2**	**27.1**	**31.5**

Implementation Details. We conducted the experiments on a computer with an NVIDIA RTX 3090 GPU. The experiments are implemented with PyTorch [32], with core codes built upon `MMdetection` [5]. The ImageNet [36] pre-trained models are used as the backbones. All CNN-based models utilize the ResNet-50 [15] backbone, trained using the Stochastic Gradient Descent (SGD) optimizer for 12 epochs with 0.9 momentum, 0.0001 weight decay, and a batch size 2. The initial learning rate is 0.005, decaying at the 8th and 11th epochs. The data processing adheres to the default configurations of each dataset (e.g, fixed at 800×800 for AI-TOD). We also train a transformer-based detector, DINO [60], with 5-scale feature maps for 36 epochs as a baseline. The training uses an Adam optimizer with a weight decay of 0.0001, following the random crop and scale augmentation strategies of DETR [4].

4.2. Results on AI-TOD

We evaluate the proposed SET method on various detectors and compare it with state-of-the-art TOD methods on the AI-TOD benchmark [51]. Tab. 1 shows that SET enhances all baselines by about 2% AP, which is significant. Specifically, SET improves the one-stage detector FCOS [39] and RetinaNet [24] by 2.2%/1.9% AP and 1.1%/1.4% AP$_t$, respectively. We further revise the detector architecture and use P2$\sim$P6 FPN features, a representative approach for TOD tasks where the high-resolution P2 layer benefits tiny objects. With the adjustment, SET boosts the RetinaNet and FCOS baselines by 3.2%/2.9% AP and 1.1%/1.3%AP$_{vt}$, a significant margin. In contrast, the state-of-the-art SOD method QueryDet [55] decreases the baseline performance, showing limitations in addressing the TOD challenge. SET also generalizes to multi-stage detectors, enhancing the Faster R-CNN [35] and Cascade R-CNN [2] baselines by 1.1%/1.2% AP and 1.9%/1.1% AP$_t$ respectively.

SET works orthogonally with the state-of-the-art TOD method RFLA [52], achieving a 3.2% AP increment over the RFLA baseline and surpassing the comparison SR-TOD [3] by 3.1% AP and 3.4% AP$_t$. We also investigate SET's compatibility with transformer-based detectors. As shown, SET results in 3.4% AP increase over the DINO-5scale [60] baseline, outperforming the prior

Table 2. Detection performance on the VisDrone2019 [65] validation set. RFLA is based on Cascade R-CNN. The **bold** denotes the best result.

Framework	AP	AP$_{vt}$	AP$_t$	AP$_s$
RetinaNet [24]	15.2	0.4	1.6	9.3
Faster R-CNN [35]	23.1	0.1	5.1	19.6
FCOS [39]	19.9	0.7	4.5	15.7
FCOS w/ SET	**21.9**	**1.6**	**5.7**	**17.9**
Cascade R-CNN [2]	23.0	0.1	5.0	19.4
Cascade R-CNN w/ SET	**24.3**	**1.2**	**6.3**	**21.0**
RFLA [52]	27.2	4.5	13.0	23.6
RFLA w/ SR-TOD [3]	27.8	4.8	13.1	24.5
RFLA w/ SET	**28.5**	**5.2**	**13.5**	**24.6**

art DNTR [25]. Notably, DINO-5scale w/SET derives 26.6% AP, outperforming competitors including DotD [50], NWD [43], and SR-TOD [3]. The results demonstrate SET as a robust solution for the tiny object detection task.

4.3. Results on VisDrone2019 and DOTA-V2.0

We further evaluate SET on the drone-shot image benchmark VisDrone2019 [65], which features tiny and small objects. As shown in Tab. 2, the proposed SET effectively improves the performance of both one-stage and multi-stage detectors and exhibits consistent AP$_{vt}$ and AP$_t$ improvements. Specifically, SET improved the the one-stage detector FCOS by 2.0%/0.9%/1.2% AP, AP$_{vt}$ and AP$_t$, respectively. SET also improves multi-stage detector Cascade R-CNN [2] by 1.1%/1.3% AP$_{vt}$ and AP$_t$, respectively. Notably, SET improves the prior art RFLA, achieving a 28.5% AP with 0.7% increment in AP$_{vt}$ and 0.5% in AP$_t$.

Tab. 3 demonstrates the DOTA-v2.0 [8] results. As shown, SET obtains a substantial performance gain of 3.1% and 1.3 % in AP$_t$ with the FCOS and AutoAssign baseline, underscoring the efficacy of our method in detecting tiny objects. Compared to the state-of-the-art comparisons RFLA [52] and DCFL [53], SET achieves 2.6% and 0.9% AP gain, respectively. More experimental results on the **TinyPerson** [56] and **SeaPerson** [57] dataset are provided in the supplementary material.

4.4. Results on COCO

We further conduct experiments on the general object detection benchmark MS COCO [21]. As shown in Tab. 9, SET achieves 1.0% AP improvement over the FCOS baseline, with a notable 1.7% improvement in AP$_t$, demonstrating that SET can enhance the detection of tiny objects in general detection tasks.

4.5. Ablation Study

In the following experiments, we explore the best structure of SET by ablating and tuning its parts using AI-TOD.

Table 3. Detection performance on DOTA-v2.0 [8]. Note that models are trained on the DOTA-v2.0 `train` and validated on the DOTA-v2.0 `val`. The **bold** denotes the best result.

Framework	AP	AP$_{vt}$	AP$_t$	AP$_s$
ATSS [61]	32.7	0.7	6.9	23.4
Faster R-CNN [35]	35.6	0.0	7.1	28.9
FCOS [39]	31.8	0.3	4.0	19.4
FCOS w/ RFLA [52]	32.1	**0.7**	6.8	23.6
FCOS w/ DCFL [53]	33.8	0.5	6.9	24.2
FCOS w/ SET	**34.7**	0.6	**7.1**	**24.8**
AutoAssign [64]	33.8	0.9	7.3	22.4
AutoAssign w/ SET	**35.2**	**1.1**	**8.6**	**23.6**

Table 4. Detection performance on MS COCO [21]. Note that models are trained on COCO `train2017` and validated on COCO `val2017`. $(\cdot)^+$ denotes performance increment.

Framework	AP	AP$_{vt}$	AP$_t$	AP$_s$	AP$_m$
FCOS [39]	36.4	7.9	19.6	27.2	43.6
FCOS w/ SET	**37.4**$^{+1.0}$	**8.3**$^{+0.4}$	**23.3**$^{+1.7}$	**27.6**$^{+0.4}$	**44.8**$^{+1.2}$

Effectiveness and Cost of Components. We present quantitative improvements of SET components in Tab. 5. As shown, applying the adaptive smoothing operations to full features ($+\phi_i(\mathbf{P}_i, r)$) achieves a 0.4% increase in AP and 0.3% in AP$_{vt}$, surpassing the performance of static high-frequency filtering in Fig. 1. With decoupling, the HBS module boosts the overall AP by 1.9% AP and 0.4% AP$_{vt}$. The API module alone achieves a 1.2% AP improvement. Combining HBS and API results in a total AP increase of 2.2%. Results demonstrate the synergy between the heterogeneous operations, where HBS effectively smooths the clutter background and API prompts the learning of object features, cumulatively improving the detection accuracy of tiny objects. Tab. 5 also presents the cost of components. Results show that the HBS module leads to 17%, 0.9%, and 0.9% increase in training time, computational and memory cost. The API module results in a more significant increase in training time due to adversarial training but adds no forward computational cost and minimal increase in memory usage. The overall minimal impact indicates that SET can be implemented with negligible additional overhead.

Hyper-parameter Selection. We first select the hyper-parameters channel reduction rate r and perturbation size ρ using the FCOS [39] framework trained three epochs in Fig. 6 (a). Model performance (AP) with different hyper-parameter combinations $\{r,\, \rho\}$ is presented. Results indicate that performance initially improves and then declines as r varies from 2 to 16, with $r = 16$ perform worse than the baseline. Additionally, SET with adversarial perturbations ($\rho > 0$) exhibits stronger performance than without ($\rho = 0$), but large-sized perturbations ($\rho = 5$) performs worse than all alternatives. Exploring the setups, we identify $\{r,\, \rho\}$ = $\{4,\, 1\}$ as the combination for SET. Fig. 6 (b) explores the

Table 5. The impact of different components and various channel reduction operations in SET. The hyper-parameters are fixed with channel reduction rate r being 4, perturbation size ρ being 1, and the balancing hyper-parameters λ of auxiliary optimization being 1.

Framework	ϕ_i	$g(S_i)$	AP	AP$_{vt}$	AP$_t$	Time$_{\mathrm{(s/batch)}}$	GFLOPs	Params (M)
FCOS [39]	-	-	12.0	2.5	11.9	0.17	126.13	37.04
+ $\phi_i(\mathbf{P}_i, r)$	-	-	12.7	2.7	12.9	0.17	127.36	37.41
+ HBS	Eqn. 3	3	13.4	2.7	12.6	0.20	127.32	37.41
	Eqn. 3	$\left\lfloor \frac{\log_2(S_i)}{2} \right\rfloor \times 2 + 1$	13.9	2.7	12.9			
	Eqn. 3	5	13.1	2.3	12.5			
	Eqn. 3	$\left\lfloor \frac{\log_2(S_i+1)}{2} \right\rfloor \times 2 - 1$	13.4	2.8	12.4			
	Eqn. 3	7	12.1	2.5	12.2			
+ API	-	-	13.2	2.6	12.4	0.39	126.13	37.16
+ Both (SET)	Eqn. 3	Eqn. 4	**14.2**	**2.9**	**13.0**	0.42	127.32	37.85

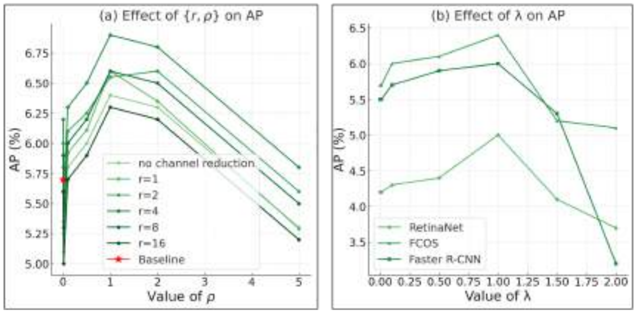

Figure 6. We select hyper-parameters channel reduction rate r with perturbation size ρ in (a), and the balancing hyper-parameter λ of Eqn. 9 in (b).

balancing hyper-parameter λ in Eqn. 9 that controls the proportion of the auxiliary optimization, using the FCOS [39], RetinaNet [24], and Faster R-CNN [35] framework. As observed, the performances increase first and then decrease with the increase of λ, with $\lambda = 1$ yielding the best performance across all frameworks. However, the efficacy of SET diminishes when λ exceeds the 1, as larger-sized λ ($\lambda > 1$) can impose too strong regularization to the detector, reducing the detection accuracy despite increasing robustness. Therefore, we set the hyper-parameter λ to 1 for experiments in this paper.

4.6. Enhanced Object Saliency

We demonstrate effectiveness of API in increasing the saliency of object features in Fig. 5. Comparing the average saliency of each object scale, tiny objects are the least salient due to their inherently limited pixel input. API leads to a notable increase in the average saliency of tiny objects (e.g., from 53.99 to 69.93), benefited from the injection of perturbations (Eqn. 7). Fig. 5 also demonstrates that API can enhance the feature saliency of small and medium objects, though the degree of improvement is less. The comparison underscores that API could benefit general object detection, with more significant and essential improvements for tiny objects.

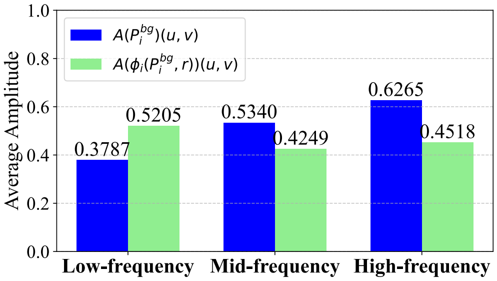

Figure 7. The analysis examines the adaptive smoothing operation, obtained from 1000 randomly selected images from the VisDrone2019 [65] train set.

4.7. Effectiveness in Smoothing

In Fig. 7, we perform a statistic analysis on P_i^{bg} to investigate the adaptive smoothing operation. We convert feature P_i^{bg} and $\phi_i(P_i^{bg}, r)$ into the frequency space using FFT [31] and separate the amplitude spectrums of these features into three parts (low-, mid-, and high-frequency) using static thresholds as Fig. 1. The amplitudes in each band are then summed and averaged to measure the signal strength. Results verify that the channel reducing and expand design effects in smoothing. Comparing $A(P_i)(u, v)$ and $A(\phi_i(P_i^{bg}, r))(u, v)$, the low-frequency signals are enhanced (from 0.3787 to 0.5205), while mid- and high-frequency signals are attenuated.

5. Conclusion

This paper proposes the Spectral Enhancement for Tiny objects (SET) method, which amplifies the frequency signatures of tiny objects in a heterogeneous architecture. SET includes two modules. The Hierarchical Background Smoothing (HBS) module suppresses high-frequency noise in the background through adaptive smoothing operations. The Adversarial Perturbation Injection (API) module leverages adversarial perturbations to prompt the refinement of object features during training. Extensive experiments on four datasets demonstrate its effectiveness.

Acknowledgments. The work was supported by the National Natural Science Foundation of China (No. 61827901, 623B2016, and 62406298). This research was also supported by the National Key Research and Development Program of China (Grant No. 2023YFC3306401), Zhejiang Provincial Natural Science Foundation of China under Grant No. LD24F020007, Beijing Natural Science Foundation L223024, L244043 and Z241100001324017, "One Thousand Plan" projects in Jiangxi Province Jxsq2023102268, and the Beijing Jiaotong University "Jingying Plan" No. K24XKRC00130.

References

[1] Yancheng Bai, Yongqiang Zhang, Mingli Ding, and Bernard Ghanem. Sod-mtgan: Small object detection via multi-task generative adversarial network. In *ECCV*, pages 206–221, 2018. 1

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018. 6, 7

[3] Bing Cao, Haiyu Yao, Pengfei Zhu, and Qinghua Hu. Visible and clear: Finding tiny objects in difference map. *ECCV*, 2024. 1, 2, 6, 7

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 6

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[6] Yukang Chen, Peizhen Zhang, Zeming Li, Yanwei Li, Xiangyu Zhang, Lu Qi, Jian Sun, and Jiaya Jia. Dynamic scale training for object detection. *arXiv preprint arXiv:2004.12432*, 2020. 2

[7] Gong Cheng, Xiang Yuan, Xiwen Yao, Kebing Yan, Qinghua Zeng, Xingxing Xie, and Junwei Han. Towards large-scale small object detection: Survey and benchmarks. *IEEE TPAMI*, 2023. 1

[8] Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Michael Ying Yang, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, et al. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE TPAMI*, 44(11):7778–7796, 2021. 7

[9] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, pages 6569–6578, 2019. 6

[10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *ICCV*, 88:303–308, 2009. 1, 5

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2): 303–338, 2010. 1

[12] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020. 4, 1

[13] Jianyuan Guo, Kai Han, Yunhe Wang, Han Wu, Xinghao Chen, Chunjing Xu, and Chang Xu. Distilling object detectors via decoupled features. In *CVPR*, pages 2154–2164, 2021. 5, 1

[14] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *CVPR*, pages 5353–5360, 2015. 1

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[16] Yanxu Hu and Andy J Ma. Adversarial feature augmentation for cross-domain few-shot classification. In *ECCV*, pages 20–37. Springer, 2022. 4

[17] Masato Ishii and Atsushi Sato. Training deep neural networks with adversarially augmented features for small-scale training datasets. In *IJCNN*, pages 1–8. IEEE, 2019. 4

[18] Weikuan Jia, Meili Sun, Jian Lian, and Sujuan Hou. Feature dimensionality reduction: a review. *Complex & Intelligent Systems*, 8(3):2663–2693, 2022. 4

[19] Kang Kim and Hee Seok Lee. Probabilistic anchor assignment with iou prediction for object detection. In *ECCV*, pages 355–371, 2020. 6

[20] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In *ICLR*, 2016. 3

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 1, 5, 7, 3

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 1, 5

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 2

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 6, 7, 8, 2, 3

[25] Hou-I Liu, Yu-Wen Tseng, Kai-Cheng Chang, Pin-Jyun Wang, Hong-Han Shuai, and Wen-Huang Cheng. A denoising fpn with transformer r-cnn for tiny object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 2024. 6, 7

[26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37, 2016. 2

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 3

[28] Jiajun Lu, Hussein Sibai, and Evan Fabry. Adversarial examples that fool detectors. *arXiv preprint arXiv:1712.02494*, 2017. 3

[29] T. Na, J. H. Ko, and S. Mukhopadhyay. Cascade adversarial machine learning regularized with a unified embedding. In *ICLR*, 2017. 3

[30] Junhyug Noh, Wonho Bae, Wonhee Lee, Jinhwan Seo, and Gunhee Kim. Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection. In *ICCV*, pages 9725–9734, 2019. 1

[31] Henri J Nussbaumer and Henri J Nussbaumer. *The fast Fourier transform*. Springer, 1982. 2, 8

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS Workshops*, pages 1–12, 2019. 6

[33] Vitali Petsiuk, Rajiv Jain, Varun Manjunatha, Vlad I Morariu, Ashutosh Mehra, Vicente Ordonez, and Kate Saenko. Black-box explanation of object detectors via saliency maps. In *CVPR*, pages 11443–11452, 2021. 2

[34] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In *CVPR*, pages 10213–10224, 2021. 6

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE TPAMI*, 39(06):1137–1149, 2017. 6, 7, 8

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *ICCV*, 115:211–252, 2015. 6

[37] Bharat Singh, Mahyar Najibi, Abhishek Sharma, and Larry S Davis. Scale normalized image pyramids with autofocus for object detection. *IEEE TPAMI*, 44(7):3749–3766, 2021. 2

[38] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th USENIX workshop on offensive technologies (WOOT 18)*, 2018. 3

[39] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, pages 9627–9636, 2019. 1, 5, 6, 7, 8, 2, 3

[40] F Tramèr, D Boneh, A Kurakin, I Goodfellow, N Papernot, and P McDaniel. Ensemble adversarial training: Attacks and defenses. In *ICLR*, 2018. 3

[41] S Velliangiri, SJPCS Alagumuthukrishnan, et al. A review of dimensionality reduction techniques for efficient computation. *Procedia Computer Science*, 165:104–111, 2019. 4

[42] Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *CVPR*, pages 5495–5504, 2018. 4

[43] Jinwang Wang, Chang Xu, Wen Yang, and Lei Yu. A normalized gaussian wasserstein distance for tiny object detection. *arXiv preprint arXiv:2110.13389*, 2021. 6, 7

[44] Jinwang Wang, Wen Yang, Haowen Guo, Ruixiang Zhang, and Gui-Song Xia. Tiny object detection in aerial images. In *ICPR*, pages 3791–3798, 2021. 1, 5, 6

[45] Kunyu Wang, Xueyang Fu, Yukun Huang, Chengzhi Cao, Gege Shi, and Zheng-Jun Zha. Generalized uav object detection via frequency domain disentanglement. In *CVPR*, pages 1064–1073, 2023. 2

[46] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987. 4, 2

[47] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *CVPR*, pages 3974–3983, 2018. 5

[48] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *ICCV*, pages 1369–1378, 2017. 3

[49] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *CVPR*, pages 501–509, 2019. 5

[50] Chang Xu, Jinwang Wang, Wen Yang, and Lei Yu. Dot distance for tiny object detection in aerial images. In *ICCV*, pages 1192–1201, 2021. 6, 7

[51] Chang Xu, Jinwang Wang, Wen Yang, Huai Yu, Lei Yu, and Gui-Song Xia. Detecting tiny objects in aerial images: A normalized wasserstein distance and a new benchmark. *ISPRS J. Photo. and Rem. Sen.*, 190:79–93, 2022. 6, 3

[52] Chang Xu, Jinwang Wang, Wen Yang, Huai Yu, Lei Yu, and Gui-Song Xia. Rfla: Gaussian receptive field based label assignment for tiny object detection. In *ECCV*, pages 526–543, 2022. 2, 6, 7, 3

[53] Chang Xu, Jian Ding, Jinwang Wang, Wen Yang, Huai Yu, Lei Yu, and Gui-Song Xia. Dynamic coarse-to-fine learning for oriented tiny object detection. In *CVPR*, pages 7318–7328, 2023. 7

[54] Sheng Xu, Mingze Wang, Yanjing Li, Mingbao Lin, Baochang Zhang, David Doermann, and Xiao Sun. Learning 1-bit tiny object detector with discriminative feature refinement. In *ICML*, 2024. 1, 2

[55] Chenhongyi Yang, Zehao Huang, and Naiyan Wang. Querydet: Cascaded sparse query for accelerating high-resolution small object detection. In *CVPR*, pages 13668–13677, 2022. 6

[56] Xuehui Yu, Yuqi Gong, Nan Jiang, Qixiang Ye, and Zhenjun Han. Scale match for tiny person detection. In *WACV*, pages 1257–1265, 2020. 7, 2

[57] Xuehui Yu, Pengfei Chen, Kuiran Wang, Xumeng Han, Guorong Li, Zhenjun Han, Qixiang Ye, and Jianbin Jiao. Cpr++: Object localization via single coarse point supervision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7):4908–4925, 2024. 7, 3

[58] Xiang Yuan, Gong Cheng, Kebing Yan, Qinghua Zeng, and Junwei Han. Small object detection via coarse-to-fine proposal generation and imitation learning. In *ICCV*, pages 6317–6327, 2023. 2

[59] Haichao Zhang and Jianyu Wang. Towards adversarially robust object detection. In *ICCV*, pages 421–430, 2019. 3

[60] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr

with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 1, 6

[61] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, pages 9759–9768, 2020. 6, 7, 3

[62] Qijie Zhao, Tao Sheng, Yongtao Wang, Zhi Tang, Ying Chen, Ling Cai, and Haibin Ling. M2det: A single-shot object detector based on multi-level feature pyramid network. In *AAAI*, pages 9259–9266, 2019. 2

[63] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *ECCV*, pages 452–467, 2018. 5

[64] Benjin Zhu, Jianfeng Wang, Zhengkai Jiang, Fuhang Zong, Songtao Liu, Zeming Li, and Jian Sun. Autoassign: Differentiable label assignment for dense object detection. *arXiv preprint arXiv:2007.03496*, 2020. 7, 2, 3

[65] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Haibin Ling, Qinghua Hu, Qinqin Nie, Hao Cheng, Chenfeng Liu, Xiaoyu Liu, et al. Visdrone-det2018: The vision meets drone object detection in image challenge results. In *ECCV Workshops*, pages 0–0, 2018. 5, 7, 8