

# The Importance of Anti-Aliasing in Tiny Object Detection

Jinlai Ning

Michael Spratling

*King's College London, Department of Informatics, London. UK.*

✉JINLAI.NING@KCL.AC.UK

MICHAEL.SPRATLING@KCL.AC.UK

**Editors:** Berrin Yanıkoğlu and Wray Buntine

## Abstract

Tiny object detection has gained considerable attention in the research community owing to the frequent occurrence of tiny objects in numerous critical real-world scenarios. However, convolutional neural networks (CNNs) used as the backbone for object detection architectures typically neglect Nyquist's sampling theorem during down-sampling operations, resulting in aliasing and degraded performance. This is likely to be a particular issue for tiny objects that occupy very few pixels and therefore have high spatial frequency features. This paper applied an existing approach WaveCNet for anti-aliasing to tiny object detection. WaveCNet addresses aliasing by replacing standard down-sampling processes in CNNs with Wavelet Pooling (WaveletPool) layers, effectively suppressing aliasing. We modify the original WaveCNet to apply WaveletPool in a consistent way in both pathways of the residual blocks in ResNets. Additionally, we also propose a bottom-heavy version of the backbone, which further improves the performance of tiny object detection while also reducing the required number of parameters by almost half. Experimental results on the TinyPerson, WiderFace, and DOTA datasets demonstrate the importance of anti-aliasing in tiny object detection and the effectiveness of the proposed method which achieves new state-of-the-art results on all three datasets. Codes and experiment results are released at <https://github.com/freshn/Anti-aliasing-Tiny-Object-Detection.git>.

**Keywords:** Tiny Object Detection; Anti-aliasing; Wavelets; Convolutional Neural Networks

## 1. Introduction

Tiny object detection is a specialized area within the object detection field that focuses on identifying and localising small objects in images. While these objects are prevalent in real-world scenarios such as maritime rescue, driving assistance and security monitoring, they are difficult for standard object detection methods (Ren et al., 2015; Cai and Vasconcelos, 2018; Liu et al., 2015; Lin et al., 2017b; Tian et al., 2019; Zhu et al., 2020a) because of the limited spatial extent and low resolution of the target objects. The key distinction between data used in tiny object detection, such as the TinyPerson (Yu et al., 2020), WiderFace (Yang et al., 2016) and DOTA (Xia et al., 2018) datasets, and general object detection datasets lies in the proportion of small and tiny objects: ones that occupy only a few pixels in an image (typically less than 20x20 pixels in TinyPerson).

Due to their small size, the distinguishing features of tiny objects tend to have high spatial frequency. However, such high spatial frequencies will tend to be filtered out by the down-sampling performed within the Convolutional Neural Networks (CNNs) that are

typically used as the feature-extracting backbones of object detectors. Furthermore, the methods used for downsampling in these CNNs ignore the issue of aliasing.

Aliasing refers to the phenomenon where high-frequency information in an image is incorrectly represented or distorted during down-sampling (Gonzalez and Woods, 2021). According to Nyquist’s sampling theorem, the sampling rate should be at least twice the highest frequency present in the signal (Nyquist, 1928). If this is not the case, unless effective anti-aliasing techniques are employed, a down-sampled signal can exhibit significant distortions and may appear completely different from the original input signal. The aliasing problem exists for most detectors because there are many down-sampling processes involved in CNNs. Down-sampling is a commonly used operation to reduce the number of parameters and discard irrelevant details. Max-pooling, average-pooling, and strided convolution are the most widely applied down-sampling methods in standard CNNs (He et al., 2016; Howard et al., 2017; Springenberg et al., 2015). These down-sampling operations usually ignore Nyquist’s sampling theorem, which results in aliasing. The broken object structures and accumulated random noise do harm to the CNN’s performance (Grabinski et al., 2022; Li et al., 2020).

Previous work has introduced anti-aliased down-sampling operations into CNNs (Zhang, 2019; Zou et al., 2020; Williams and Li, 2018) and has shown the effectiveness of anti-aliasing in tasks such as object classification. However, despite the fact that aliasing is likely to be a particular issue when objects are small, these anti-aliasing techniques have not previously been considered in the field of tiny object detection. This inspires us to design anti-aliased backbones for tiny object detection. Our proposed anti-aliased backbone is heavily based on WaveCNet (Li et al., 2020). WaveCNet employs WaveletPool to replace the down-sampling processes in a standard ResNet (He et al., 2016), and has been shown to achieve excellent performance in image classification and standard object detection tasks. Here we show that this method can also be used to great effect in tiny object detection. Our contributions are as follows:

- We apply WaveletPool to reduce distortions in the representations of tiny objects and preserve more discriminate information about such objects.
- We modify WaveCNet so that the order in which Wavelet Pooling is performed is consistent among all connections in the backbone to improve the performance of tiny object detection.
- We combine anti-aliasing with a bottom-heavy architecture (Ning et al., 2022) to improve tiny object detection performance further.
- We achieve state-of-the-art performance on three tiny object detection benchmarks.

## 2. Related Work

### 2.1. Tiny Object Detection

Previous approaches to improving the performance of tiny object detectors can be roughly categorized into seven types (Tong and Wu, 2022). In the subsequent paragraphs, representative examples from each category will be reviewed.

Super-resolution techniques play a crucial role in enhancing the fine details and characteristics of tiny targets, enabling standard object detection techniques designed for larger objects to be applied effectively to images that contain tiny targets. For instance, a generative neural network approach can be employed to enhance the resolution of the entire image (Bai et al., 2018). Alternatively, the super-resolution technique can be selectively applied to small regions of interest within the image (Yang et al., 2019b).

Contextual information, obtained from regions of the image surrounding an object, can greatly assist in the detection of that object. Many techniques incorporate contextual cues into their networks. Hu and Ramanan (2017) defined templates that utilized very large receptive fields to capture extensive contextual information to aid in the detection of small faces. Another example is the scale selection pyramid network (SSPNet; Hong et al., 2022). SSPNet encompasses a context attention module (CAM) to generate hierarchical attention heatmaps, allowing the network to prioritize areas that are more likely to contain small objects of interest.

Data augmentation can be used to enhance performance by expanding the training dataset through image transformations. PyramidBox (Tang et al., 2018) exploited the data-anchor-sampling (DAS) technique that reshaped a randomly selected object within the image to match a smaller anchor size. Yu et al. (2020) proposed an efficient augmentation technique named "scale match" to align the object size distribution differences between general datasets and tiny object detection datasets. The scale match technique involves sampling a size-bin from the histogram of object sizes in the external dataset. To ensure that the range of sampled size-bins does not deviate significantly, the authors also introduced a monotone scale match (MSM) strategy for the sampling in a monotonically increasing or decreasing order. Jiang et al. (2021) introduced an enhanced version of the scale match technique called SM+, which extended it from the image level to the instance level.

Multi-scale representation learning is a crucial and effective strategy employed in the detection of small or tiny objects. This approach is commonly applied in general object detection to handle objects of varying sizes, such as through the feature pyramid network (FPN) proposed by Lin et al. (2017a). Gong et al. (2021) argued that while FPN brought positive impacts, it also introduced negative effects caused by the top-down connections to tiny object detection. To address this, they introduced a statistic-based fusion factor that dynamically adjusted the weight of different layers during feature fusion. Tang et al. (2018) introduced a low-level feature pyramid network (LFPN) that initiated from a middle layer instead of starting the top-down structure from a high-level layer because of the observation that not all high-level semantic features are equally beneficial for smaller targets. Liu et al. (2021b) proposed a feature rescaling and fusion (SFRF) network that incorporated a nonparametric adaptive dense perceiving algorithm (NADPA), which had the capability to automatically select and generate a resized feature map that focused on the high-density distribution of tiny objects.

The anchor mechanism predetermines locations which can efficiently scan the image and match potential object regions where objects are anticipated to be detected. Zhang et al. (2017) maintained consistent anchor density across different scales and proposed a scale compensation anchor matching strategy to ensure that all scales had an adequate number of matched anchors. Deng et al. (2020) ensured proper matching for objects of all sizes by incorporating anchors of varying sizes across multiple layers. Zhang et al. (2018)

developed a specialized single-level face detection framework that utilized different dilation rates for anchors with different sizes. [Xu et al. \(2022\)](#) introduced a novel label assignment method to address the challenges related to insufficient positive samples and the disparity between uniformly distributed prior anchors and the Gaussian-distributed receptive field. To handle the severe misalignment between anchor boxes and axis-aligned convolutional features, S<sup>2</sup>ANet ([Han et al., 2021](#)) included a feature alignment module. In addition to anchor-based methods, some point-based anchor-free methods are also designed for tiny object detection. For example, Oriented RepPoints ([Wentong et al., 2022](#)) represent objects in a set of sample points that effectively bound the spatial extent of the objects and identify semantically significant local areas.

Training detectors for large objects is relatively straightforward, a significant challenge lies in effectively training detectors for small objects. [Luo et al. \(2019\)](#) enhanced the robustness and overall performance by using multi-scale training images. Similarly, [Hu and Ramanan \(2017\)](#) proposed an efficient approach where separate face detectors are trained with features extracted from multiple layers of a single feature hierarchy.

Optimizing the loss function is a valuable strategy that helps enhance the overall performance of detecting tiny objects. [Liu et al. \(2021a\)](#) introduced a novel feedback-driven loss function leveraging the information from the loss distribution as a feedback signal, enabling the model to be trained in a more balanced manner. SSPNet ([Hong et al., 2022](#)) included an attention-based loss function to supervise the different layers for extracting information from different ranges of objects.

Our proposed approach considers a new aspect of tiny object detection: anti-aliasing. It is, therefore, independent of the previous work mentioned. Hence, there is a potential that all these different existing techniques could be enhanced by integrating them with our proposed method.

## 2.2. Anti-Aliasing Filters

The low-pass filter is a textbook-style solution to aliasing. Following this idea, several CNN anti-aliasing methods have been proposed ([Zhang, 2019](#); [Zou et al., 2020](#); [Li et al., 2020](#)). [Zhang \(2019\)](#) proposed BlurPool which incorporated a Gaussian blur layer before each down-sampling module. [Vasconcelos et al. \(2021\)](#) indicated the order of low-pass filters and convolutions had an impact on the effectiveness of anti-aliasing and updated the residual networks architecture accordingly ([Vasconcelos et al., 2020](#)). [Zou et al. \(2020\)](#) introduced an adaptive content-aware low-pass filtering layer (AdaBlurPool) to generate distinct filter weights for each spatial location and channel group. Although these methods have achieved impressive results on other tasks such as image classification, domain generalization, instance segmentation, and semantic segmentation, they do not seem promising for tiny object detection, and the experimental results in Tab. 1 confirm this. Gaussian blur ([Zhang, 2019](#); [Vasconcelos et al., 2021](#)) is likely to result in a loss of information about tiny features and make tiny objects less recognizable. Whereas AdaBlurPool ([Zou et al., 2020](#)) is not theoretically justified, and the adaptive low-pass filter is generated by learnable group convolutional layers that are not guaranteed to learn an appropriate low-pass filter (even after many iterations and even if initialized to Gaussian filters).

In contrast to the above methods, Li et al. (2020) ensures that low-frequency features are not corrupted by high-frequency artifacts by replacing standard down-sampling layers with a discrete wavelet transform (DWT) and an inverse discrete wavelet transform (IDWT). They implemented these transforms using various orthogonal and biorthogonal discrete wavelets including Haar, Daubechies, and Cohen. Such WaveletPool operations were used to replace standard down-sampling operations in ResNets. The resulting WaveCNet (Li et al., 2020) demonstrated the robustness of WaveletPool layers for general object detection in adversarially-attacked scenes. Here we apply WaveCNet, with slight modification, to tiny object detection.

### 3. Methods

The impact of aliasing is worse in tiny object detection compared to general object detection because tiny objects have more high-frequency features than general objects. The values of pixels of a tiny object could change rapidly in a small range. Tiny objects are therefore more sensitive to being affected by aliasing. However, current backbones used in tiny object detection use CNNs that employ down-sampling operations that ignore Nyquist’s sampling theorem, resulting in severe aliasing. To investigate the harm this does to the performance of tiny object detectors we applied WaveletPool (as described in Sec. 3.1) in a widely used standard backbone ResNet50 (as detailed in Sec. 3.2) and a bottom-heavy architecture previously shown to be effective for tiny object detection (as described in Sec. 3.3).

#### 3.1. Wavelet Pooling

Wavelet Pooling (Li et al., 2020) depends on the wavelet transform theorem (Mallat, 1989), and is implemented using the DWT and the IDWT.

**1-dimensional (1D) DWT/IDWT.** In 1D, DWT decomposes a given vector  $\mathbf{x} = \{x_n\}_{n \in \mathbb{Z}}$  into its low-frequency component  $\mathbf{x}^{\text{low}} = \{x_k^{\text{low}}\}_{k \in \mathbb{Z}}$  and high-frequency component  $\mathbf{x}^{\text{high}} = \{x_k^{\text{high}}\}_{k \in \mathbb{Z}}$ , and IDWT reconstructs the vector  $\mathbf{x}$  using  $\mathbf{x}^{\text{low}}$  and  $\mathbf{x}^{\text{high}}$ , as follows:

$$\mathbf{x}^{\text{low}} = \mathcal{L}\mathbf{x}, \quad \mathbf{x}^{\text{high}} = \mathcal{H}\mathbf{x} \quad (1)$$

$$\mathbf{x}^* = \mathcal{L}^T \mathbf{x}^{\text{low}} + \mathcal{H}^T \mathbf{x}^{\text{high}} \quad (2)$$

where  $\mathbf{x}^*$  denotes the reconstructed vector,  $\mathcal{L} = \{\mathbf{l}_{n-2}, \dots, \mathbf{l}_{n-2k}\}^T$  and  $\mathcal{H} = \{\mathbf{h}_{n-2}, \dots, \mathbf{h}_{n-2k}\}^T$ .  $\mathbf{l}_j = \{l_i\}_{i=j-2k, \dots, j+n-2k}$  and  $\mathbf{h}_j = \{h_i\}_{i=j-2k, \dots, j+n-2k}$  are the low-pass and high-pass filters of an orthogonal wavelet,  $n$  denotes length of the input vector,  $k$  denotes the length of the frequency component.

**2-dimensional (2D) DWT/IDWT.** The 2D DWT is achieved by performing 1D DWT on both rows and columns, *ie.*,

$$\mathbf{X}_{\text{ll}} = \mathcal{L}\mathbf{X}\mathcal{L}^T, \quad \mathbf{X}_{\text{lh}} = \mathcal{H}\mathbf{X}\mathcal{L}^T, \quad \mathbf{X}_{\text{hl}} = \mathcal{L}\mathbf{X}\mathcal{H}^T, \quad \mathbf{X}_{\text{hh}} = \mathcal{H}\mathbf{X}\mathcal{H}^T \quad (3)$$

The corresponding 2D IDWT is:

$$\mathbf{X}^* = \mathcal{L}^T \mathbf{X}_{\text{ll}} \mathcal{L} + \mathcal{H}^T \mathbf{X}_{\text{lh}} \mathcal{L} + \mathcal{L}^T \mathbf{X}_{\text{hl}} \mathcal{H} + \mathcal{H}^T \mathbf{X}_{\text{hh}} \mathcal{H} \quad (4)$$

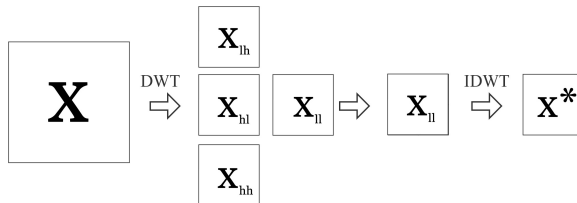


Figure 1: Wavelet pooling. The discrete wavelet transformation is applied to the input  $\mathbf{X}$ . The inverse discrete wavelet transformation is applied only to the low-frequency components in order to construct the down-sampled output  $\mathbf{X}^*$ . The notations used are the same as those used in the ‘2-dimensional (2D) DWT/IDWT’ part of Section 3.1.

Integrating such operations into modern deep learning frameworks is possible as the gradients can be easily calculated (Li et al., 2020).

**Down-sampling based on 2D DWT/IDWT** Wavelet Pooling, as illustrated in Fig 1, consists of applying one 2D DWT and one 2D IDWT in sequence. The application of DWT decomposes the image (or feature maps of a CNN layer) into the high-frequency detail subbands  $\mathbf{X}_{lh}$ ,  $\mathbf{X}_{hl}$ ,  $\mathbf{X}_{hh}$ , and the low-frequency approximation subband  $\mathbf{X}_{ll}$  in the wavelet domain. The IDWT is applied only to low-frequency subband  $\mathbf{X}_{ll}$  without using the detail subbands  $\mathbf{X}_{lh}$ ,  $\mathbf{X}_{hl}$ ,  $\mathbf{X}_{hh}$ , which cuts the resolution in half. The reconstructed image or feature map will still capture the low-frequency information from the input,  $\mathbf{X}$ , while filtering out the high-frequency information without aliasing.

**Pooling with different wavelets** A number of different wavelets can be used. One example of an orthogonal wavelet is the Daubechies wavelet. It uses  $l^{\text{low}}$  and  $h^{\text{high}}$  filters to calculate DWT and IDWT. If the length of the filter is  $2k$ , the Daubechies wavelet has an approximation order parameter  $k$ , noted as Daubechies( $k$ ). Daubechies(1) is also known as Haar wavelet.

The Cohen wavelet is an example of a biorthogonal wavelet, which means that it uses  $l^{\text{low}}$  and  $h^{\text{high}}$  to implement the DWT but uses their dual vectors  $\tilde{l}^{\text{low}}$  and  $\tilde{h}^{\text{high}}$  for the IDWT. A typical Cohen wavelet is written as Cohen( $k, \tilde{k}$ ) where  $k$  and  $\tilde{k}$  indicate the length of original and dual filters. Cohen(1,1) is identical to the Daubechies(1), which is also the Haar wavelet. In our experiments, we used the Haar wavelet, denoted ‘haar’, and the Cohen( $k, \tilde{k}$ ) wavelet, denoted ‘ch $k, \tilde{k}$ ’.

### 3.2. WaveCNet with Consistent Order of Wavelet Pooling

WaveCNet (Li et al., 2020) replaces all down-samplings in a ResNet with WaveletPool, as described in the preceding section. Max-pooling is directly replaced by WaveletPool. Strided convolution is replaced by a convolution with the stride of 1 followed by a wavelet-pooling, *ie.*,

$$\begin{aligned} \text{MaxPool}_{s=2} &\rightarrow \text{WaveletPool}_{X_{ll}} \\ \text{Conv}_{s=2} &\rightarrow \text{Conv}_{s=1} \circ \text{WaveletPool}_{X_{ll}} \end{aligned} \quad (5)$$

where ‘Pool’ and ‘Conv’ denote pooling and convolution respectively and  $\circ$  denotes ‘followed by’.

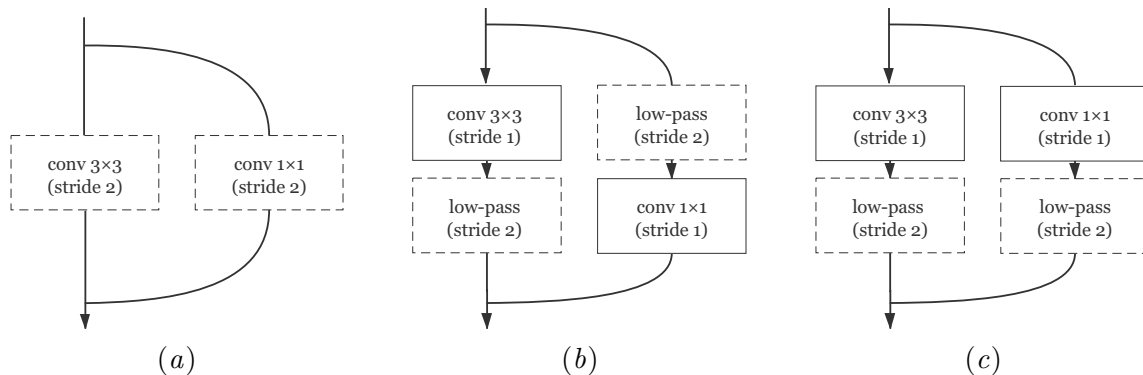


Figure 2: Different implementations of a residual block. (a) original residual block, (b) previous anti-aliasing, (c) our anti-aliasing block.

As discussed in (Vasconcelos et al., 2021), whether to apply a low-pass filter before or after a convolutional layer matters for anti-aliasing. Their experiments show better performance is achieved if the low-pass filter comes after the convolution. Despite this, previous anti-aliasing networks (Vasconcelos et al., 2021; Zou et al., 2020; Li et al., 2020), have not consistently done so, and the order in which convolution and low-pass filtering is performed differs across different branches in ResNet blocks, as shown in Fig. 2. As Fig. 2(a) shows, a standard down-sampling residual block could be simplified as a main path and a skip connection. Fig. 2(b) shows the regular order used in most other anti-aliasing CNNs. The low-pass filter follows convolution in the main path but, these operations occur in the reverse order in the skip connection. Experimental results (shown in Tab. 1) confirm that using a consistent order in both pathways is beneficial for tiny object detection, and hence, we use the configuration shown in Fig. 2(c).

### 3.3. Bottom-Heavy Backbone

Our previous work showed that using a bottom-heavy version of ResNet50 produced better performance on tiny object detection (Ning et al., 2022). This bottom-heavy backbone, BHResNet50, basically shifts calculations from the top layers to the early layers without introducing additional computational costs. This is done by delaying the down-sampling layers, resulting in a decrease in the number of convolutional layers applied to low-resolution features in deeper layers, and an increase in the number of convolutional layers used in earlier, higher-resolution, layers. Here, we implement a bottom-heavy anti-aliased architecture by replacing all down-sampling processes with WaveletPool in a BHResNet50.

## 4. Experiments

### 4.1. Pre-Training Datasets

The backbone of an object detector is typically pre-trained on an image classification task as this leads to more rapid and stable training on the detection task (Girshick et al., 2014; Kornblith et al., 2019). Here, two pre-training datasets were employed.



**ImageNet.** ImageNet, also known as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset (Deng et al., 2009), is a widely used benchmark dataset for image classification tasks in computer vision. It consists of over 1.2 million images from a thousand different categories. The dataset includes a diverse set of categories such as animals, vehicles, household items, plants, and various everyday objects. It is widely recognized as a standard dataset for pre-training the backbone of object detection models. The results shown in Tables 3, 4 and 5 were produced using backbones pre-trained for 100 epochs on ImageNet. The learning rate started from 0.1 and reduced by 1/10 at 30, 60 and 90 epochs.

**CIFAR100.** CIFAR100 (Krizhevsky and Hinton, 2009) is a popular image dataset used for classification tasks in computer vision. It contains 60,000 32x32 colour images. These images are divided into 100 fine-grained classes, with each class having 600 images. The 100 classes in CIFAR100 cover a wide range of object categories, including animals, vehicles, household items, and various natural and man-made objects. CIFAR100 is a good choice for prototyping network architectures because of the reasonable training time. The results shown in Tab. 1, 2 were obtained using backbones pre-trained for 200 epochs on CIFAR100. The learning rate started from 0.1 and dropped to 1/10 at 100 and 150 epochs.

## 4.2. Tiny Object Detection Datasets

Performance on tiny object detection tasks was evaluated using three standard datasets.

**TinyPerson.** This dataset (Yu et al., 2020) is collected from real-world videos, consisting of 1610 large-scale seaside images with more than 70000 bounding boxes. The 794 images are divided into subsets for training and validation (with 1/10 or the images in the latter). On average, the target objects within the images have an absolute size of approximately 18 pixels. The size range (pixels) of the objects is divided into three sub-intervals for TinyPerson: tiny [2, 20], small [20, 32], and all [2, inf]. The tiny set is partitioned into three (overlapping) sub-intervals: tiny1 [2, 8], tiny2 [8, 12], and tiny3 [12, 20]. Results are reported using mean Average Precision ( $mAP$ ) metrics for all intervals.

**WiderFace.** This dataset (Yang et al., 2016) is a widely-used benchmark for face detection. It consists of 32,203 images that contain various human faces with a wide range of scales, poses, and occlusions. The average size of the objects in the WiderFace dataset is approximately 32 pixels. Based on the level of difficulty in detecting the objects, the dataset is divided into three evaluation sets: easy, medium, and hard. Results are reported as  $mAP$  for these three subsets.

**DOTA.** This dataset (Xia et al., 2018) is for object detection in aerial images. It contains 2,806 aerial images obtained from various sensors and platforms, with sizes ranging from approximately 800x800 pixels to 4,000x4,000 pixels. The fully annotated DOTA benchmark contains 188,282 instances in 15 common object categories. Half of the original images are randomly assigned to the training set, while 1/6 is allocated to the validation set, and the remaining 1/3 is used as the testing set. We report categorical  $AP$  and  $mAP$  for the oriented bounding boxes task.



Anti-Aliasing Method	Order	$mAP_{tiny}$	$mAP_{all}$	$mAP_{tiny1}$	$mAP_{tiny2}$	$mAP_{tiny3}$	$mAP_{small}$
BlurPool	(b)	41.17	43.39	24.06	44.98	54.02	58.55
	(c)	44.65	46.74	27.71	49.45	56.14	60.30
AdaBlurPool	(b)	42.34	45.13	25.21	47.04	54.09	58.30
	(c)	46.28	47.12	30.94	<b>52.23</b>	56.07	59.63
WaveletPool ch3.3	(b)	46.08	47.62	29.24	51.22	57.05	<b>63.19</b>
	(c)	<b>46.34</b>	<b>47.70</b>	<b>30.98</b>	51.21	<b>57.24</b>	60.14

Table 1: Performance of Faster R-CNN on TinyPerson when using a ResNet50 backbone employing different methods of anti-aliasing and different orders of application. (b), (c) indicate the order illustrated in Fig 2. Backbones are pre-trained on CIFAR100.

### 4.3. Selection of Anti-Aliasing Method and Order of Application

As described in Sec. 2.2, a number of different anti-aliasing methods have been proposed. Tab. 1 compares the performance of these different methods on tiny object detection. The results demonstrate that Wavelet Pooling has superior performance, compared to the alternatives, in this application.

As described in Sec. 3.2, we reverse the order in which convolution and anti-aliased down-sampling are performed in the skip connections in the proposed backbones. The experimental results, shown in Tab. 1, show that this change produces a small increase in performance on tiny object detection for Wavelet Pooling and other, alternative, anti-aliased down-sampling methods. We therefore use the order shown in Fig 2(c) in all subsequent experiments.

### 4.4. Selection of Wavelet

To select an appropriate wavelet to use in the anti-aliasing module, the performance produced with different wavelets was tested on TinyPerson. These results were compared to those produced using a standard CNN backbone. These backbones were pre-trained on CIFAR100 to reduce pre-training time. As shown in Tab. 2, Cohen(3,3) wavelet produced the best results for most metrics, both for the ResNet50 and BHRNet50 backbones. Cohen(3,3) was, therefore, selected as the wavelet used in subsequent experiments.

### 4.5. Results on TinyPerson

By replacing the standard ResNet50 backbone in the current state-of-the-art method SSPNet with our proposed anti-aliasing backbones, we get better  $mAP$  among all intervals of TinyPerson. As shown in Tab. 3, using WaveletPool increases the  $mAP_{tiny}$  by 1.31% compared to plain SSPNet while integrating WaveletPool with the bottom-heavy backbone increases performance by an additional 0.1%. Hence, overall the proposed method improves on the previous state-of-the-art result by 1.41% in terms of  $mAP_{tiny}$ . All networks were trained for 12 epochs with stochastic gradient descent (SGD) as the optimizer.

Backbone	Anti-aliasing	$mAP_{tiny}$	$mAP_{all}$	$mAP_{tiny1}$	$mAP_{tiny2}$	$mAP_{tiny3}$	$mAP_{small}$	Params	FLOPs
ResNet	None	44.07	45.79	28.38	49.39	55.28	60.66	41.07	75.58
	WaveletPool haar	45.37	46.62	29.78	49.26	56.51	59.70	41.30	90.78
	WaveletPool ch3.3	<b>46.34</b>	47.70	<b>30.98</b>	51.21	57.24	60.14	41.30	90.78
	WaveletPool ch5.5	43.63	45.43	28.96	47.24	54.84	59.87	41.30	90.78
BHResNet	None	44.51	45.99	30.59	50.01	54.36	60.28	28.75	75.79
	WaveletPool haar	45.33	46.95	30.15	49.48	56.72	<b>61.54</b>	28.75	91.90
	WaveletPool ch3.3	46.04	<b>47.70</b>	29.28	<b>51.21</b>	<b>57.33</b>	61.01	28.75	91.90
	WaveletPool ch5.5	44.61	46.18	29.34	48.97	55.18	60.15	28.75	91.90

Table 2: Performance of Faster R-CNN on TinyPerson when using different anti-aliasing filters. Backbones are pre-trained on CIFAR100. ‘Params’ indicates the number of parameters (in millions) required by the detector. ‘FLOPs’ indicates the number of floating-point operations (in billions) performed on the standard TinyPerson input size of  $640 \times 512$  pixels. Wavelets compared in this table are described in the ‘Pooling with different wavelets’ part of Section 3.1.

Methods	$mAP_{tiny}$	$mAP_{all}$	$mAP_{tiny1}$	$mAP_{tiny2}$	$mAP_{tiny3}$	$mAP_{small}$
Faster R-CNN (Ren et al., 2015)	47.81	49.99	31.78	53.54	58.34	64.56
Adap FCOS (Tian et al., 2019)	48.45	51.03	29.75	52.71	60.36	65.28
S- $\alpha$ + Faster R-CNN (Gong et al., 2021)	48.39	*	31.68	52.20	60.01	65.15
RFLA + Faster R-CNN (Xu et al., 2022)	48.86	51.92	30.35	54.15	61.28	66.69
SM + Faster R-CNN (Yu et al., 2020)	49.09	51.57	32.93	55.42	59.70	64.24
SM+ + Faster R-CNN (Jiang et al., 2021)	51.46	*	33.74	55.32	62.95	67.37
SFRF + Faster R-CNN (Liu et al., 2021b)	57.24	59.03	51.49	64.51	67.78	65.33
SSPNet w ResNet50 (Hong et al., 2022)	57.93	62.23	45.33	60.26	67.09	71.56
SSPNet w BHResNet50 (Ning et al., 2022)	58.97	62.02	47.22	61.61	67.45	72.37
SSPNet w ResNet50 + WaveletPool	59.24	63.16	<b>47.76</b>	60.23	68.04	73.42
SSPNet w BHResNet50 + WaveletPool	<b>59.34</b>	<b>64.02</b>	46.55	<b>61.73</b>	<b>68.95</b>	<b>73.87</b>

Table 3: Comparison of the performance of the proposed methods with previous results on TinyPerson. All backbones were pre-trained on ImageNet. \* indicates results that are unavailable because of unreleased code.

The learning rate was initialized to 0.002/0.003 for ResNet50+WaveletPool and BHResNet50+WaveletPool respectively and decreased by a factor of 0.1 after 8 and 11 epochs.

#### 4.6. Results on WiderFace

Table 4 shows that integrating WaveletPool into the ResNet50 backbone, and its bottom-heavy variant, used by TinaFace (Zhu et al., 2020b), the current state-of-the-art method for WiderFace, achieves new state-of-the-art results on all three sub-tasks. Both backbones are trained using SGD optimizer with batch size 16. Training was performed for a total of 630

Method	$mAP_{easy}$	$mAP_{medium}$	$mAP_{hard}$
HR (Hu and Ramanan, 2017)	92.5	91.0	80.6
S3FD (Zhang et al., 2017)	93.7	92.4	85.2
FaceGAN (Bai et al., 2018)	94.4	93.3	87.3
SFA (Luo et al., 2019)	94.9	93.6	86.6
LSFHI (Zhang et al., 2018)	95.7	94.9	89.7
Pyramid-Box (Tang et al., 2018)	96.1	95.0	88.9
RetinaFace (Deng et al., 2020)	96.5	95.6	90.4
TinaFace w ResNet50 (Zhu et al., 2020b)	96.3	95.7	93.1
TinaFace w ResNet50 + WaveletPool	96.4	95.8	93.1
TinaFace w BHResNet50 + WaveletPool	<b>96.6</b>	<b>96.0</b>	<b>93.4</b>

Table 4: Performance of previous methods and our methods on WiderFace.

Method	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SF	RO	HA	SP	HE	$mAP$
SSD (Liu et al., 2015)	39.8	9.1	0.6	13.2	0.3	0.4	1.1	16.2	27.6	9.2	27.2	9.1	3.0	1.0	1.0	10.6
FDLoss (Liu et al., 2021a)	79.0	38.2	28.7	36.9	44.2	40.9	57.8	65.4	54.6	36.3	34.1	39.6	42.9	45.1	16.3	43.2
S <sup>2</sup> ANet (Han et al., 2021)	<b>89.2</b>	<b>78.0</b>	47.7	68.1	77.4	73.1	79.1	90.9	79.5	85.0	56.7	58.6	59.9	65.6	<b>51.2</b>	70.1
RepPoints (Yang et al., 2019a)	87.8	77.7	49.5	66.5	78.5	73.1	86.6	90.7	<b>83.8</b>	84.3	53.1	<b>65.6</b>	63.7	<b>68.7</b>	45.9	71.7
RepPoints+WaveletPool	88.6	73.8	<b>52.4</b>	70.9	79.1	73.3	86.2	<b>90.9</b>	81.2	<b>85.7</b>	58.8	61.4	<b>64.5</b>	66.2	45.4	<b>71.9</b>
RepPoints w BH+WaveletPool	88.3	74.6	49.8	<b>72.4</b>	<b>79.4</b>	<b>74.0</b>	<b>86.6</b>	90.8	78.6	84.7	<b>59.9</b>	60.1	62.6	67.0	44.7	71.6

Table 5: Categorical and mean average precision of previous and our methods on DOTA. BH indicates BHResNet50. The abbreviations indicate the following categories: PL-Plane, BD-Baseball diamond, BR-Bridge, GTF-Ground field track, SV-Small vehicle, LV-Large vehicle, SH-Ship, TC-Tennis court, BC-Basketball court, ST-Storage tank, SBF-Soccer-ball field, RA-Roundabout, HA-Harbor, SP-Swimming pool, and HC-Helicopter.

epochs and the learning rate varied between  $3.75e-3$  and  $3.75e-5$  every 30 epochs following the cosine schedule (Loshchilov and Hutter, 2016).

#### 4.7. Results on DOTA

For the DOTA detection task, we evaluated the proposed backbones incorporating Wavelet-Pool by integrating them with the current state-of-the-art method: Oriented RepPoints (Wentong et al., 2022). The proposed method achieved the best  $mAP$ , and better  $AP$  for the majority of object categories. Both backbones were trained for 12 epochs with SGD as the optimizer. The learning rate was initialized to 0.008 and decreased by a factor of 0.1 after 8 and 11 epochs. The results were produced using the DOTA evaluation server.

#### 4.8. Knowledge-Distillation for Efficient Learning

Knowledge distillation is a method used in machine learning to transfer knowledge from an accurate model, known as the teacher model, to a simpler model, called the student

Methods	SS	KD	$mAP_{tiny}$	$mAP_{all}$	Top-1 Acc	Training Time
ResNet50			51.72	53.62	77.47	3807 mins
+ WaveletPool	✓		49.52	51.12	65.31	284 mins
	✓	✓	51.42	53.66	73.05	283 mins
BHResNet50			51.78	53.60	76.31	3807 mins
+ WaveletPool	✓		49.37	51.51	64.51	284 mins
	✓	✓	50.61	53.39	70.86	283 mins

Table 6: Performance of the proposed backbones when trained using different methods and schedules on TinyPerson. All results are evaluated with Faster R-CNN. SS indicates the short schedule. KD indicates the knowledge distillation training method.

model (Hinton et al., 2015). The goal is to distill the knowledge learned by the teacher model into the student model, enabling the student model to achieve comparable performance to the teacher model but with reduced computational requirements. Inspired by (Suzuki et al., 2022), we use the original ResNet50 as the teacher and an architecture incorporating WaveletPool (either ResNet50 or BHResNet50) as the student. The student network was trained using a short training schedule which had only ten epochs and the learning rate was reduced by 1/10 at epochs 3, 6 and 9. As can be seen in Tab. 6, using a short schedule alone clearly has a negative impact on the performance of the model. However, when combined with the knowledge-distillation approach, the achieved  $mAP$  was similar to that achieved using the original, full, training schedule. This demonstrates that it is possible to train the proposed backbones efficiently which could be helpful for exploring different architectures based on WaveletPool in future research.

## 5. Conclusion

This study explores the detrimental impact of aliasing on tiny object detection. To suppress the aliasing effect, we replace standard down-sampling methods in the commonly used ResNet50 backbone with WaveletPool. Experimental results demonstrate the superior object detection accuracy achieved by our approach. We have achieved state-of-the-art performance on DOTA using this method. Furthermore, extending our previously proposed bottom-heavy backbone architecture to also use WaveletPool, we achieve state-of-the-art performance on the TinyPerson and WiderFace datasets while utilizing fewer parameters. Additionally, we explore the potential for expediting the training process of our network using knowledge distillation techniques, providing an efficient route for future investigations to incorporate our methods into other backbones.

## Acknowledgments

The authors gratefully acknowledge the use of the King’s Computational Research, Engineering and Technology Environment (CREATE) and Joint Academic Data Science Endeavour (JADE). This research was funded by the King’s-China Scholarship Council (K-CSC).

## References

- Yancheng Bai, Yongqiang Zhang, Mingli Ding, and Bernard Ghanem. Finding tiny faces in the wild with generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018.
- J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR09*, 2009.
- Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. RetinaFace: Single-shot multi-level face localisation in the wild. In *CVPR*, 2020.
- Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, May 2014. doi: 10.1109/cvpr.2014.81. URL <http://dx.doi.org/10.1109/cvpr.2014.81>.
- Yuqi Gong, Xuehui Yu, Yao Ding, Xiaoke Peng, Jian Zhao, and Zhenjun Han. Effective fusion factor in FPN for tiny object detection. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1159–1167, 2021.
- Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing, Global Edition, 4/E*. Prentice Hall, Feb 2021.
- Julia Grabinski, Janis Keuper, and Margret Keuper. Aliasing and adversarial robust generalization of CNNs. *Machine Learning*, 111:1–27, 08 2022. doi: 10.1007/s10994-022-06222-8.
- J. Han, J. Ding, J. Li, and G. S. Xia. Align deep features for oriented object detection. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–11, 2021. doi: 10.1109/TGRS.2021.3062048.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, May 2016. doi: 10.1109/cvpr.2016.90. URL <http://dx.doi.org/10.1109/cvpr.2016.90>.
- GeoffreyE. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *arXiv: Machine Learning*, Mar 2015.
- Ming Hong, Shuiwang Li, Yuchao Yang, Feiyu Zhu, Qijun Zhao, and Li Lu. SSPNet: Scale selection pyramid network for tiny person detection from UAV images. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.

- Andrew Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, M. Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv: Computer Vision and Pattern Recognition*, Apr 2017.
- Peiyun Hu and Deva Ramanan. Finding tiny faces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- Nan Jiang, Xuehui Yu, Xiaoke Peng, Yuqi Gong, and Zhenjun Han. SM+: Refined scale match for tiny person detection. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1815–1819, 2021. doi: 10.1109/ICASSP39728.2021.9414162.
- Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, May 2019. doi: 10.1109/cvpr.2019.00277. URL <http://dx.doi.org/10.1109/cvpr.2019.00277>.
- Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.
- Qiufu Li, Linlin Shen, Sheng Guo, and Zhihui Lai. WaveCNet: Wavelet integrated CNNs to suppress aliasing effect for noise-robust image classification. *IEEE Transactions on Image Processing*, page 7074–7089, Dec 2020. doi: 10.1109/tip.2021.3101395. URL <http://dx.doi.org/10.1109/tip.2021.3101395>.
- Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017a.
- Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017b. URL <http://arxiv.org/abs/1708.02002>.
- Gen Liu, Jin Han, and Wenzhong Rong. Feedback-driven loss function for small object detection. *Image and Vision Computing*, 111:104197, 2021a.
- Jingwei Liu, Yi Gu, Shumin Han, Zhibin Zhang, Jiafeng Guo, and Xueqi Cheng. Feature rescaling and fusion for tiny object detection. *IEEE Access*, 9:62946–62955, 2021b. doi: 10.1109/ACCESS.2021.3074790.
- Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015. URL <http://arxiv.org/abs/1512.02325>.
- Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv: Learning*, Aug 2016.
- Shi Luo, Xiongfei Li, Rui Zhu, and Xiaoli Zhang. SFA: Small faces attention face detector. *IEEE Access*, 7:171609–171620, 2019. doi: 10.1109/ACCESS.2019.2955757.

- S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 674–693, Jun 1989. doi: 10.1109/34.192463. URL <http://dx.doi.org/10.1109/34.192463>.
- Jinlai Ning, Haoyan Guan, and Michael Spratling. Rethinking the backbone architecture for tiny object detection. In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Dec 2022. doi: 10.5220/0011643500003417. URL <http://dx.doi.org/10.5220/0011643500003417>.
- H. Nyquist. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, 1928. doi: 10.1109/t-aiee.1928.5055024. URL <http://dx.doi.org/10.1109/t-aiee.1928.5055024>.
- Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015. URL <http://arxiv.org/abs/1506.01497>.
- Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2015.
- Satoshi Suzuki, Shoichiro Takeda, Naoki Makishima, Atsushi Ando, Ryo Masumura, and Hayaru Shouno. Knowledge transferred fine-tuning: Convolutional neural network is born again with anti-aliasing even in data-limited situations. *IEEE Access*, 10:68384–68396, 2022. doi: 10.1109/ACCESS.2022.3186101.
- Xu Tang, Daniel K. Du, Zeqiang He, and Jingtuo Liu. PyramidBox: A context-assisted single shot face detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
- Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9626–9635, 2019.
- Kang Tong and Yiquan Wu. Deep learning-based detection from the perspective of small or tiny objects: A survey. *Image and Vision Computing*, 123:104471, 2022. ISSN 0262-8856. doi: <https://doi.org/10.1016/j.imavis.2022.104471>. URL <https://www.sciencedirect.com/science/article/pii/S0262885622001007>.
- CristinaNader Vasconcelos, Hugo Larochelle, Vincent Dumoulin, NicolasLe Roux, and Ross Goroshin. An effective anti-aliasing approach for residual networks. *arXiv: Computer Vision and Pattern Recognition*, Nov 2020.
- CristinaNader Vasconcelos, Hugo Larochelle, Vincent Dumoulin, Rob Romijnders, NicolasLe Roux, and Ross Goroshin. Impact of aliasing on generalization in deep convolutional networks. *Cornell University - arXiv*, Aug 2021.
- Li Wentong, Chen Yijie, Hu Kaixuan, and Zhu Jianke. Oriented reppoints for aerial object detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.



- Travis Williams and Robert Li. Wavelet pooling for convolutional neural networks. *International Conference on Learning Representations*, Feb 2018.
- 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- Chang Xu, Jinwang Wang, Wen Yang, Huai Yu, Lei Yu, and Gui-Song Xia. RFLA: Gaussian receptive based label assignment for tiny object detection. In *European Conference on Computer Vision (ECCV)*, 2022.
- Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019a.
- Zixuan Yang, Xiujuan Chai, Ruiping Wang, Weijun Guo, Weixuan Wang, Li Pu, and Xilin Chen. Prior knowledge guided small object detection on high-resolution images. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 86–90, 2019b. doi: 10.1109/ICIP.2019.8802612.
- Xuehui Yu, Yuqi Gong, Nan Jiang, Qixiang Ye, and Zhenjun Han. Scale match for tiny person detection. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1246–1254, 2020.
- Richard Zhang. Making convolutional networks shift-invariant again. *arXiv: Computer Vision and Pattern Recognition*, Apr 2019.
- Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z. Li. S3FD: Single shot scale-invariant face detector. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- Zhishuai Zhang, Wei Shen, Siyuan Qiao, Yan Wang, Bo Wang, and Alan L. Yuille. Robust face detection via learning small faces on hard images. *CoRR*, abs/1811.11662, 2018. URL <http://arxiv.org/abs/1811.11662>.
- Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. *arXiv: Computer Vision and Pattern Recognition*, Oct 2020a.
- Yanjia Zhu, Hongxiang Cai, Shuhan Zhang, Chenhao Wang, and Yichao Xiong. TinaFace: Strong but simple baseline for face detection. *arXiv preprint arXiv:2011.13183*, 2020b.
- Xueyan Zou, Fanyi Xiao, Zhiding Yu, and Yong Jae Lee. Delving deeper into anti-aliasing in ConvNets. In *BMVC*, 2020.