*ECAI 2024*
*U. Endriss et al. (Eds.)*


*doi:10.3233/FAIA240536*

# Detecting Objects as Cascade Corners

**Chenglong Liu[a,†], Jintao Liu[a,†], Haorao Wei[a], Jinze Yang[a], Liangyu Xu[a], Yuchen Guo[b,*] and Lu Fang[b]**

[a]University of Chinese Academy of Sciences
[b]Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing, China

**Abstract.** The corner-based detection paradigm enjoys the potential to produce high-quality boxes. But the development is constrained by three factors: 1) Hard to match corners. Heuristic corner matching algorithms can lead to incorrect boxes, especially when similar-looking objects co-occur. 2) Poor instance context. Two separate corners preserve few instance semantics, so it is difficult to guarantee getting both two class-specific corners on the same heatmap channel. 3) Unfriendly backbone. The training cost of the hourglass network is high. Accordingly, we build a novel corner-based framework, named Corner2Net. To achieve the corner-matching-free manner, we devise the cascade corner pipeline which progressively predicts the associated corner pair in two steps instead of synchronously searching two independent corners via parallel heads. Corner2Net decouples corner localization and object classification. Both two corners are class-agnostic and the instance-specific bottom-right corner further simplifies its search space. Meanwhile, RoI features with rich semantics are extracted for classification. Popular backbones (*e.g.*, ResNeXt) can be easily connected to Corner2Net. Experimental results on COCO show Corner2Net surpasses all existing corner-based detectors by a large margin in accuracy and speed.

## 1 Introduction

Object detection [21, 12, 24, 20, 13, 25] is a research hotspot in computer vision, which aims to know where and what objects are in a given image. With the development of deep learning [9], the accuracy and efficiency of object detection have been greatly improved, enabling it to develop into practical applications.

In the current era, according to the representations of modeling objects, more popular object detectors are the center-based type. They encode a bounding box by taking the object center as a reference and regressing the height/width. More specifically, the classic center-based methods [21, 1] initialize each pixel as a center point and place a large number of artificially sized anchors. Later, some excellent center-based detectors [19, 12, 24, 23] are proposed to optimize the learning of dense anchor queries and make them more sparse and efficient. Different from the aforementioned methods, to get rid of anchors relying on manually set hyperparameters, a corner-based detector [10] has emerged. This model constructs a detection box by predicting its top-left and bottom-right corner keypoints and matching them into a pair. Afterward, some impressive efforts [5, 26, 14] have been made to pursue more robust corner matching algorithms with higher quality and higher reliability.

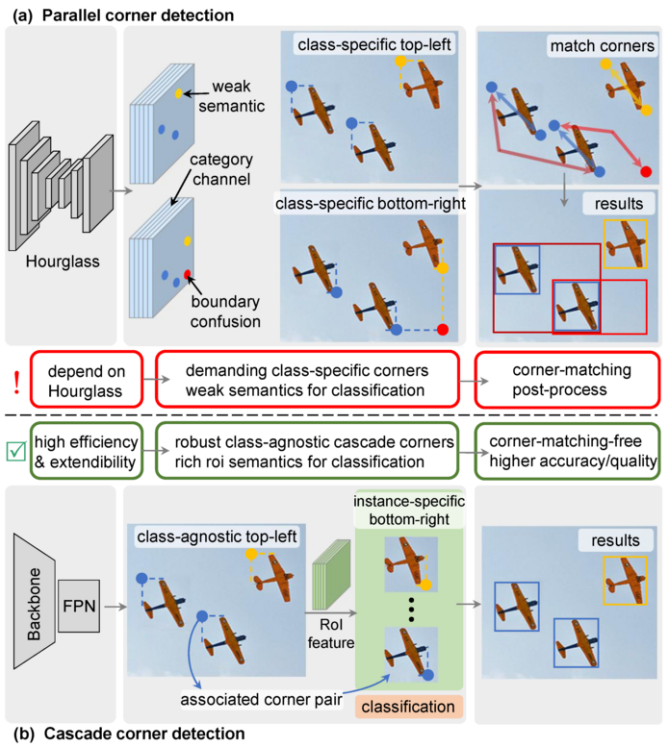


**Figure 1.** Comparision of parallel corner detection and the proposed cascade corner detection. All existing corner-based methods fall into the parallel detection pipeline, which predicts two separate class-specific corners and relies on the corner-matching algorithm when decoding boxes. Corner2Net adopts the proposed cascade corner detection pipeline that decouples the corner localization and object classification. Corner2Net mines all objects via the class-agnostic associated corner pair which is more robust, and the instance-specific bottom-right corner further simplifies its search space. The object category is predicted using RoI features with rich instance semantics. Our Corner2Net runs in a corner-matching-free manner, and it can achieve higher accuracy and efficiency under various backbones and is not limited to the Hourglass network.

Actually, the center-based modeling approaches can be harder to determine object boundaries via its center, because they directly optimize the bounding box with four degrees of freedom and most training pipelines are less sensitive to precise object boundaries. Compared to them, the corner-based methods only require two independent corner keypoints with two degrees of freedom each. Further, this corner-based paradigm is similar to what annotators do, which is to label a precise bounding box from the top-left corner to the bottom-

* Corresponding Author. Email: yuchen.w.guo@gmail.com
† Equal contribution.

right corner. Thus, the idea of detecting an object with two corners implies strong human empirical knowledge. We hold the view that the corner-based paradigm enjoys excellent research value and the potential to produce high-quality boxes.

In this era where center-based detectors dominate, there are several reasons why corner-based ones are not as popular. First, and most importantly, the process of constructing a box from two independent corners relies on a heuristic matching algorithm, which needs to filter out ambiguous corner keypoints caused by boundary confusion and find out the corner pair belonging to the same object instance (shown in Figure 1 (a)), while its reliability and preciseness are not perfect. Second, it is difficult to predict class-specific keypoints because the corners preserve poor instance context. This is unfavorable for the way of obtaining the object category by the same channel ID of the heatmap where the corresponding corner pair lies. Sometimes one of the two class-specific corners is wrongly estimated due to the weak robustness and learning difficulty, which directly leads to the loss of a detection box. Third, the existing corner-based detectors all adopt the keypoint-friendly backbone Hourglass [17] network, which has low training efficiency and slow inference speed, making it difficult to deploy in practical applications.

Accordingly, to address these issues, we build a novel corner-based framework, named Corner2Net, which focuses more on each instance and frees the corner detector from the shackles it carried before. In general, Corner2Net decouples corner localization and object classification and changes the conventional parallel prediction to a cascade prediction pipeline (see Figure 1 (b)). To alleviate the learning difficulty and fully mine all the potential objects, Corner2Net locates objects using class-agnostic associated corner pairs instead of hard class-specific independent keypoints. Moreover, each bottom-right corner is determined within the corresponding instance-specific RoI space to further simplify corner search. Obviously, the associated cascade corner pair enables Corner2Net to get rid of the dependence on the matching algorithm and directly produce a box. Meanwhile, to improve classification accuracy, the category of each object is obtained by feeding RoI features into a lightweight head, which maintains rich instance context. For more details, unlike existing corner-based methods [10, 5, 26] that use a single feature from Hourglass network, we exploit the multi-level prediction in FPN layers to fully adapt to objects of different sizes when pinpointing corners. In addition, our Corner2Net can easily connect popular backbones (*e.g.*, ResNeXt [27]) and yield a pleasing performance.

We evaluate the proposed Corner2Net on three challenging datasets, *i.e.*, MS-COCO [11], CityPersons [3] and UCAS-AOD [33]. Corner2Net achieves an AP of 47.8% under the ResNeXt-101-DCN backbone on the COCO test-dev, exceeding all existing corner-based detectors by a large margin. Without the cumbersome hourglass and corner-matching process, Corner2Net enjoys a 2.1 times faster inference speed than the CornerNet baseline. Furthermore, the remarkable $\mathrm{AP}_{80}$/$\mathrm{AP}_{90}$ of 44.6%/22.4% verify the superiority of producing high-quality boxes. In addition, on the CityPersons and UCAS-AOD datasets that baseline detector CornerNet struggle with, the proposed Corner2Net gains significant improvements of 36.2% and 18.0% on $\mathrm{AP}_{50}$, respectively. This indicates that our Corner2Net enjoys strong robustness and applicability. We firmly believe that the neat yet efficient Corner2Net can become an excellent baseline in the corner detection paradigm.

To summarize, our contributions are three folds:

- We propose an innovative corner-based detector named Corner2Net, achieving a corner-matching-free manner by modeling objects as associated cascade corners.
- Corner-based localization and classification are decoupled and optimized by class-agnostic corner pairs and RoI semantics. Corner2Net can robustly benefit from mainstream backbones.
- Corner2Net far outperforms all existing corner detectors in terms of accuracy and speed. Further, the quality of detection boxes is significantly improved.

## 2 Related Work

In this section, we briefly review some related approaches, mainly involving center-based and corner-based detectors.

### *2.1 Center-based Detectors*

The center-based paradigm is a research hotspot in object detectors. They model the bounding box from the center of the object and encode its length and width. The famous Faster-RCNN [21] places numerous dense center-based anchors and regresses the offsets between the initial anchors and the ground truths. To improve efficiency and practicability, YOLO series [20, 19] abandons the proposal stage and directly performs regression and classification for each box. Libra R-CNN [18] investigates the impact of the training process on the anchor-guided detector and optimizes it by the balanced learning at the sample level, feature level, and objective level. Later, FCOS [24] performs per-pixel prediction within the center of the object and regresses a 4D vector. Later, Dynamic R-CNN [29] uses dynamic statistics of proposals instead of static configuration during training to make the detector more effective. Sparse RCNN [23] discards conventional dense priors and employs a few learnable boxes to detect objects, which eliminates many-to-one label assignment and non-maximum suppression post-processing. DETR [2] solves the detection problem as a direct set prediction and designs the transformer architecture to query the position of objects based on centers. Center-based detectors have undoubtedly achieved great results, however, this basic idea of requiring four boundaries to search the center of an object may limit their performance ceiling.

### *2.2 Corner-based Detectors*

The corner-based paradigm adopts a more bionic route, which evades the anchor generation mechanism and then constructs the bounding box of the instance by pairing corners. The concept is first introduced by CornerNet [10], which predicts the top-left and bottom-right corners on heatmaps separately and matches them by measuring the distance of the corresponding instance embedding. To better perceive the visual patterns within proposals and suppress the wrong decoded boxes, CenterNet [5] improves the correctness of the corner pair by upgrading them into keypoint triplets via an additional predicted center point. With the combination of position and geometry information, CentripetalNet [4] devises a 2D centripetal shift to clearly judge whether the two corners are from the same instance. With greater progress, an affinity function [26] is devised to boost the accuracy and box quality by incorporating both structure and context measurements. These works all contribute to optimizing the accuracy of the corner-matching process, yet it is still not perfect. Furthermore, the difficulty of learning two corners of the same category due to weak semantics is not considered. And they all adopt Hourglass as the backbone which is not friendly to follow. In this paper, we refresh the corner-based detection framework and propose Corner2Net to fully demonstrate the true power of this detection paradigm.

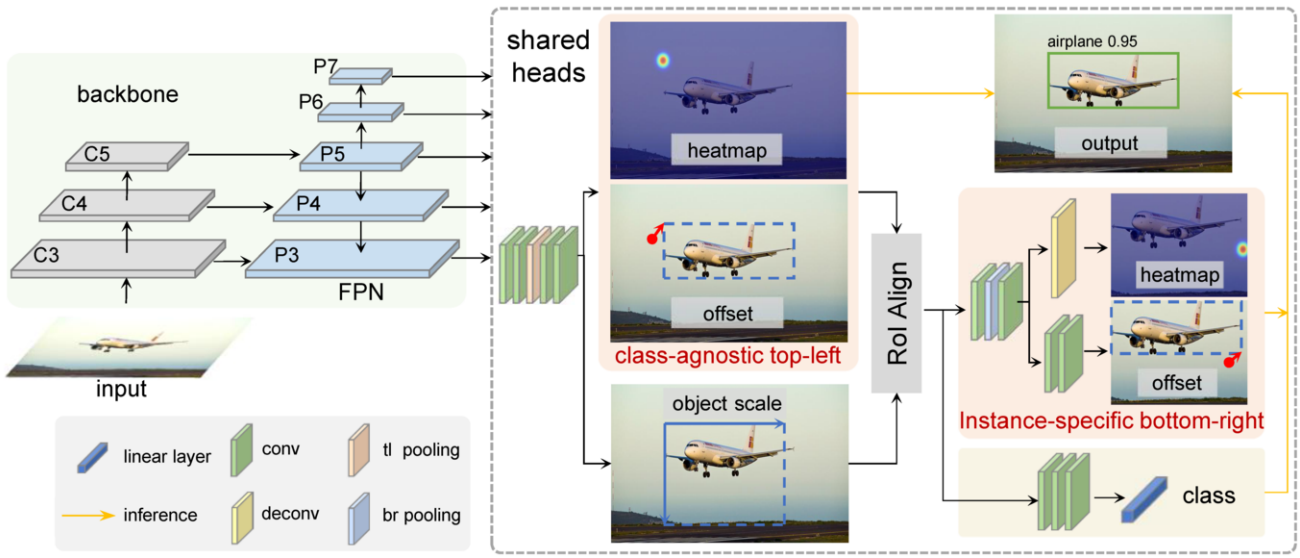


**Figure 2.** The framework of the proposed Corner2Net. Corner2Net has two cascade stages to conquer the corner-based detection task. In the first stage, all class-agnostic top-left corners are located on the image-level heatmap and adjusted by offsets, and each RoI space is determined to establish an association between two cascade corners. Next, the RoI features are fed into the second stage, where the precise bottom-right corner of each specific instance is obtained by its instance-level heatmap and offset. Meanwhile, the category is predicted by a lightweight head with rich instance semantics.

## 3 Method

### 3.1 Framework of Cascade Corner Detection

The corner-based detection paradigm is to determine the position of the object by finding its top-left and bottom-right corners. Existing methods utilize two parallel detection heads to predict two class-specific keypoints respectively, thus the post-processing is necessary to pair the independent corners into boxes. Different from the aforementioned pipeline, we divide the prediction task of the two corners into two steps instead of synchronously predicting them. As shown in Figure 2, the first step is to exclusively mine the top-left corner points of all instances regardless of category, and return the RoI space for each instance. Next, the second step is to extract the RoI features with rich context to predict the more precise instance-specific bottom-right corner by the single-channel heatmap. And the object category is also obtained in the second step with a simple classification head via instance semantics. Obviously, these two solving steps endow the proposed Corner2Net to be a two-stage detector. Next, we elaborate on each component for cascade corner detection.

The previous corner-based detectors [10, 5, 26] are all developed based on the keypoint-friendly hourglass-104 [17] backbone to gain the best performance. But it has unpleasant training costs and inference speed. Instead of that, to align with commonly used strategies in other detectors, we also exploit the structure of multi-level feature prediction. Given an image with $H \times W$ pixels, it is input into the backbone to gain high-level semantics. The deep feature maps C3 to C5 are extracted and fed into FPN. The feature levels P3 to P7 of FPN are used for subsequent prediction through the shared heads. The downsampling ratios $s \in (8, 16, 32, 64, 128)$ lead to the output maps with the size of $\frac{H}{s} \times \frac{W}{s}$. In addition, we assign objects of different sizes to FPN layers based on the previous strategy [24, 31]. Thus, Corner2Net uses multi-level features to help predict the corners of multi-scale objects. Corner2Net can flexibly connect common backbones (*e.g.*, ResNet [7], and ResNeXt [27]), and experimental results prove that robust accuracy can be obtained.

### 3.2 Mine Every Class-agnostic Top-left Corner

In the proposed cascade corner detection framework, the main goal of the first stage is to fully explore the top-left corner keypoints of all instances. We decouple position and category prediction and only estimate the class-agnostic keypoints for the top-left corner. Specifically, at each FPN feature level, Corner2Net predicts a $\frac{H}{s} \times \frac{W}{s}$ heatmap with a single channel via several convolution layers to estimate all top-left corners of the cared objects at the current level. We also adopt the corner pooling [10] to encode more visual evidence for the corners. For each top-left corner, the only ground-truth pixel on the heatmap is positive and set to 1 and other pixels are negative and set to 0. During training, the corner target is mapped as a Gaussian region [10], reducing the penalty for the pixels near the ground-truth corners and helping balance positive and negative samples. We employ the class-agnostic focal loss [12] with the distance-aware penalty as the optimization target of the top-left corner heatmap, *i.e.*,

$$\mathcal{L}_h^{tl} = -\frac{1}{N_p} \sum_{x=1}^{W'} \sum_{y=1}^{H'} \begin{cases} (1-p'_{xy})^{\alpha} \log(p'_{xy}), & \text{if } p_{xy}=1 \\ (1-p_{xy})^{\beta} (p'_{xy})^{\alpha} \log(1-p'_{xy}), & \text{otherwise} \end{cases} \quad (1)$$

where $N_p$ denotes the sum of objects. $W'$ and $H'$ are the width and height of the heatmap. $p_{xy}$ is the ground-truth value at the coordinate $(x, y)$ on the top-left heatmap and $p'_{xy}$ is the predicted value in the same location. The hyper-parameter $\alpha$ can adjust the weights for hard and easy samples. The $\beta$ is used to control the term of distance-aware penalty. Referring to previous work [10], we set $\alpha$ and $\beta$ to 2 and 4.

After getting the multi-level predicted heatmaps, we need to extract keypoints with higher scores and map the coordinates back to the original image resolution. Due to the existence of the downsampling ratio at each layer of FPN, there is a discretization error for the coordinates on the downsampled heatmap. To compensate for this lost precision, Corner2Net predicts a position offset map for each top-left heatmap at different FPN levels, as shown in Figure 2. We adopt the Smooth L1 Loss [6] as the objective of the offset map, *i.e.*,

$$\mathcal{L}_o^{tl} = \frac{1}{N_p} \sum_{j=1}^{N_p} \text{SmoothL1Loss}((\Delta x'_j, \Delta y'_j), (\Delta x_j, \Delta y_j)) \quad (2)$$

where $(\Delta x'_j, \Delta y'_j)$ are the predicted values on the output offset map. $(\Delta x_j, \Delta y_j)$ are the corresponding ground truth and can be calculated by $(x_j - x_j^f, y_j - y_j^f)$. Here, $(x_j, y_j)$ is the discretized position on the heatmap and $(x_j^f, y_j^f)$ is the float coordinate of the ground-truth corner. $N_p$ means the number of positives. Thus, the extracted top-left corner location on the heatmap can be refined by the predicted offset. It is worth mentioning that this offset compensation is of great significance to the detection for small objects.

### 3.3 Establish Association between Corners via the RoI Space

To achieve cascade corner detection, we intend to pinpoint the specific bottom-right keypoint for each top-left corner via its region of interest. This means that we need to determine the RoI space for establishing an association between the top-left and bottom-right corners in the first stage. In each FPN layer, Corner2Net also outputs a regression map that indicates an instance scale at each top-left positive position. For training, we utilize the GIoU loss [22] as the objective function, which is formulated as:

$$\mathcal{L}_b = \frac{1}{N_p}\sum_{j=1}^{N_p}\mathcal{L}_{GIoU}\left[(z_j, z_j), (z_j^f, z_j^f)\right] \tag{3}$$

where $z_j$ and $z_j^f$ are the predicted scale and the ground-truth value, respectively. The ground-truth scale is the maximum value of the object's width/height. At this point, we can decode a proposal box $(x_{tl}, y_{tl}, z, s_{tl})$ at the top-left corner location. Here, $(x_{tl}, y_{tl})$ is the top-left corner coordinate refined by its offset and $s_{tl}$ is the confidence predicted on the heatmap. The width and height of the proposal are equal to the approximate scale $z$, that is, the expected RoI space is square. The square RoI space can help minimize the axial compression ratio, so that the feature blocks obtained by RoIAlign [8] can retain more guidance of spatial information.

Note that the proposal RoI space is provided via a rough predicted object scale, which cannot completely enclose the instance. Accordingly, to ensure that the bottom-right corner can fall inside the proposal RoI, we expand the vanilla RoI space by enlarging its regressed scale, and related formulations are described as follows:

$$(x_{tl}, y_{tl}, z_e, s_{tl}) \longleftarrow (x_{tl}, y_{tl}, (1+\eta)\cdot z, s_{tl}) \tag{4}$$

where $z_e$ is the enlarged scale and $\eta$ denotes the enlarge factor. Then, Corner2Net can locate the bottom-right keypoint of each instance within the specific enlarged RoI space via the next stage.

### 3.4 Locate Each Instance-specific Bottom-right Corner

In the second stage of the cascade corner detection framework, the bottom-right corner is predicted for the specific instance in the RoI space determined by the corresponding top-left corner, as shown in Figure 2. Clearly, our solution devises two cascade corners (or an associated corner pair) instead of two completely independent corners, avoiding the heuristic corner-matching algorithms that are indispensable in parallel corner-based pipelines [10, 5, 26].

We use the RoIAlign [8] operation with an output size of $14 \times 14$ to extract the features used to locate bottom-right corners. Then, there are two convolutions with a bottom-right corner pooling layer in between. Next, a deconvolution layer is placed behind to output a single-channel heatmap with the $28 \times 28$ pixels. For training, similarly to the top-left point, we render the unique ground-truth bottom-right corner of each object as a Gaussian region [10] by $\exp(-\frac{x_k^2+y_k^2}{2\sigma^2})$, and the radius $\sigma$ is calculated by the object size. After that, we need to project each point $(x_k, y_k)$ in the RoI region of the original image space onto the coordinate system of the heatmap branch. The transformation can be formulated as:

$$\begin{cases} x_h = (x_k - x_{tl}) \cdot \frac{m}{z_e} \\ y_h = (y_k - y_{tl}) \cdot \frac{m}{z_e} \end{cases} \tag{5}$$

where $(x_h, y_h)$ is the coordinate on the heatmap. $m$ denotes the side length of the heatmap and the default value is 28. $x_{tl}$, $y_{tl}$, and $z_e$ are the same as in Eq. 4. And we employ the binary cross entropy loss to train the instance-specific bottom-right heatmap branch, and the function is as follows:

$$\begin{aligned}\mathcal{L}_h^{br} = \frac{1}{N_{fb}\cdot m^2}\sum_{n=1}^{N_{fb}}\sum_{x=1}^{m}\sum_{y=1}^{m} & -p_{nxy}\cdot\log\left(\phi(p_{n'xy})\right) \\ & -(1-p_{nxy})\cdot\log\left(1-\phi(p_{n'xy})\right)\end{aligned} \tag{6}$$

where $p_{nxy}$ and $p_{n'xy}$ are the ground truth and predicted value, respectively. $\phi(\cdot)$ denotes the sigmoid function. $N_{fb}$ is the number of foreground RoIs. For inference, the predicted heatmap first undergoes sigmoid function and $3 \times 3$ max pooling, and then the top-1 keypoint is picked to decode the bottom-right corner location.

It is easy to find that, as illustrated in Eq. 5, there is a quantization error in the instance-specific heatmap. Therefore, it is necessary to return an offset to compensate for the precision loss. Specifically, along with the heatmap branch, we apply two convolution layers to predict an offset for the specific bottom-right corner, so as to fine-tune the position of the bottom-right keypoint. During training, the Smooth L1 Loss [6] is chosen as the objective function:

$$\mathcal{L}_o^{br} = \frac{1}{N_b}\sum_{j=1}^{N_b}\mathrm{SmoothL1Loss}((\Delta u'_j, \Delta v'_j), (\Delta u_j, \Delta v_j)) \tag{7}$$

where $N_b$ is the number of RoI spaces. $\Delta u'_j$ $\Delta v'_j$ are the predicted offsets in the horizontal and vertical directions. During inference, the coordinates of the bottom-right corner $(x_{br}, y_{br})$ on the input image can be calculated by Eq. 8.

$$\begin{cases} x_{br} = \left(x'_{br} + \Delta u'_j\right)\cdot\frac{z_e}{m} + x_{tl} \\ y_{br} = \left(y'_{br} + \Delta v'_j\right)\cdot\frac{z_e}{m} + y_{tl} \end{cases} \tag{8}$$

where $(x'_{br}, y'_{br})$ is the keypoint extracted on the bottom-right heatmap. $z_e$ denotes the enlarged scale of RoI space and $m$ is the same as in Eq. 5. Finally, we obtain the purified bottom-right corner and can utilize this associated corner pair to decode a high-quality detection box.

### 3.5 Predict Category by RoI Features with Rich Instance Semantics

In the proposed Corner2Net, the classification score can be obtained by a lightweight head in the second stage. Compared to those existing corner-based detectors that determine the object category through the channel id of unrobust class-specific keypoints, Corner2Net utilizes several convolutions and two fully connected layers to perform classification by RoI features which preserve richer instance semantics. To alleviate the interference of background noise on the classifier,

we utilize vanilla RoIs without being enlarged to produce the features with the spatial size of $7 \times 7$ to be classified. And we adopt the cross entropy [21] loss function to train the classification head. During inference, we take into account both the localization score and the classification score. Note that the score of the cascade corner pair represents the localization score of the constructed detection box. Then, the score of the final detection box is computed as follows:

$$s_{box} = \sqrt{[0.5\,(s_{tl} + s_{br})]_{loc} \cdot s_{cls}} \tag{9}$$

where $s_{cls}$ is the predicted class confidence. $[\cdot]_{loc}$ represents the localization term, which is the arithmetic mean of the top-left and bottom-right corner confidence.

## 4 Experiments

### 4.1 Datasets and Evaluation Metrics

**MS-COCO.** MS-COCO [11] is a large-scale detection dataset with a resolution of about $600 \times 800$ and labels containing 80 categories. Our Corner2Net is trained on the train2017 set including $118k$ images. We report the performance on the test-dev set including $20k$ images for the state-of-the-art comparison with other published detectors. We perform ablation studies on the COCO val2017 set including $36k$ instances within $5k$ images.

**CityPersons.** We perform evaluations on CityPersons [3] dataset to verify the robustness of our model in the scene with dense occlusion. We filter the vanilla dataset, and merge the main annotations of pedestrians and riders. As a result, the training set contains 2471 images and 18k persons. The evaluation results are achieved on the validation set including 439 images and 3666 persons.

**UCAS-AOD.** UCAS-AOD [33] is selected to demonstrate the generalization and applicability of the proposed Corner2Net when facing many similar and evenly symmetrically distributed targets. There are 1000 high-resolution aerial images in this dataset. And the total number of annotated aircraft is 7482.

**Evaluation metrics.** We adopt commonly used AP (Average Precision) and FPS (frames per second) as metrics to measure detection accuracy and inference speed, respectively.

### 4.2 Implementation Details

During training, we adopt the SGD optimizer to minimize the training loss and we initialize the backbone using parameters pretrained on the ImageNet [9]. And our model is trained on 8 RTX 3090 GPUs with two images on each for 24 epochs. The initial learning rate is set to 0.02 for the first 16 epochs, and it is decayed by $\times 10$ at the 16th epoch and 22nd epoch.

During inference, the predicted class-agnostic top-left heatmap is performed $3 \times 3$ max pooling, and then the top 128 proposal keypoints are extracted to search each specific bottom-right corner. The duplicate boxes are filtered out by NMS with an IoU threshold of 0.6. For the single-scale testing, each image is resized to a fixed short side of 640 pixels. When performing multi-scale testing, the short side of each resized image is in the range of $[400 : 200 : 1400]$.

### 4.3 Comparison with State-of-the-art Methods

**Main results on COCO.** To demonstrate the performance of the proposed Corner2Net, we conduct the accuracy comparison with state-of-the-art detectors on the COCO test-dev, and the evaluation results are reported in Table 1. Generally, the proposed Corner2Net equipped with the backbone ResNeXt-101-DCN-32×8d can achieve the AP of 46.8%/47.8% under the single-scale/multi-scale testing, outperforming the counterparts with the same settings.

**Corner2Net versus parallel corner-based detectors.** For corner-based detection methods, which is the paradigm to which our method belongs, our method showcases state-of-the-art accuracy and requires only a small number of training epochs. Specifically, in comparison with CornerAffinity [26] that enjoys an almost ceiling-level corner-matching algorithm, our Corner2Net brings the $+1.7\%\mathrm{AP}_{50}$ improvement using 7.5% epochs (24 *vs.* 320). This is because Corner2Net avoids struggling in matching unrobust class-specific corners via reliable cascade class-agnostic corners.

**Corner2Net versus center-based detectors.** Benefiting from decoupled instance-informative-rich classification and matching-free corner detection, Corner2Net surpasses the common center-based FCOS [24] (*i.e.*, 46.8% *vs.* 46.6% on AP) and SAPD [32] under the same settings. Moreover, compared to the performance from popular center-based detectors, *e.g.*, Cascade R-CNN [1] and Reppoints [28], Corner2Net with the same ResNet-101 backbone also lifts AP by at least 1.0%. This phenomenon shows that searching for corner points is easier than center points when detecting objects, which supports the opinion stated in Section 1. Notably, compared to Grid R-CNN [16] using multiple keypoints to localize centers in the second stage, the AP of Corner2Net increases by 2.3%, which demonstrates the superiority of the proposed framework that directly find two precise corner kepoints in two stages.

**Training efficiency.** Besides, compared to other corner detectors that rely on hourglass networks and require 200/300 training epochs, Corner2Net with SwinTransformer (tiny) [15] backbone only needs 24 epochs to yield a pleasing AP of 46.0%, surpassing the CornerNet [10] baseline by 5.4% AP. This shows that our corner detection framework is more efficient and convenient to explore further.

**Quality of the detection box.** We hold the view that corner-based detectors can generate higher-quality boxes than center-based ones. To examine this, we conduct the accuracy evaluation at high IoU thresholds on MS-COCO val2017. As shown in Table 2, using only 12% of the training epochs of CornerNet, our model comprehensively outperforms CornerNet in several indicators, including $\mathrm{AP}_{50}$, $\mathrm{AP}_{60}$, $\mathrm{AP}_{70}$, and $\mathrm{AP}_{80}$. It is worth noting that our Corner2Net reaches an impressive $\mathrm{AP}_{80}$ of 44.6%, surpassing CornerNet by a large margin of 5.8%. The 22.4% $\mathrm{AP}_{90}$ of Corner2Net has a 1% gap compared to CornerNet. This is because the downsampling stride of the feature map in Corner2Net is much larger than that of the vanilla CornerNet, making the compensation of quantization error more difficult. Nonetheless, these results are sufficient to demonstrate that the proposed Corner2Net has achieved a reasonable trade-off between box quality and computational costs.

### 4.4 Comparisons on Inference Speed

The proposed Corner2Net no longer relies on the sluggish hourglass backbone network, and matching-free cascade corner pairs facilitate a faster inference speed. We test the inference speed of all models on Titan Xp GPU. As shown in Table 1, CornerNet, CenterNet [5], CentripetalNet [4], CornerAffinity [26] and our Corner2Net yield 3.9, 3.3, 3.4, 3.7 and 8.0 FPS, respectively. Further, our Corner2Net with SwinTransformer (tiny) is 2.9 times faster than vanilla CornerNet with Hourglass-104 and lifts AP by 5.4%. During inference, CornerNet [10] processes $100^2$ proposals constructed by 100 top-left and

**Table 1.** Comparisons with state-of-the-art methods in term of accuracy (%) on the MS-COCO test-dev set. ($*$) means the implementation of multi-scale testing. The full name of "DCN" is the Deformable Convolutional Networks. "T" means "tiny".

| Method **(test-dev set)** | Backbone | FPS | Params (M) | Epoch | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|---|---|---|
| **Center-based:** | | | | | | | | | | |
| Faster R-CNN [21] | ResNet-101 | 9.5 | 60.75 | 24 | 36.2 | 59.1 | 39.0 | 18.2 | 39.0 | 48.2 |
| Cascade R-CNN [1] | ResNet-101 | 8.1 | 88.39 | 24 | 42.8 | 62.1 | 46.3 | 23.7 | 45.5 | 55.2 |
| RetinaNet [12] | ResNet-101 | 8.9 | 56.96 | 36 | 40.8 | 61.1 | 44.1 | 24.1 | 44.2 | 51.2 |
| Grid R-CNN [16] | ResNet-101 | 8.0 | 83.31 | 24 | 41.5 | 60.9 | 44.5 | 23.3 | 44.9 | 53.1 |
| Libra R-CNN [18] | ResNet-101 | 8.8 | 60.78 | 24 | 41.1 | 62.1 | 44.7 | 23.4 | 43.7 | 52.5 |
| Reppoints [28] | ResNet-101 | 8.4 | 55.62 | 24 | 41.0 | 62.9 | 44.3 | 23.6 | 44.1 | 51.7 |
| CenterNet [30] | Hourglass-104 | 4.2 | 186.44 | 200 | 42.1 | 61.1 | 45.9 | 24.1 | 45.5 | 52.8 |
| FCOS [24] | ResNeXt-101-DCN | 6.9 | 89.79 | 24 | 46.6 | 65.9 | 50.8 | 28.6 | 49.1 | 58.6 |
| SAPD [32] | ResNeXt-101-DCN | 6.2 | 101.20 | 24 | 46.6 | 66.6 | 50.0 | 27.3 | 49.7 | 60.7 |
| **Parallel corner-based:** | | | | | | | | | | |
| CornerNet [10] | Hourglass-104 | 3.9 | 192.04 | 200 | 40.6 | 56.4 | 43.2 | 19.1 | 42.8 | 54.3 |
| CenterNet [5] | Hourglass-104 | 3.3 | 201.20 | 190 | 44.9 | 62.4 | 48.1 | 25.6 | 47.4 | 57.4 |
| CentripetalNet [4] | Hourglass-104 | 3.4 | 197.76 | 210 | 45.8 | 63.0 | 49.3 | 25.0 | 48.2 | 58.7 |
| CornerAffinity [26] | Hourglass-104 | 3.7 | 194.60 | 320 | 46.3 | 64.0 | 49.9 | 27.4 | 49.3 | 58.7 |
| **Cascade corner-based:** | | | | | | | | | | |
| Corner2Net (Ours) | ResNeXt-101-DCN | 8.0 | 119.23 | **24** | **46.8** | 65.7 | **51.9** | **29.3** | **49.8** | 57.9 |
| Corner2Net$^*$ (Ours) | ResNeXt-101-DCN | - | - | 24 | 47.8 | 65.5 | 53.9 | 31.5 | 50.2 | 57.6 |
| Corner2Net (Ours) | ResNet-101 | 9.4 | 67.87 | 24 | 43.8 | 61.8 | 48.5 | 26.5 | 46.7 | 53.7 |
| Corner2Net$^*$ (Ours) | ResNet-101 | - | - | 24 | 45.0 | 61.8 | 50.6 | 29.5 | 47.3 | 53.8 |
| Corner2Net (Ours) | SwinTransformer-T | **11.2** | **53.03** | 24 | 46.0 | 65.2 | 50.7 | 28.2 | 48.4 | 57.9 |
| Corner2Net$^*$ (Ours) | SwinTransformer-T | - | - | 24 | 46.6 | 63.9 | 52.5 | 30.2 | 49.0 | 57.8 |

**Table 2.** Comparisons in terms of box quality on MS-COCO val2017 set. Higher IoU corresponds to higher-quality detection boxes. "-" denotes the model weights of the original paper are not available.

| Method **(val set)** | Backbone | Epoch | AP | $AP_{50}$ | $AP_{60}$ | $AP_{70}$ | $AP_{80}$ | $AP_{90}$ |
|---|---|---|---|---|---|---|---|---|
| CornerNet [10] | Hourglass-104 | 200 | 40.6 | 56.1 | 52.0 | 46.8 | 38.8 | 23.4 |
| CenterNet [5] | Hourglass-104 | 190 | 44.8 | 62.5 | 58.1 | 52.3 | 42.9 | 22.9 |
| CentripetalNet [4] | Hourglass-104 | 210 | 44.7 | 62.6 | 58.1 | 52.8 | 42.3 | 22.3 |
| CornerAffinity [26] | Hourglass-104 | 320 | 45.1 | 62.9 | - | - | - | - |
| Corner2Net (Ours) | ResNeXt-101-DCN | **24** | **46.3** | **65.1** | **61.5** | **55.6** | **44.6** | 22.4 |

**Table 3.** Accuracy comparisons on CityPersons ($AP^c_{50}$) and UCAS-AOD ($AP^u_{50}$) dataset. SwinT-T means SwinTransformer (tiny).

| Model | Backbone | epoch | $AP^c_{50}$ | $AP^u_{50}$ |
|---|---|---|---|---|
| CornerNet | Hourglass-104 | 50 | 29.1 | 79.1 |
| CenterNet | Hourglass-104 | 50 | 51.5 | 86.8 |
| CentripetalNet | Hourglass-104 | 50 | 54.7 | 93.6 |
| CornerAffinity | Hourglass-104 | 50 | 64.9 | 96.3 |
| Corner2Net (Ours) | SwinT-T | 12 | 65.3 | 97.1 |

**Table 4.** Effect of the number ($k$) of extracted cascade corner pairs. The results are obtained on MS-COCO val2017 under the same backbone SwinTransformer (tiny).

| top-k | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|
| $k$ =100 | 45.6 | 64.5 | 50.0 | 29.8 | 48.7 | 59.7 |
| $k$ =128 | 45.7 | 64.7 | 50.1 | 29.9 | 48.8 | 59.8 |
| $k$ =256 | 45.7 | 64.6 | 50.3 | 30.1 | 48.9 | 59.7 |
| $k$ =512 | 45.6 | 64.5 | 50.3 | 30.1 | 48.9 | 59.7 |
| $k$ =1024 | 45.6 | 64.3 | 50.3 | 30.0 | 48.8 | 59.7 |

100 bottom-right corners, while our Corner2Net only extracts 128 cascade corner pairs to achieve the promising performance.

### *4.5 Comparisons in Extreme Scenarios*

To further demonstrate the practicality and generalization, we evaluate the proposed Corner2Net in some extreme scenes, including the CityPersons dataset with dense occlusion and the UCAS-AOD dataset containing the symmetrical arrangement of numerous rotated similar objects. The results are reported in Table 3. These parallel corner-based methods suffer from performance bottlenecks due to heuristic corner-matching algorithms. Compared to these corner baselines, the proposed cascade corner-based Corner2Net achieves the best $AP^c_{50}$ of 65.3% and $AP^u_{50}$ of 97.1%. This shows that our method of predicting the cascade associated corner pairs is more robust to extreme scenarios and the corner-matching free manner avoids confusion when matching corners of similar objects.

### *4.6 Ablation Study*

In this section, we implement ablation analysis to verify the effectiveness of each component and explore better hyper-parameters. And the results are obtained on the MS-COCO val2017 set.

**Effect of the number of cascade corner pairs.** In the inference stage, the number of cascade corner pairs is equal to the number of top-left keypoints extracted on the heatmap. As shown in Table 4, the accuracy of Corner2Net is not sensitive to the number of corner pairs, proving that it is robust and sufficient to locate objects by predicting class-agnostic cascade corners. We choose the value of $k$ to be 128.

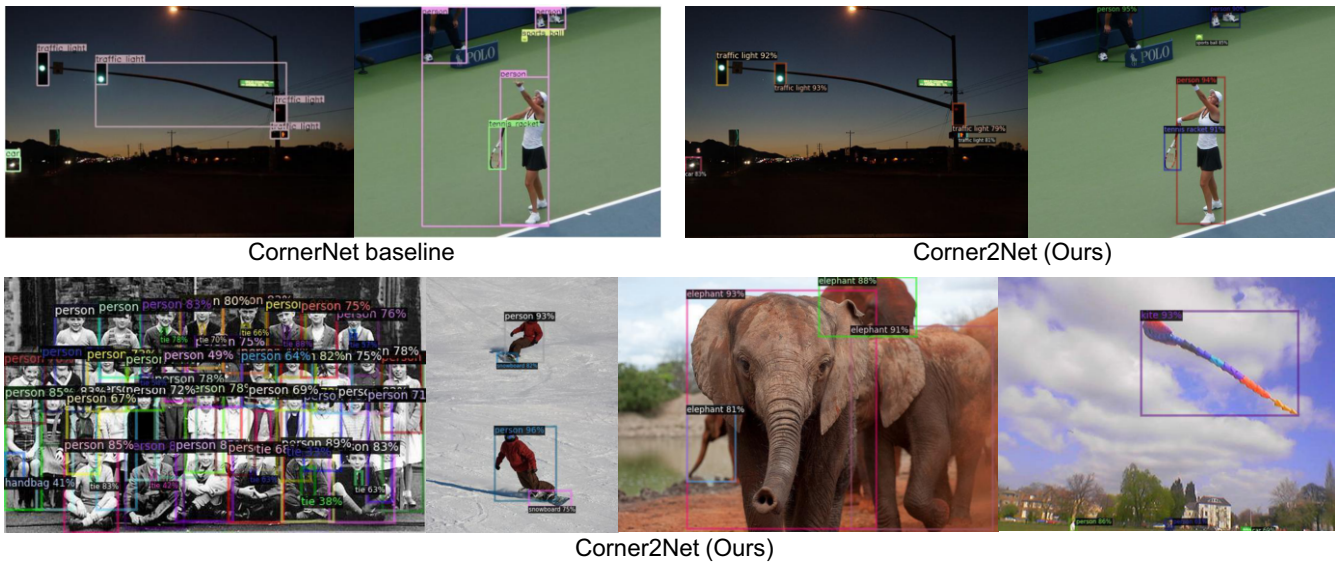


**Figure 3.** Qualitative detection results of CornerNet baseline and the proposed Corner2Net on MS-COCO val2017 set.

**Table 5.** Effect of the enlarge factor ($\eta$) in Eq. 4. The results are obtained on MS-COCO val2017 under the same backbone SwinTransformer (tiny).

| enlarge factor | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|
| $\eta$ =0.00 | 32.5 | 52.5 | 33.5 | 18.3 | 31.2 | 49.7 |
| $\eta$ =0.10 | 45.0 | 64.0 | 49.1 | 28.4 | 48.4 | 59.8 |
| $\eta$ =0.25 | 45.7 | 64.7 | 50.1 | 29.9 | 48.8 | 59.8 |
| $\eta$ =0.35 | 44.8 | 64.1 | 48.7 | 28.9 | 48.2 | 59.3 |
| $\eta$ =0.50 | 42.7 | 62.2 | 46.2 | 26.5 | 46.2 | 57.3 |

**Effect of the enlarge factor.** We conduct experiments to find the optimal enlarge factor. As shown in Table 5, the accuracy with $\eta$ =0.10 results in an AP drop of 0.7%, because some bottom-right corners fall outside the RoI space due to the smaller enlarge factor. A larger factor ($\eta$ =0.5) also leads to a 3% decrease on AP, because the larger RoI space contains more noise, which is not conducive to locating the instance-specific bottom-right corner. Thus, we set $\eta$ to 0.25 in all experiments.

### *4.7 Qualitative Analysis*

**COCO.** As illustrated in Figure 3, CornerNet baseline produces incorrect boxes because its corner-matching algorithm can not distinguish objects with similar appearances. Compared to it, the mismatched corners that often occur in previous corner-based methods do not exist in the results of the proposed Corner2Net due to the corner-matching-free manner. Furthermore, the high-quality visual detection boxes decoded by our Corner2Net have precise boundaries, which verifies the excellent $AP_{80}$ and $AP_{90}$. Hence, the detection box decoded by the proposed cascade corner pair is more reliable than that obtained by matching corners heuristically.

**UCAS-AOD and CityPersons.** Some visualization results of extreme scenarios are shown in Figure 4. CornerNet baseline outputs many false boxes due to confusion about similar objects. CenterNet mismatches some corners because the center points of some other objects may fall within the determination area of the current two corners. The proposed Corner2Net can produce accurate bounding boxes when similar objects co-occur or are partially occluded. This indicates that our model enjoys strong robustness and practicality.

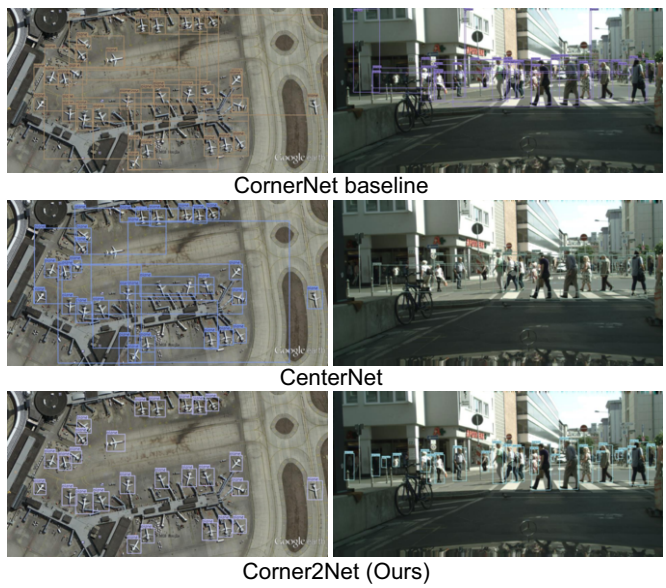


**Figure 4.** Visualization results on UCAS-AOD (left column) and CityPersons (right column) datasets.

## 5 Conclusion

In this paper, we deeply analyze factors limiting the development of parallel corner-based methods and propose a novel cascade corner detection framework to get rid of these constraints. The proposed Corner2Net runs in a corner-matching-free manner and it is also more robust to different popular backbones. Both in accuracy and speed, it surpasses all existing corner-based detectors and enjoys great untapped potential. We hope that this novel cascade corner detection baseline will attract more researchers to revitalize this paradigm with much room for improvement.

## Acknowledgements

This work was supported by National Science and Technology Major Project (No.2022ZD0119402), "Pioneer" and "Leading Goose" R&D Program of Zhejiang (No.2024C01142), National Natural Science Foundation of China (No.U21B2013).

## References

[1] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[2] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[4] Z. Dong, G. Li, Y. Liao, F. Wang, P. Ren, and C. Qian. Centripetalnet: Pursuing high-quality keypoint pairs for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10519–10528, 2020.

[5] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6569–6578, 2019.

[6] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[10] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[12] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[13] C. Liu, H. Yu, H. Wei, X. Sun, and K. Fu. S2cgnet: A robust aircraft detector based on the sword-shaped component geometry. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–15, 2023.

[14] C. Liu, H. Wei, J. Yang, J. Liu, W. Li, Y. Guo, and L. Fang. Gigahumandet: Exploring full-body detection on gigapixel-level images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 10092–10100, 2024.

[15] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[16] X. Lu, B. Li, Y. Yue, Q. Li, and J. Yan. Grid r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7363–7372, 2019.

[17] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 483–499. Springer, 2016.

[18] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 821–830, 2019.

[19] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[21] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[22] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[23] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14454–14463, 2021.

[24] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[25] H. Wei, P. Guo, Y. Zhu, C. Liu, and P. Wang. Humanliker: A human-like object detector to model the manual labeling process. *Advances in Neural Information Processing Systems*, 35:2294–2306, 2022.

[26] H. Wei, C. Liu, P. Guo, Y. Zhu, J. Fu, B. Wang, and P. Wang. Corner affinity: A robust grouping algorithm to make corner-guided detector great again. In *IJCAI*, pages 1458–1464, 2022.

[27] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[28] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9657–9666, 2019.

[29] H. Zhang, H. Chang, B. Ma, N. Wang, and X. Chen. Dynamic r-cnn: Towards high quality object detection via dynamic training. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 260–275. Springer, 2020.

[30] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[31] X. Zhou, V. Koltun, and P. Krähenbühl. Probabilistic two-stage detection. *arXiv preprint arXiv:2103.07461*, 2021.

[32] C. Zhu, F. Chen, Z. Shen, and M. Savvides. Soft anchor-point object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 91–107. Springer, 2020.

[33] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye, and J. Jiao. Orientation robust object detection in aerial images using deep convolutional neural network. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 3735–3739. IEEE, 2015.