Corner Affinity: A Robust Grouping Algorithm to Make Corner-guided Detector Great Again

Haoran Wei[1,†,*], **Chenglong Liu**[1,†], **Ping Guo**[2,‡], **Yangguang Zhu**[1], **Jiamei Fu**[1], **Bing Wang**[1] and **Peng Wang**[2]
[1]University of Chinese Academy of Sciences
[2]Intel Labs China
{weihaoran18, liuchenglong20, wangbing181}@mails.ucas.ac.cn,
{ping.guo, patricia.p.wang}@intel.com, {ygzhu98, fujiamei0508}@gmail.com

Abstract

Corner-guided detector enjoys potential ability to yield precise bounding boxes. However, unreliable corner pairs, generated by heuristic grouping guidance, hinder the development of this detector. In this paper, we propose a novel corner grouping algorithm, termed as Corner Affinity, to significantly boost the reliability and robustness of corner grouping. The proposed Corner Affinity is a couple of two interactional factors, namely, 1) the structure affinity (SA), applied to mine preliminary similarity of corner pairs through the corresponding object's shallow construction knowledge. 2) the contexts affinity (CA), running as optimizing corner similarity via deeper semantic features of affiliated instances. Equipped with the Corner Affinity, a detector can produce high-quality bounding boxes upon preferable paired corner keypoints. Experimental results show the superiority of our design on multiple benchmark datasets. Specifically, compared with CornerNet baseline, the proposed Corner Affinity brings AP boosting of 5.8% on COCO, 35.8% on Citypersons, and 17.2% on UCAS-AOD datasets without bells and whistles. Code will be available at https://github.com/Ucas-HaoranWei/CornerAffinity.

1 Introduction

We roughly classify existing object detectors into center-guided [Ren *et al.*, 2015; Cai and Vasconcelos, 2018; Lin *et al.*, 2017b; Tian *et al.*, 2019; Zhou *et al.*, 2019; Carion *et al.*, 2020; Liu *et al.*, 2021] and corner-guided [Law and Deng, 2018; Duan *et al.*, 2019; Dong *et al.*, 2020], among which we argue the corner-guided manner is more bionic in the production of an object bounding box.

Center-guided methods usually model object upon one center along with width and height yet corner-guided ones utilize an instance via paired of corner keypoints. Intuitively, when we want to obtain high-quality bounding boxes manually, we usually label each box from the top-left corner to the bottom-right one and we rarely label the center point directly. This is because the degree of freedom to determine a corner is lower than a center (a corner needs two boundaries *vs.* a center requires four boundaries) and thus a corner can more easily confirm the objects' boundaries. Accordingly, corner-guided detector enjoys potential ability to yield higher-quality detection boxes. Therefore, we believe this type of detector existing a higher potential deserves to be explored further. However, the challenging corner grouping process hinders the development of this method.

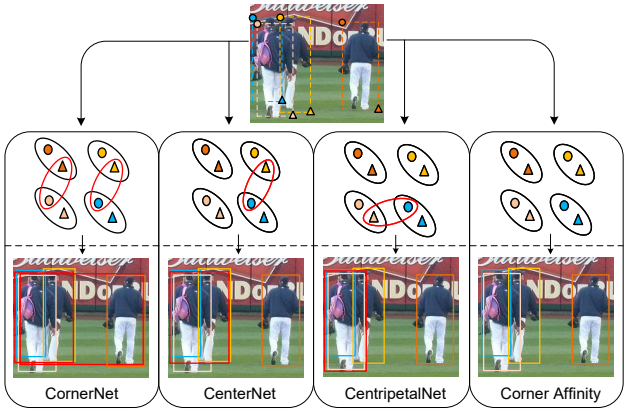

Figure 1: Comparison of CornerNet, CenterNet, CentripetalNet, and our Corner Affinity. "Circle" and "triangle" represent the top-left and bottom-right corners, respectively. Corners with the same color belong to the same instance. The correct grouping results are within the black-line "ellipses" and the wrong ones in the red-line "ellipses".

Although previous methods [Law and Deng, 2018; Duan *et al.*, 2019; Dong *et al.*, 2020] have made great progress in corner grouping, they also face their distinctive shortcomings. As shown in Figure 1, CornerNet, whose corner grouping method relies on local response of object feature, fails to distinguish corners of different objects with similar appearance. Suppose we copy-and-paste an instance to form a new input, CornerNet cannot work because the exact same instance appears twice. Upon CornerNet, CenterNet filters out false-

[*]This work was done when the first author was interning at Intel Labs China.
[†]Equal contribution
[‡]Corresponding author

positives (FP) via center points yet cannot solves the scenario that a center point of the third object lies exactly within the center of an FP. However, this scenario often occurs in some datasets, such as aircraft parked side by side in aerial images. CentripetalNet cannot address the object occlusion problem due to the center overlapping, *e.g.,* pedestrian detection scenario. In summary, existing heuristic corner grouping methods are dataset (COCO [Lin *et al.*, 2014]) driven with poor generalization, but much less effort has been paid to solve it. To tackle this problem, we propose the Corner Affinity, whose motivation is to cover all scenarios aforementioned and bring robust grouping capability for corner-guided detector.

Specifically, Corner Affinity embeds three attributes for each target corner, *i.e.,* locations, shapes, and semantic. Locations and shapes are embedded in the structure affinity (SA) that is a key part of the Corner Affinity, in which we encode the shape (*e.g.,* width and height) knowledge of each instance in its corner location via a strong supervision manner. SA only embedded low-level construction similarity of corners is not enough to accomplish grouping in some extreme scenarios, such as two objects with similar shapes coincide. To this end, we devise the contexts affinity (CA) part for Corner Affinity, in which we utilize the pull-push [Newell *et al.*, 2017; Law and Deng, 2018] algorithm to mine high-level semantic similarity of the corresponding corner pairs to make coincident objects distinguishable. In the CA part, the affinity value is encoded upon the contexts response of instance in a self-supervision manner with no need for a real ground-truth value. Besides, to make the two affinities better interact, we devise the Corner Affinity function, which runs as coupling CA and SA. In this way, SA and CA interplay so that even if the SA value of two corners belong to different instances is large, CA will decay it to make sure that the value of overall Corner Affinity is low, vice versa.

We select three datasets covering different detection scenarios, *i.e.,* COCO [Lin *et al.*, 2014], Citypersons [Cordts *et al.*, 2016], and UCAS-AOD [Zhu *et al.*, 2015], to verify the effectiveness of our design. Experimental results show that the proposed Corner Affinity boosts AP of 5.8%, 35.8%, and 17.2%, on the above three benchmarks upon CornerNet baseline, proving the solid effectiveness of our design. We hope the efficient Corner Affinity can attract more attention to promote the development of bionic corner-guided detector.

2 Related Work

2.1 Center-guided Detector

Center-guided detectors generally use potential center points/areas, acting as reference positives, to regress the object bounding box (*e.g.,* height and width). Classical anchor-based detector belongs in this field, which regards center as an attribute of the preset anchors. Faster R-CNN [Ren *et al.*, 2015] popularizes the center-guided anchor mechanism in its Region Proposal Network (RPN). The aim of RPN is generating a few of proposals from a set of candidate boxes that are encoded via their centers along with heights and widths. In addition, the design of the center-guided RPN makes the detector can be end-to-end trainable. Afterwards, the center-driven anchor boxes are widely used in RPN-based two-stage detectors. To further explore the efficiency of models, some center-guided one-stage detectors also appeared. They remove the RPN and directly do regression and classification at the centers of anchor boxes. For example, SSD [Liu *et al.*, 2016] improves the performance via densely placing center anchors in multiple layers.

Unlike the center-guided anchor-based detectors [Redmon and Farhadi, 2017; Liu *et al.*, 2016; Lin *et al.*, 2017b], FCOS [Tian *et al.*, 2019] proposes a center-guided anchor-free manner. It treats lots of pixels in center areas of bounding boxes as positive samples and directly regresses four vectors (the distances from each pixel to the corresponding box's borders) without anchor guidance. Instead of four vectors, Reppoints [Yang *et al.*, 2019] predicts a set of representative points in the center area (positive pixels).

Despite the center-guided mechanism's great success, it is actually difficult to pinpoint the center of a box. This is because a center requires to be determined via all four boundaries of the object, needing four degrees of freedom.

2.2 Corner-guided Detector

Corner-guided detectors usually predict corners via outputting heatmaps, which we argue is more bionic in the generation of the object bounding box. Intuitively, when we want to obtain high-quality bounding boxes manually, each box is labeled from the top-left corner to the bottom-right one and we rarely label the object box by a center point along with height and width. A corner only needs two boundaries (degrees of freedom) to be determined.

CornerNet [Law and Deng, 2018] detects objects by predicting and grouping pairs of corner points. The grouping method in CornerNet is that if a top-left corner and a bottom-right corner belong to the same object, the distance between their embedding vectors will be small. To further optimize this grouping strategy, CenterNet [Duan *et al.*, 2019] adds a prediction branch of center points based on corners pairs estimation, making the corners matching become triplets matching. CentripetalNet [Dong *et al.*, 2020] proposes a new centripetal grouping approach to match paired corner points and achieves state-of-the-art performance.

To sum up, corner grouping is meaningful yet challenge for this kind of detector. Our Corner Affinity aims to produce more robust corner pairs to further advance the development of such human-like corner-guided detector.

3 Method

As shown in Figure 2, the execution of Corner Affinity requires an output of the corresponding corner affinity map. Each (top-left or bottom-right) map is composed of three channels, *i.e.,* two for encoding the structure affinity (SA, blue arrow-lines) and one for the contexts affinity (CA, green arrow-lines). The SA and CA interact via a devised function to be the overall Corner Affinity. In the following subsections, we will detail the SA, CA, and Corner Affinity function.

3.1 The Structure Affinity

The structure affinity (SA) is a key part of Corner Affinity, aiming to mine preliminary construction similarity of cor-

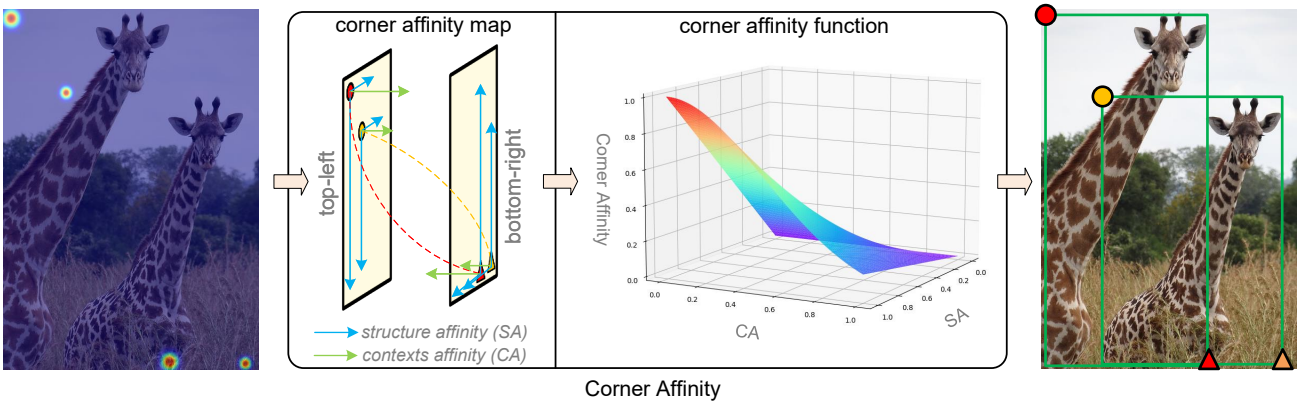

Figure 2: Diagram of the proposed Corner Affinity. Equipped with Corner Affinity, the model will output two corner affinity maps for top-left and bottom-right corner keypoints, respectively. Each corner affinity map is composed of three embedding dimensions: two for the structure affinity (SA) and one for the contexts affinity (CA). The SA and CA are coupled via designed functions to constitute the overall Corner Affinity. By the distance metric of Corner Affinity, estimated corner points can be accurately matched to be detection boxes.

ner pairs through the corresponding object's shallow structure knowledge. We define the shape and location of an instance as the structure knowledge. To obtain that, we regress the shape (*e.g.,* width and height) information of each instance at the ground-truth corner location, as shown in Figure 2 (the blue arrow-line).

Specifically, the network needs to generate two (top-left and bottom-right) 2-dimension (width and height) regression maps. Note that we utilize regression to encode SA values, which is very different from popular detectors that use regression to obtain detection boxes. We adopt the smoothL1 [Ren *et al.*, 2015] to mine the shape knowledge of each instance:

$$\mathcal{L}_{sa} = \frac{1}{N}\sum_{k=1}^{N}(\text{smoothL1}(\log(\vec{w_k}), \log(\hat{w}_k)) + \text{smoothL1}(\log(\vec{h_k}), \log(\hat{h}_k))), \tag{1}$$

where $\hat{w}_k$ and $\hat{h}_k$ are ground-truth width and height vectors, and $\vec{w_k}$ and $\vec{h_k}$ are predicted ones. N is the number of instances. The log is the logarithmic function used to constrain the length of height and width vectors to allow the learning process easier.

After obtaining the SA regression map, we can decode the predicted width ($\vec{w}$) and height ($\vec{h}$) vectors at each estimated corner location. As shown in Figure 3, the $\vec{w}$ and $\vec{h}$ can not only form a rough object box but also decode a new vector that point to the opposite corner. Accordingly, we design the structure affinity (SA) as follows:

$$\text{SA} = \frac{\overbrace{box_{tl} \cap box_{br}}^{\text{IoU}}}{box_{tl} \cup box_{br}} - \min((\underbrace{\frac{d_{tl} + d_{br}}{2D}}_{\text{corner drifting}})^2, 1), \tag{2}$$

where the box_{tl} and box_{br} represent boxes formed via the corresponding regressed $\vec{w}$ and $\vec{h}$. d means the value of corner drifting from the end of decoded vectors to target corners. And D represents the distance value of two predicted corners. More details are shown in Figure 3.

As described in Eq. 2, the SA is composed of the IoU (Intersection-over-Union) of two formed boxes and a bias named corner drifting. It is intuitive and reasonable that if different identity corners (top-left and bottom-left) belong to the same instance, their formed boxes will overlap significantly. Thus, we utilize the IoU as the basic distance metric of SA. However, vanilla IoU cannot measure offsets of the decoded vectors directly. Therefore, we propose the corner drifting as a bias of SA. The corner drifting is the mean of top-left and bottom-right drfitings, as shown in Figure 3. Based on the above design, the value range of SA is -1 to 1 and the closer the value is to 1, the higher the possibility that two corners belong to the same instance.

3.2 The Contexts Affinity

As mentioned in Section 3.1, the structure affinity only embeds low-level construction information, which is not enough to perform grouping under extreme scenarios, *e.g.,* two objects with similar shapes coincide. To this end, we introduce the contexts affinity (CA) part to mine high-level distinguishable semantic knowledge for Corner Affinity.

Inspired by CornerNet, we utilize the Associative Embedding [Newell *et al.*, 2017] method to predict an embedding value for each corner. The value is predicted based on the feature local response via a self-supervision manner, which is no need for a real ground-truth value. More specifically, we use the "pull" loss to close the embedding distance of paired corners and the "push" loss to separate the embedding distance of irrelevant corners as shown in Eq. 3. We don't care what the value of each corner is and just need minimize embedding distances of corners that belong to the same object and maximize those of different ones. Thus, each embedding without a real ground-truth mine the high-level semantic knowledge

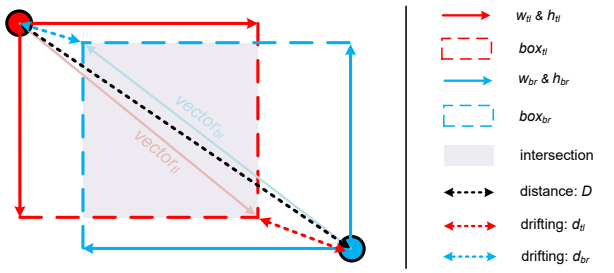

Figure 3: The structure affinity. The regressed w and h can not only form a detection box but also indicate vectors that point to the opposite corners. Based on this, we design the SA via coupling IoU and corner drifting.

of an instance.

$$\mathcal{L}_{ca} = \frac{1}{N}\sum_{k=1}^{N}\overbrace{\left[(e_{tl_k}-e_k)^2+(e_{br_k}-e_k)^2\right]}^{\text{pull loss}} + \frac{1}{N(N-1)}\sum_{k=1}^{N}\sum_{\substack{j=1\\ j\neq k}}^{N}\underbrace{\max\left(0,\Delta-|e_k-e_j|\right)}_{\text{push loss}}, \tag{3}$$

where e_k is the average of e_{tl_k} and e_{br_k}. Δ is set to be 1 following CornerNet. N is the number of objects. Similar to the SA loss, we only apply the CA loss at the ground-truth corner location.

We smooth the distance of embedding to be the contexts affinity. Suppose a top-left corner with an embedding e_{tl} and a bottom-right one with an e_{br}, we define the corresponding CA as follows:

$$\mathrm{CA} = \tanh(|e_{tl} - e_{br}|) \tag{4}$$

where the $\tanh$ function is employed to normalize the distance of embedding values. The value range of CA is 0 to 1, yet unlike SA, that the closer the CA value is to 1, the lower the possibility that two corners belong to the same object.

3.3 The Corner Affinity Function

The overall Corner Affinity is composed of the structure affinity and contexts affinity. To allow these two parts better interact, we devise corner affinity function, which runs as coupling CA and SA, as follows;

$$\mathrm{CornerAffinity} = \max(\mathrm{SA}\cdot\exp(-\frac{\mathrm{CA}^2}{\sigma}), 0) \tag{5}$$

where SA and CA are represented in Eq. 2 and Eq. 4, respectively. σ is a manually set Gaussian variance and its value is set to 0.5 empirically.

Upon the above designs, Corner Affinity encodes not only low-level structural knowledge but also high-level semantic knowledge. Even in extreme situation that two objects with similar shape overlap (the SA value of two corners belong to different instances may be large), CA will decay SA to make sure that the value of overall Corner Affinity is low, *v.v.*. In brief, only when the SA value is large and the CA value is small, the Corner Affinity value is large so that the two corners are grouped together. The hyper-surface of corner affinity function can be seen in Figure 2.

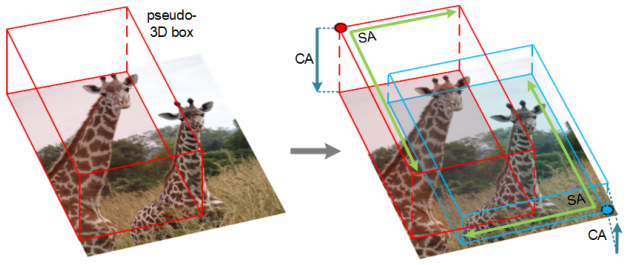

Figure 4: The qualitative analysis of Corner Affinity. We can regard the architecture of Corner Affinity as a pseudo-3D box in which the SA is the "underside" and CA represents the "depth".

For a better explanation of the Corner Affinity, we conduct qualitative analysis of it. As shown in Figure 4, we can think of the construction of Corner Affinity as a pseudo-3D box in which the SA is the "underside" and CA represents the "depth". In this way, inseparable corners with similar instance shapes or coincident centers in the 2-dimensional plane become separable in the 3-dimensional plane through the promotion of dimension (CA). Hence, the interaction of the SA and CA plays an important role in the grouping process to cover varied kinds of scenes.

4 Experiments

4.1 Datasets and Evaluation Metrics

We use three benchmark datasets to verify the effectiveness and generalization of the proposed Corner Affinity. Their instructions are as follows:

MS-COCO COCO [Lin *et al.*, 2014] is a large-scale and challenging benchmark in object detection, which contains 80 categories and more than 1.5 million object instances. We train on the train2017 split, which contains $118k$ images and $860k$ instances. We compare the proposed Corner Affinity with state-of-the-art methods on both the val and test-dev splits. All ablation studies are performed on the val2017 set that contains $5k$ images and $36k$ objects.

Citypersons We select this dataset [Cordts *et al.*, 2016] to test the effectiveness of the proposed Corner Affinity under dense occlusion scenario. Citypersons contains six different labels, *i.e.,* ignore regions, pedestrians, riders, sitting persons, other persons with unusual postures, and group of people. We keep and merge the labels of pedestrians and riders that accounts a large proportion in vanilla data. There are 18204 persons in 2471 images on our processed training set. We show the performance of the proposed Corner Affinity on the validation set that contains 439 images and 3666 persons.

UCAS-AOD There are 7482 aircraft in 1000 images in UCAS-AOD [Zhu *et al.*, 2015]. This dataset exits an extreme scenario that numerous similar objects arranged evenly and regularly. Thus, we use it to further verify the robust of our Corner Affinity. The annotations we use is the cleaned and refined version offered by [Wei *et al.*, 2020].

Method (test-dev set)	Backbone	Input size	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Center-guided(anchor-based):								
Faster R-CNN [Ren *et al.*, 2015] w/ FPN	ResNet-101	1000 × 600	36.2	59.1	39.0	18.2	39.0	48.2
RetinaNet w/ FPN [Lin *et al.*, 2017b]	ResNet-101	1333 × 800	40.8	61.1	44.1	24.1	44.2	51.2
Cascade R-CNN [Cai and Vasconcelos, 2018]	ResNet-101	1333 × 800	42.8	62.1	46.3	23.7	45.5	55.2
YOLOv4 [Bochkovskiy *et al.*, 2020]	CSPDarkNet-53	608 × 608	43.5	65.7	47.3	26.7	46.7	53.3
Center-guided(anchor-free):								
CenterNet [Zhou *et al.*, 2019]	Hourglass-104	512 × 512	42.1	61.1	45.9	24.1	45.5	52.8
FCOS [Tian *et al.*, 2019]	ResNeXt-101	1333 × 800	42.1	62.1	45.2	25.6	44.9	52.0
Reppoints [Yang *et al.*, 2019]	ResNet-101	1333 × 800	45.0	66.1	49.0	26.6	48.6	57.5
Corner-guided:								
CenterNet [Duan *et al.*, 2019]	Hourglass-104	511 × 511	44.9	62.4	48.1	25.6	47.4	57.4
CentripetalNet [Dong *et al.*, 2020]	Hourglass-104	511 × 511	45.8	63.0	49.3	25.0	48.2	58.7
CPN (two-stage) [Duan *et al.*, 2020]	Hourglass-104	511 × 511	47.0	65.0	51.0	26.5	50.2	60.7
CornerNet baseline [Law and Deng, 2018]	Hourglass-104	511 × 511	40.5	56.5	43.1	19.4	42.7	53.9
w/ Corner Affinity	Hourglass-104	511 × 511	46.3	64.0	49.9	27.4	49.3	58.7
Improvement	-	-	↑ 5.8	↑ 7.5	↑ 6.8	↑ 8.0	↑ 6.6	↑ 4.8
Method (val set)	Backbone	Feature	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DETR [Carion *et al.*, 2020]	ResNet-50	Encoder	43.3	63.1	45.9	22.5	47.3	61.1
DETR [Carion *et al.*, 2020]	ResNet-101	Encoder	44.9	64.7	47.7	23.7	49.5	62.3
Sparse R-CNN [Sun *et al.*, 2021]	ResNet-50	FPN	42.8	61.2	45.7	26.7	44.6	57.6
Sparse R-CNN [Sun *et al.*, 2021]	ResNet-101	FPN	44.1	62.1	47.2	26.1	46.3	59.7
w/ Corner Affinity	Hourglass-52	Single	43.2	60.8	46.1	25.1	46.8	57.7
w/ Corner Affinity	Hourglass-104	Single	45.1	62.9	48.3	26.7	48.5	59.8

Table 1: Comparisons with different object detectors on COCO both test-dev and val set. With our Corner Affinity, CornerNet achieves an AP of 46.3%, yielding very competitive accuracy among reported advanced detectors. For COCO val set, with the proposed Corner Affinity, CornerNet (in single feature without feature enhancement) yields an AP of 45.1%, proving to be a superb baseline again.

Evaluation Metrics We use the AP (average precision) metric to measure performance of object detection. The AP used in COCO is computed over ten different IoU thresholds (*i.e.,* 0.5:0.05:0.95) and all 80 categories. For Citypersons and UCAS-AOD, since the annotation (bounding box) is not as precise as COCO, the AP under a high IoU is meaningless, so we only test the AP with 0.5 IoU.

4.2 Implementations Details

Training details During training, we set the input resolution to 511 × 511, which yields an output resolution of 128 × 128 for all models. For COCO, we train our model on eight NVIDIA V100 GPUs with 8 × 32 GB RAM. The model is trained from scratch to $600k$ iterations with a batch size of 64. The learning rate is set to $2.5e$-4 and dropped $10\times$ at the $500k$ iteration. For Citypersons and UCAS-AOD, we use two NVIDIA P100 GPUs with 2×16 GB RAM. All models are trained from scratch to $50k$ iterations with a learning rate of $1e$-4 enjoying a batch size of 12. Standard random color jittering, cropping, and mirror flipping are performed as data augmentation. We use Adam as the optimizer for the full training objective. Other settings are the same as CornerNet baseline.

Testing details We follow CornerNet to extract each corner coordinate. Firstly, we use softmax and 3×3 max-pooling on heatmaps and remain the top100 scored top-left and bottom-right corners. Then we use the predicted offsets to refine the locations of corners. Next, we decode the structure affinity and contexts affinity in the corresponding corner location and combine them to be the overall Corner Affinity. Finally, Corner Affinity is performed to group corners that belong to the same object instance and Soft-NMS is run to filter the detection results. In this stage, each image is is fed into the network with its original resolution.

4.3 Main Result

COCO We utilize the challenging COCO dataset to verify the effectiveness of our Corner Affinity. As shown in Table 1, the Corner Affinity brings 5.8% boosting, from 40.5% to 46.3%, on AP for CornerNet baseline on COCO test-dev set without bells and whistles. And only updated with the proposed new grouping algorithm, CornerNet surpasses popular detection baseline [Cai and Vasconcelos, 2018; Bochkovskiy *et al.*, 2020; Tian *et al.*, 2019; Yang *et al.*, 2019], with a large margin, proving the corner-guided detector exists high ceiling and our Corner Affinity remarkably promotes the development of this detector via optimizing corner grouping. It is worth noting that our pure grouping optimization produces more improvements for vanilla CornerNet than the other two single-stage variants, *i.e.,* CenterNet [Duan *et al.*, 2019] and CentripetalNet [Dong *et al.*, 2020] and is competitive to the two-stage variant CPN [Duan *et al.*, 2020]. The single-stage two not only optimize corner grouping upon CornerNet, but also use stronger corner enhanced features yet still gain weaker accuracy than our pure grouping optimization. The two-stage CPN, without grouping process, introduces sub-networks with numerous parameters to classify proposals is

Method	Backbone	Epoch	AP_c	AP_u
Faster R-CNN	ResNet-101	36	25.0	90.1
RetinaNet	ResNet-101	36	27.9	87.3
CenterNet	Hourglass-104	50	51.5	86.8
CentripetalNet	Hourglass-104	50	54.7	-
CornerNet baseline	Hourglass-104	50	29.1	79.1
w/ Corner Affinity	Hourglass-104	50	64.9	96.3
Improvement	-	-	↑ 35.8	↑ 17.2

Table 2: Comparisons with different detectors on Citypersons and UCAS-AOD. AP_c and AP_u represent the AP obtained on Citypersons and UCAS-AOD, respectively.

not as neat as our design. The experimental results show that the proposed Corner Affinity is the optimal corner grouping strategy at present.

We also produce comparisons with the newest favorite Transformer-based approaches. As shown in Table 1, employing Corner Affinity, CornerNet (Hourglass-104) yields an AP of 45.1% and an AP_{75} of 48.3% on COCO val set, surmounting recent advanced self-attention-based single-stage detectors (*e.g.,* DETR [Carion *et al.*, 2020] and Sparse R-CNN [Sun *et al.*, 2021]), even without any feature enhanced module, *e.g.,* FPN [Lin *et al.*, 2017a]. Higher AP_{75} means higher-quality detection boxes. All the above experimental results demonstrate the proposed new grouping algorithm, Corner Affinity, is splendid, which allows the antiquated corner-guided detector, CornerNet, to be a superb baseline again even in the Transformer-based detection era.

Citypersons & UCAS-AOD We select these two datasets to test the generalization performance of Corner Affinity under two extreme scenarios, *i.e.,* occlusion scene (Citypersons) and symmetrical arrangement of similar objects scene (UCAS-AOD). As shown in Table 2, Corner Affinity can still produce excellent accuracy upon higher-quality corner pairs under all aforementioned challenging situations. Specifically, compared with vanilla CornerNet, Corner Affinity boosts AP of amazing 35.8% and 17.2% on Citypersons and UCAS-AOD, respectively, firmly proving our design brings more robust grouping capability for corner-guided detectors.

4.4 Ablation Study

In this section, we conduct ablation analyses mainly on COCO val2017 set and partially on Citypersons.

Effectiveness of each part of Corner Affinity The proposed Corner Affinity is coupled of the structure affinity (SA) and contexts affinity (CA). SA is utilized to generate primary corner pairs via construction information of objects and CA runs as amending corner pairs via deeper semantic features. Each part can work individually, and thus we mask one to test the usefulness of the other. As shown in Table 3, for COCO datasets, only with the SA, a model can yield a decent AP of 44.3%. When only employing CA, the model only obtain an AP of 40.2%. Then when we couple the SA and CA, the AP reaches highest 45.1%. One may argue the AP of overall Corner Affinity is only 0.8% higher than that of pure SA. It is because there are too few cases of occlusion in the COCO dataset. To better show the superiority of Corner Affinity,

Corner Affinity		**COCO**			**Citypersons**
SA	CA	AP	AP_{50}	AP_{75}	AP_c
✓		44.3	62.1	47.4	56.8
	✓	40.2	58.1	43.5	28.4
✓	✓	45.1	62.9	48.3	64.9

Table 3: Effectiveness of each part of Corner Affinity. We disassemble the Corner Affinity into SA and CA to verify the effectiveness of each part on both COCO and Citypersons datasets.

The structure affinity		**COCO**			**Citypersons**
IoU	corner drifting	AP	AP_{50}	AP_{75}	AP_c
✓		44.7	62.6	47.8	63.6
✓	✓	45.1	62.9	48.3	64.9

Table 4: Effectiveness of corner drifting. We verify the effectiveness of corner drifting in the structure affinity.

Threshold	0.05	0.1	0.15	0.2	0.25	0.3
AP	44.8	45.1	45.1	44.9	44.6	44.5

Table 5: Different threshold values of Corner Affinity. We perform ablation experiments (on COCO) to find the optimal threshold value for Corner Affinity to execute grouping process.

we also do experiments on Citypersons dataset. For this data, Corner Affnity surpass 8.1% AP than that only with pure SA, showing the proposed Corner Affinity can cover more scenarios and is more reasonable.

Effectiveness of corner drifting The structure affinity (SA) is composed of an IoU and a bias, *i.e.,* corner drifting. We devise the corner drifting to directly measure the distance of regressed vectors and true predicted corners. As shown in Table 4, the proposed corner drifting lifts 0.4% and 1.3% AP on COCO and Citypersons, respectively. The experimental results prove the designed corner drifting is resultful.

Optimal threshold for Corner Affinity Corner Affinity needs a threshold to be applied to group corners. Accordingly, we conduct ablation experiments to search the best threshold value. According to the Table 5, 0.1 is a suitable choice. Therefore, we set the value of Corner Affinity threshold to 0.1 for all experiments.

5 Conclusion

In this paper, we propose a robust grouping algorithm, termed as Corner Affinity, to produce more reliable and precise corner pairs for the bionic corner-guided detector. Equipped with the proposed Corner Affinity, the antiquated CornerNet baseline becomes great again. Along with Corner Affinity, we provide a new grouping idea, *i.e.,* lifting the dimension of output attributes for each corner to cope extreme scenarios. We believe the human-like corner-guided detector still enjoys potentials and we hope our Corner Affinity can attract more attention to the corner-guided detection manner in the era dominated by the center-guided methods.

References

[Bochkovskiy *et al.*, 2020] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[Cai and Vasconcelos, 2018] Zhaowei Cai and Nuno 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.

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[Cordts *et al.*, 2016] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt 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.

[Dong *et al.*, 2020] Zhiwei Dong, Guoxuan Li, Yue Liao, Fei Wang, Pengju Ren, and Chen 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.

[Duan *et al.*, 2019] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6569–6578, 2019.

[Duan *et al.*, 2020] Kaiwen Duan, Lingxi Xie, Honggang Qi, Song Bai, Qingming Huang, and Qi Tian. Corner proposal network for anchor-free, two-stage object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 399–416. Springer, 2020.

[Law and Deng, 2018] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[Lin *et al.*, 2017a] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[Lin *et al.*, 2017b] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[Liu *et al.*, 2016] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[Liu *et al.*, 2021] Fanfan Liu, Haoran Wei, Wenzhe Zhao, Guozhen Li, Jingquan Peng, and Zihao Li. Wb-detr: Transformer-based detector without backbone. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2979–2987, 2021.

[Newell *et al.*, 2017] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in neural information processing systems*, pages 2277–2287, 2017.

[Redmon and Farhadi, 2017] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[Sun *et al.*, 2021] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu 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.

[Tian *et al.*, 2019] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 9627–9636, 2019.

[Wei *et al.*, 2020] Haoran Wei, Yue Zhang, Bing Wang, Yang Yang, Hao Li, and Hongqi Wang. X-linenet: Detecting aircraft in remote sensing images by a pair of intersecting line segments. *IEEE Transactions on Geoscience and Remote Sensing*, 59(2):1645–1659, 2020.

[Yang *et al.*, 2019] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9657–9666, 2019.

[Zhou *et al.*, 2019] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[Zhu *et al.*, 2015] Haigang Zhu, Xiaogang Chen, Weiqun Dai, Kun Fu, Qixiang Ye, and Jianbin 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.