Research Article

INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced Robotic Systems
January-February 2021: 1–11

DOI: 10.1177/1729881421993323
journals.sagepub.com/home/arx
SAGE

Geometric property-based convolutional neural network for indoor object detection

Xintao Ding[1,2], Boquan Li[2,3] and Jinbao Wang[1,2]

Abstract

Indoor object detection is a very demanding and important task for robot applications. Object knowledge, such as two-dimensional (2D) shape and depth information, may be helpful for detection. In this article, we focus on region-based convolutional neural network (CNN) detector and propose a geometric property-based Faster R-CNN method (GP-Faster) for indoor object detection. GP-Faster incorporates geometric property in Faster R-CNN to improve the detection performance. In detail, we first use mesh grids that are the intersections of direct and inverse proportion functions to generate appropriate anchors for indoor objects. After the anchors are regressed to the regions of interest produced by a region proposal network (RPN-RoIs), we then use 2D geometric constraints to refine the RPN-RoIs, in which the 2D constraint of every classification is a convex hull region enclosing the width and height coordinates of the ground-truth boxes on the training set. Comparison experiments are implemented on two indoor datasets SUN2012 and NYUv2. Since the depth information is available in NYUv2, we involve depth constraints in GP-Faster and propose 3D geometric property-based Faster R-CNN (DGP-Faster) on NYUv2. The experimental results show that both GP-Faster and DGP-Faster increase the performance of the mean average precision.

Keywords

Indoor object detection, robot application, geometric constraint, CNN

Date received: 27 February 2020; accepted: 20 January 2021

Topic Area: Vision Systems
Topic Editor: Antonio Fernandez-Caballero
Associate Editor: Loredana Zollo

Introduction

Indoor object detection is a very demanding and important task for robot applications. Generally, object detection contains two main tasks: the localization and classification problems.[1] Object detection is not an easy task due to the uncertainty of the location of the interest object. In this work, we focus on the indoor object detection.

Our work is motivated by two questions on the robot application. First, is the geometric property helpful for mobile robot to detect indoor object? Second, if the first answer is positive, how can the geometric property be used to detect indoor object? Since the depth information is not always available, we employ the shape of the bounding box as a universal geometric property to improve the performance of the indoor object detection.

In the last two decades, object detectors based on convolutional neural networks (CNNs)[2–6] have achieved state-of-the-art results on various challenging benchmarks.[7,8] As a representative region-based CNN detector,

[1] School of Computer and Information, Anhui Normal University, Wuhu, China
[2] Anhui Province Key Laboratory of Network and Information Security, Wuhu, China
[3] School of Mathematics and Statistics, Anhui Normal University, Wuhu, China

Corresponding author:
Xintao Ding, School of Computer and Information, Anhui Normal University, Wuhu 241003, China.
Email: xintaoding@163.com

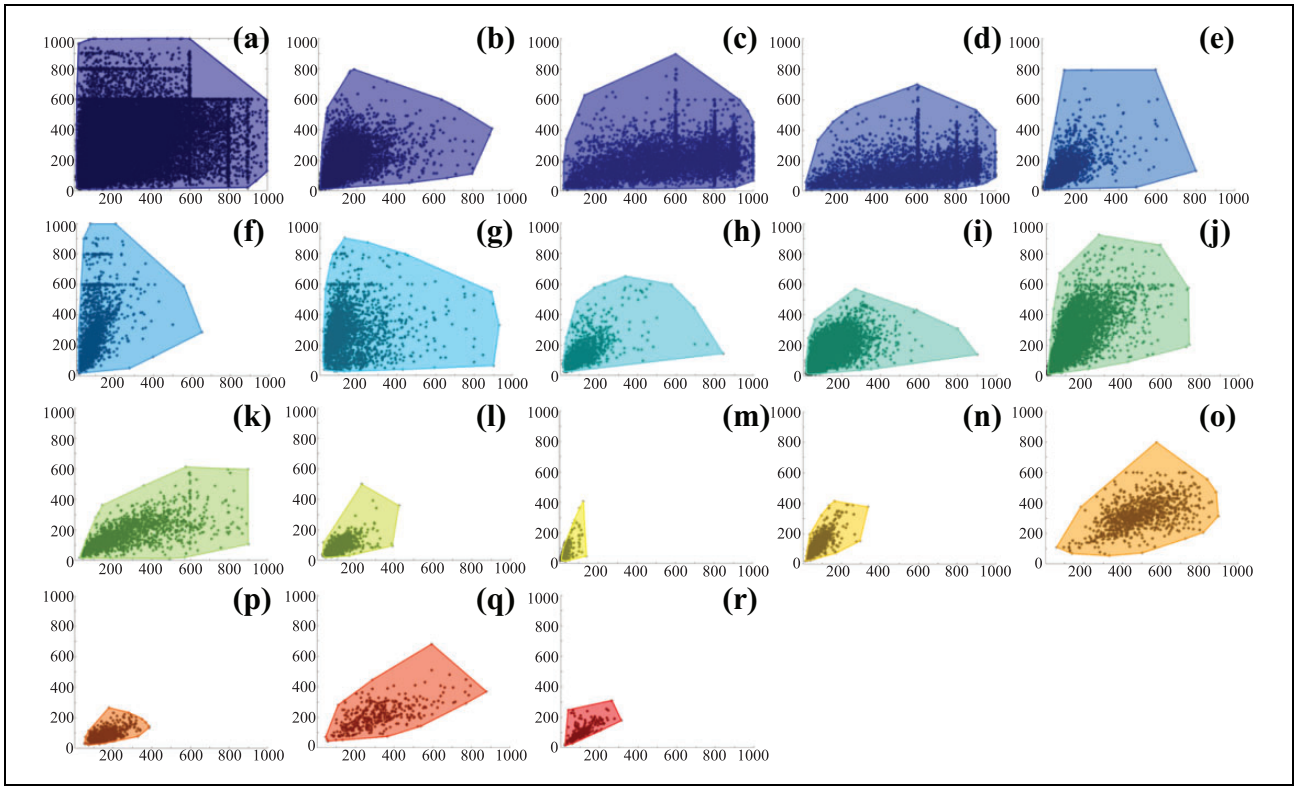

Figure 1. The 2D geometric constraints of the 18 indoor classes of SUN2012. The black points show the width and height coordinates of the annotated bounding boxes. (a) wall, (b) window, (c) floor, (d) ceiling, (e) plant, (f) door, (g) curtain, (h) painting, (i) chair, (j) person, (k) table, (l) cushion, (m) bottle, (n) desk lamp, (o) bed, (p) pillow, (q) sofa, and (r) television.

Faster R-CNN[4] uses a region proposal network (RPN) to generate proposals. The regions of interest (RoIs) produced by RPN (RPN-RoIs) are chosen to train the proposals if (1) an RPN-RoI overlaps a ground-truth box with a highest intersection-over-union (IoU) overlap and (2) its IoU overlap is greater than a threshold. Every selected RPN-RoI is assigned a training label, which is the ground-truth label with the highest IoU overlap. Although the selection strategy of RPN-RoI is efficient, it does not focus on the geometric property of the candidates. The knowledge of the indoor object, such as geometric shape and context, may be helpful for detection. In this study, we use the shape of the bounding box as a two-dimensional (2D) geometric property to improve Faster R-CNN for the indoor object detection.

We first run over the indoor dataset to result in the widths and heights of the annotated bounding boxes. Then, we put them in the first quadrant of the Cartesian plane with their left-bottom points at the origin. The 2D geometric constraint of every classification is a convex hull region enclosing corresponding right-upper points. Figure 1 shows the 2D constraints of the 18 indoor classes collected from SUN2012 (http://groups.csail.mit.edu/vision/SUN/) database.[9] The black points show the coordinates composed by the widths and heights of the annotated bounding boxes.

From Figure 1, it can be seen that the scales and aspect ratios on the indoor classes vary in a large range. The "cushion," "bottle," "desk lamp," "pillow," and "television" are small objects. The "door" and "bottle" are thin objects with large aspect ratios, while "ceiling" and "table" are thick objects with small aspect ratios.

In this study, we propose 2D geometric property-based Faster R-CNN method (GP-Faster) for indoor object detection. For indoor applications, small objects, such as "bottle," "pillow," and "television", are common targets in indoor scene. Because the coverage of the anchors generated from standard Faster R-CNN cannot fit the size of the small indoor objects, we first use mesh grids to generate appropriate anchors for small indoor objects, in which the grids are the intersections of direct and inverse proportion functions. After the anchors are regressed to RPN-RoIs, we then use the 2D constraints to refine the RPN-RoIs, that is, the width and height coordinates of the RPN-RoIs that fall in their 2D constraints are employed as inliers. The 2D constraints may remove outliers in RPN-RoIs. Extensive experiments implemented on SUN2012[9] and NYUv2 (https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html)[10] demonstrate that GP-Faster is able to improve detection performance for indoor object. The geometric property is helpful for region proposal.

Our main contributions are summarized as follows:

1. We use mesh grids to generate anchors with the help of direct and inverse proportion functions.
2. We use the shape of the bounding box as a universal geometric property to improve region proposals for classification.
3. We use geometric constraints to remove the RPN-RoIs that may produce negative predictions.
4. Compared with Faster R-CNN, our GP-Faster approach increases the performance of the mean average precision (mAP).

Related work

There are an increasing number of recent studies that focus on CNN-based indoor object detection. Gopan and Aarthi designed a CNN to identify bottles in the indoor environment.[11] Mordan et al. designed a context-based residual auxiliary block to combine ResNet and single-shot multibox detector for indoor object detection.[12] Zheng et al. combined CNN and recurrent neural network (RNN) to implement indoor semantic segmentation.[13] Ammirato et al.[14] first extracted CNN features from the target and scene images, and then, fed the features to a target-driven instance detector for object detection. Ehsan and Nahvi detected indoor people violence based on a combination of motion trajectory and spatiotemporal features.[15] Zhou et al. proposed a multimodal fusion deep CNN framework for object detection and segmentation.[16] The CNN-based indoor detectors usually pay much attention to the architectures of their networks and do not focus on application scenario. For indoor applications, the property of the indoor object may be helpful for detection.

Because mobile robot usually needs to detect indoor object for their navigation and service, many kinds of literature focus on vision-based object detection for robot. Reyes et al. proposed a CNN method based on You Only Look Once[5] to detect object for pepper.[17] Zhu et al. proposed a CNN-based indoor landmark detector with the help of a topological matching algorithm.[18] Together with classical classifier, Jiang et al. proposed a CNN-based tracking method for person-following robot.[19] Loghmani et al. proposed a two-stream fusion method for robot vision,[20] in which the features of the two CNN streams of RGB and depth images are fed into an RNN to detect objects. Sampedro et al. proposed a fully autonomous aerial robotic solution for search and rescue missions in indoor environments.[21] After employing Mask-RCNN[6] for image segmentation, Kowalewski et al. presented a full solution that produces object-level semantic perception of the environment for indoor mobile robot.[22] Although the vision-based detectors show advances in robot vision, many of them tend to assemble techniques.

Because geometric property is helpful for object understanding, geometric property is used in both traditional method[23–25] and CNN-based method.[26–28] Wu and Wang employed geometric property to detect elliptical object.[23] Batool and Chellappa applied geometric constraints to localize curvilinear shapes for wrinkles detection.[24] Ismail et al. estimated the indoor spatial layout using the vanishing point and then detected the object by studying the relation of the scene to the object.[25] Pham et al. first predicted a boundary map using a CNN and then employed a hierarchical segmentation tree to produce geometric and object segmentation.[26] Mizginov and Danilov combined generative adversarial networks and three-dimensional (3D) geometric modeling to detect traffic target.[27] Cai et al. employed geometric prior knowledge to improve a CNN-based method for planar object detection.[28] Since a kind of road object (e.g. car or bus) is usually in a standard size and the surveillance camera is static, the scale distribution of the class in the video frames can be estimated after the horizon is estimated by scene geometry. Amin and Galasso applied the scale distributions over the road classes to prune proposals.[29] However, the method is not appropriate for indoor objects due to the nonuniqueness of the scale distribution in indoor scene. The indoor robot moves in room and the indoor objects, such as wall, floor, or ceiling, may be not in standard size. A same object may be occurred in an image at the same position but in different pixel sizes. Although geometric properties are used for object detection, the CNN-based study of geometric property for indoor object detection is insufficient. Motivated by the application of the robot vision, we use geometric property to improve CNN-based detector in this study.

Proposed method

In our design, the main task of GP-Faster is to use 2D geometric property to improve region proposals. The main design of GP-Faster is shown in Figure 2. The blue modules show the loss of training. The red modules show our improvements. The FG prob in Figure 2 is the abbreviation of foreground probability, which is reshaped from a softmax layer. With the help of direct and inverse proportion functions, we first use mesh grids to generate appropriate anchors for indoor object location in the module of anchors generation. With the help of geometric prior knowledge, we then incorporate 2D geometric constraints in Faster R-CNN to train proposals for classification, as the module of geometric constraint shown in Figure 2.

Generating appropriate anchors

Faster R-CNN produces anchors to regress the bounding boxes of objects. Each anchor is generated with a scale s and an aspect ratio r, where $s^2 = wh$ is the size of the anchor and $r = h/w$ is the ratio of the height to the width of the anchor. Let $s_{an} = (s_1, s_2, \cdots, s_m)$ be a group of scales and $r_{an} = (r_1, r_2, \cdots, r_n)$ be a group of aspect ratios. The anchors are generated by all the combinations of $r_i \in r_{an}$ and $s_j \in s_{an}$, that is, $(r_i, s_j), i = 1, 2, \cdots, n; j = 1, 2, \cdots, m$. The main design of the anchors is to choose appropriate s_{an}

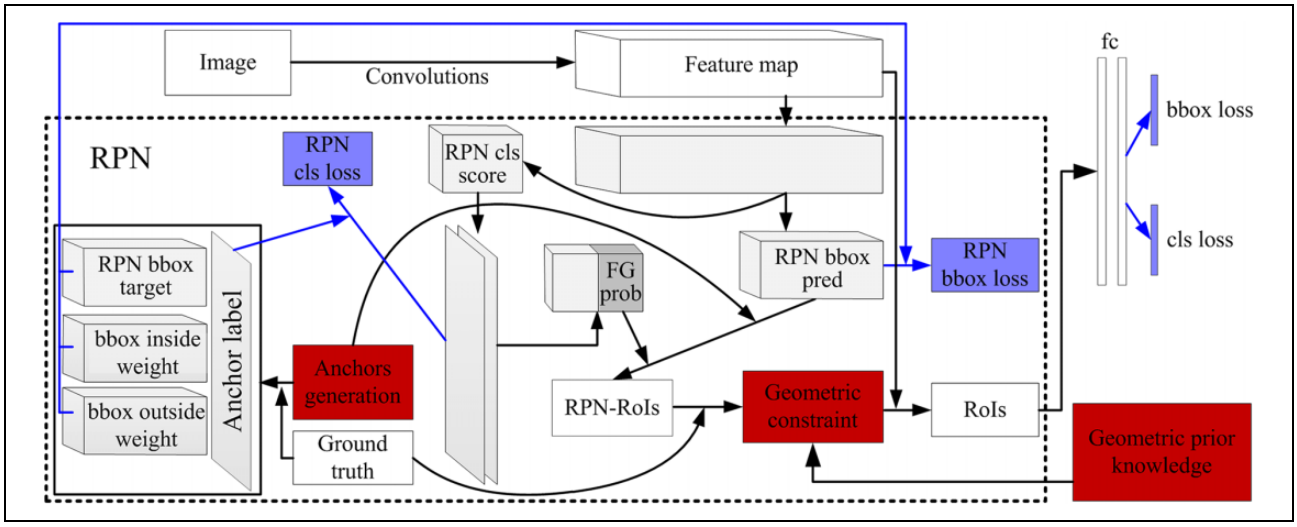

Figure 2. The training framework of our proposed GP-Faster. The red modules show our improvements. GP-Faster: geometric property-based Faster R-CNN method.

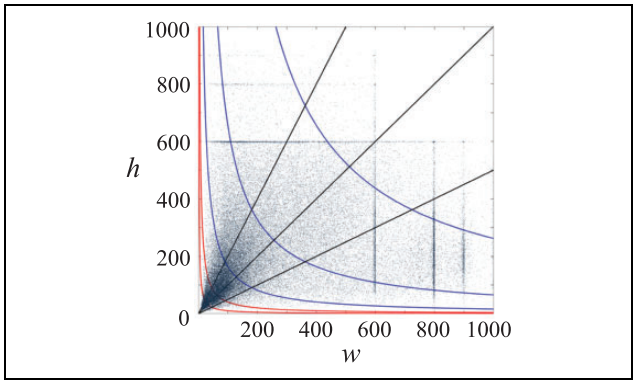

Figure 3. The generation schedule of the anchors. The points show the widths and heights of the ground-truth boxes. The black lines show the aspect ratios of the anchors. The curves show the sizes of the anchors.

and r_{an} so that the generated anchors can be easily regressed to ground boxes. In this study, we use mesh grids that are the intersections of direct and inverse proportion functions to generate anchors.

Figure 3 shows the generation schedule of our design. The points show the widths and heights of the ground-truth boxes of the 18 indoor classes in SUN2012, in which the images are rescaled such that their shorter side is 600 and the other side is no more than 1000. The constraints of aspect ratios can be regarded as directly proportional functions as follows

$$h = r_i w, i = 1, 2, \cdots, n. \tag{1}$$

They are shown as the black lines in Figure 3. Similarly, the size constraints of the anchors can be regarded as inverse proportion functions

$$h = \frac{s_j^2}{w}, j = 1, 2, \cdots, m. \tag{2}$$

The curves in Figure 3 show the size constraints. The width and height of every anchor are determined by equations (1) and (2) with a certain combination $(r_i, s_j), i = 1, 2, \cdots, n; j = 1, 2, \cdots, m$, that is, for a given scale and aspect ratio, the width and height of the generated anchor are the solution of the following equations

$$\begin{cases} h = r_i w \\ h = \dfrac{s_j^2}{w} \end{cases}. \tag{3}$$

The anchors on each feature point are the intersections of the direct and inverse proportion functions, as the intersection of the lines and curves shown in Figure 3.

Different from the common dataset, there are a considerable number of small objects in the indoor dataset. As shown in Figure 3, there are a large number of black points near origin. Although Faster R-CNN uses bounding-box regression to adjust the error from an anchor box to a ground-truth box, the regression may be powerless when the error between them is too large. An anchor that overlaps a ground-truth box with a high IoU overlap is employed to train the regression. In other words, an object cannot be involved in training if there is no anchor overlapping it with a high IoU overlap. In this study, we design inverse proportion curves near origin to trap the small ground boxes, as the red curves shown in Figure 3.

GP-Faster designs the scales of the anchors as follows

$$\begin{aligned} & s_{an} = (s_1, s_2, \cdots, s_m) \\ & \text{s.t.} \begin{cases} s_1 < s_2 < \cdots < s_m \\ b_s s_1 \geq c_s \\ b_s s_m \leq d_{\min}/2 \end{cases} \end{aligned} \tag{4}$$

where $b_s = 8$ is the base size, c_s is the zoom scale from the input image to the last shared convolutional layer, and $d_{\min}$ is the length of the shorter side of the input image.

2D geometric constraints

To improve the inliers ratio on RPN-RoIs, we use 2D geometric constraints to refine RPN-RoIs, as shown in Figures 1 and 2. We first prepare 2D geometric prior knowledge. For convenience, let the coordinate of the j'th annotated object in the i'th training image I_i be $p_{ij} = (w_{ij}, h_{ij})$, which is the width and height of the annotated box. Let the class of p_{ij} be $gc(p_{ij})$, and $P_k = \{p_{ij}|gc(p_{ij}) = k\}$ be the set of p_{ij} with $gc(p_{ij}) = k, k = 0, 1, \cdots, K-1$ on the training set, and K is the number of classes. We use Graham scan to obtain the convex hull of P_k.[30] Let the resulting boundary of the convex hull be $B_k \subset P_k$, which is a list of convex polygon vertexes. The geometric prior knowledge, which is the boundary of the convex hull of all the classes, is prepared as follows:

1. Run over the training set to obtain the point sets $P_k = \{p_{ij}|gc(p_{ij}) = k\}, k = 0, 1, \cdots, K-1$.
2. Obtain $B_k, k = 0, 1, \cdots, K-1$ using Graham scan.

We then use the prepared prior knowledge to result in 2D constraint. Faster R-CNN uses anchors for bounding box regression. Not only the anchors inside of the region of B_k but also the anchors near outside of the region of B_k may be used for box regression. Let the current training image be I_i, the anchors set of the resulting RPN-RoIs be $A_{\text{RPN}} = \{a_t, t = 0, 1, \cdots, N_{\text{RR}} - 1\}$, correspondingly, the coordinates of the anchors be $R_{\text{RPN}} = \{r_t\} = \{(w_t, h_t), t = 0, 1, \cdots, N_{\text{RR}} - 1\}$. We use the ray-tracing method[30] to check a point $r_t \in R_{\text{RPN}}$ is inside or outside of the polygon B_k. Let the region inside of B_k be $R(B_k)$, $d(r_t, B_k)$ be the distance from r_t to B_k. If r_t is in $R(B_k)$, then let $d(r_t, B_k) = 0$; otherwise, the distance is the minimum distance from r_t to l_p, which is the line segment of B_k, that is

$$d(r_t, B_k) = \begin{cases} \min\limits_{l_p \in B_k} d(r_t, l_p), & r_t \notin R(B_k) \\ 0, & r_t \in R(B_k) \end{cases}. \quad (5)$$

The 2D constraint of $r_t \in R_{\text{RPN}}$ is as follows

$$d(r_t, B_k) \le T_g s(B_k), k = 0, 1, \cdots, K-1, \quad (6)$$

where T_g is a threshold, $s(B_k)$ is the region area bounded in B_k.

To implement the refinement of RPN-RoIs, let the number of RoIs we choose in RPN-RoIs be N_{RoI}. Our main task is to select N_{RoI} RoIs in R_{RPN} using the 2D constraints. The involved refinement parameters of GP-Faster are R_{RPN} and thresholds: T_g, N_{RoI}, T_{IoU}, T_{BGL}, and T_{BGH}, where T_{IoU} is a threshold of IoU overlap; T_{BGL} and T_{BGH} are, respectively, the lower and upper boundary thresholds used for background selection. The set of RoIs for classification contains two parts, that is, foreground F_{RoI} and background B_{RoI}. The refinement of RPN-RoIs using 2D geometric constraints is implemented as follows:

1. Initialize F_{RoI} and B_{RoI} with $\varnothing$.
2. For $a_t \in A_{\text{RPN}}$ induced from the current training image I_i, obtain $\text{maxIoU}(a_t, b_{ih})$, where b_{ih} is the bounding box of the h'th annotated object in I_i.
3. Add a_t to F_{RoI}, that is, $F_{\text{RoI}} = F_{\text{RoI}} \cup \{a_t\}$, if a_t satisfies
   (a) $\text{maxIoU}(a_t, b_{ih}) \ge T_{\text{IoU}}$,
   (b) the 2D constraint equation (6), where $d(r_t, B_k)$ is the distance from r_t to B_k, $k = gc(p_{ij})$, $j = \arg\max\limits_h \text{maxIoU}(a_t, b_{ih})$.
4. Add a_t to B_{RoI}, that is, $B_{\text{RoI}} = B_{\text{RoI}} \cup \{a_t\}$, if a_t satisfies $T_{\text{BGL}} \le \text{maxIoU}(a_t, b_{ih}) < T_{\text{BGH}}$.
5. Repeat steps 2 to 4 until all the a_t have been processed.
6. Update F_{RoI} by randomly choosing N_F elements in F_{RoI}, where $N_F = \min(f_{\text{RoI}} N_{\text{RoI}}, \text{Card}(F_{\text{RoI}}))$, f_{RoI} is a fixed ratio of the foreground in RoIs, and $\text{Card}(\bullet)$ is the cardinality of $\bullet$.
7. Update B_{RoI} by randomly choosing N_B elements in B_{RoI}, where $N_B = N_{RoI} - N_F$.
8. The set of RoIs is $RoI = F_{\text{RoI}} \cup B_{\text{RoI}}$.

The geometric constraints are only employed to train models. They are not used for model test. After the refinement of RPN-RoIs is implemented, the resulting RoIs are fed to full connections to result in training loss. Overall, GP-Faster generates appropriate anchors and implements 2D geometric refinement on the training set for improvement.

Implemental details

Besides thresholds N_{RR}, T_{IoU}, T_{BGL}, T_{BGH}, and N_{RoI}, two main parameters s_{an} and T_g are introduced in this work, where s_{an} and T_g are, respectively, the scales to generate anchors and the threshold used for 2D constraints. T_{IoU}, T_{BGL}, and T_{BGH} are, respectively, set to 0.5, 0, and 0.5, which are set the same as those in Faster R-CNN. Overall, N_{RR}, N_{RoI}, s_{an}, and T_g are the four parameters that we tune in this work.

We tune the parameters on SUN2012 using VGG16.[31] After comparing the classes of the datasets Indoor09,[32] SUN2012,[9] and NYUv2,[10] we use 18 common indoor classes for implementation. They are "wall," "window," "floor," "ceiling," "plant," "door," "curtain," "painting," "chair," "person," "table," "cushion," "bottle," "desk lamp," "bed," "pillow," "sofa," and "television" (Figure 1). The model is trained on the 18 classes of the SUN2012 training set. It is evaluated on corresponding classes of the SUN2012 test set using mAP. Because the object occluded is listed as a new class in SUN2012, the eight classes in SUN2012, including "person occluded," "person sitting

Table 1. The ablation experiments of GP-Faster16 on the SUN2012 test set.

N_{RR}	N_{RoI}	s_{an}	T_g	mAP (%)	Rec (%)	Pre (%)
256	128	(2, 4, 8, 16, 32)	10^{-4}	46.0	65.7	27.5
448	196	(2, 4, 8, 16, 32)	10^{-4}	49.8	74.0	19.1
512	256	(2, 4, 8, 16, 32)	10^{-4}	50.1	73.8	20.7
512	256	(2, 4, 8, 16, 32)	0	49.9	73.2	20.6
512	256	(2, 4, 8, 16, 32)	2×10^{-4}	50.2	73.3	20.7
512	256	(8, 16, 32)	10^{-4}	49.3	71.9	22.2
512	256	(4, 8, 16, 32)	10^{-4}	50.0	73.8	20.3

mAP: mean average precision; Pre: precision; Rec: recall.

occluded," "person," "person standing," "person walking," "person crop," "person sitting," and "person sitting crop," are fused to the class "person." Similarly, the eight classes, including "chair occluded," "chair," "chair crop," "armchair," "armchair occluded," "swivel chair," "armchair crop," and "deck chair," are fused to the class "chair."

The publicly available VGG16 model pretrained on ImageNet[2] is used for initialization. We train and test networks on images of a single scale in which the shorter side is 600 pixels. We initialize a learning rate of 0.001 and make the learning rate drop 10 times after every 80 k iterations on the dataset. A total of 100 k training iterations are run.

We run experiments to tune the training parameters N_{RR}, N_{RoI}, T_g, and s_{an} based on VGG16. Every model is tested with the same group of parameters $(Nd^{pre}_{NMS}, Nd^{post}_{NMS}) = (24\,k, 1200)$, where Nd^{pre}_{NMS} and Nd^{post}_{NMS} are, respectively, the number of top-scored RPN proposals before and after applying nonmaximum suppression (NMS) in the stage of detection. Table 1 summarizes ablation results on the four parameters. The Rec and Pre in Table 1 are, respectively, the abbreviations of the recall and precision. Together with ground truth, the numbers of true-positive and false-positive samples are used to calculate the recall and precision of every class. A predicted detection is regarded as a true positive if the predicted class label is the same as the ground-truth label and the IoU overlap between the predicted bounding box and the ground-truth one is greater than 0.5, otherwise, the detection is a false positive one. The results of Rec and Pre listed in Table 1 are the averages of the recalls and precisions over all the classes. The ablation results on N_{RR} and N_{RoI} are listed in the first three lines in Table 1. In Table 1, the results on T_g are listed in lines 3, 4, and 5, and the results on s_{an} are listed in lines 3, 6, and 7.

As shown the first three lines in Table 1, the mAP obtained by $N_{RR} = 512$ and $N_{RoI} = 256$ is 50.1%, therefore, $N_{RR} = 512$ and $N_{RoI} = 256$ take advantage in mAP. Although the ablation experiments on T_g with three different levels show that $T_g = 2 \times 10^{-4}$ is in favor of mAP, $T_g = 10^{-4}$ results in a greater recall with the same precision, as shown the lines 3, 4, and 5 in Table 1. In addition, $T_g = 10^{-4}$ results in a mAP of 50.1%, which is only 0.1% smaller than that obtained by $T_g = 2 \times 10^{-4}$. $T_g = 10^{-4}$ is employed as a reasonable geometric parameter in this study. As shown the ablation results on s_{an} listed in lines 3, 6, and 7 in Table 1, $s_{an} = (2, 4, 8, 16, 32)$ takes advantage in mAP. Overall, the parameters s_{an}, N_{RR}, N_{RoI}, and T_g are, respectively, tuned to be (2, 4, 8, 16, 32), 512, 256, and 10^{-4} for VGG16-based GP-Faster.

Besides the four parameters, extended experiments show that N^{pre}_{NMS} and N^{post}_{NMS}, which are, respectively, the number of top-scoring boxes to keep before and after applying NMS to RPN proposals in the stage of training, is desired to be tuned for ResNet101-based GP-Faster. The RPN parameters N^{pre}_{NMS}, N^{post}_{NMS}, s_{an}, N_{RR}, N_{RoI}, and T_g are, respectively, tuned to be 24 k, 4 k, (2, 4, 8, 16, 32), 448, 448, and 10^{-4} for ResNet101-based GP-Faster.

Experiments and results

We evaluate our method on two datasets: SUN2012[9] and NYUv2.[10] Our experiments are implemented based on the framework of Faster R-CNN.[4] Both VGG16[31] and ResNet101[33] are employed as our backbone networks. The VGG16-based and ResNet101-based experiments are, respectively, carried out on Caffe[34] and TensorFlow.[35] The publicly available VGG16 and ResNet101 models pretrained on ImageNet are used for corresponding initialization. For the sake of brevity, the standard Faster R-CNN implemented with the backbone networks of VGG16 and ResNet101 are, respectively, abbreviated as Faster16 and Faster101. We use a 1-GPU implementation, and thus, the minibatch size of RPN is 1. The VGG16 models are trained starting from conv3_1 using an end-to-end schedule. The ResNet101 models are trained starting from block2, that is, the parameter FIXED_BLOCKS is set to 1. We use a momentum of 0.9 and a weight decay of 5×10^{-4}. All the implementation results are reported on a dual-core i-3 4160 CPU (3.6 GHz) equipped with 16 GB RAM and an NVIDIA GTX1080 GPU.

Experiments and results on SUN2012

In this section, we evaluate GP-Faster on the SUN2012. We initialize a learning rate of 0.001 and make the learning rate drop 10 times after every 60 k iterations. A total of 100 k training iterations are run. To implement comparisons, we first run standard Faster R-CNN on the 18 classes collected from the SUN2012 set. Then, we run GP-Faster using the parameters in the aforementioned section. Table 2 presents our experimental results on the test set of SUN2012. The standard metric mAP is the average precision evaluated at IoU = 0.5. The columns

of GP-Faster16 and GP-Faster101 show the results of our method using VGG16 and ResNet101 as the backbone networks, respectively.

Table 2. Detection results on the SUN2012 test set (%).

Object	Faster16	Faster101	GP-Faster16	GP-Faster101
Wall	57.8	59.1	64.6	65.5
Wind	43.7	48.7	53.4	56.3
Floor	70.3	72.1	74.4	75.6
Ceiling	76.1	76.8	77.6	81.5
Plant	26.4	31.3	35.5	40.3
Door	28.8	33.0	35.1	35.6
Curtain	54.4	55.6	59.3	62.3
Painting	50.1	55.1	50.8	55.7
Chair	53.0	56.9	62.6	65.4
Person	56.7	61.2	65.6	69.1
Table	26.3	30.0	32.4	35.4
Cushion	44.6	50.6	57.1	57.8
Bottle	10.6	3.7	8.7	12.0
Desk lamp	73.4	72.6	76.2	79.5
Bed	67.7	73.3	76.7	76.2
Pillow	33.7	38.3	46.6	51.7
Sofa	34.7	38.8	51.1	46.4
TV	39.6	43.9	43.0	51.1
mAP	47.1	50.1	53.9	56.5

mAP: mean average precision.

As provided in Table 2, we compare our method with the standard Faster R-CNN. GP-Faster outperforms Faster R-CNN. GP-Faster16 and GP-Faster101 achieve mAPs of 53.9% and 56.5%, respectively. Compared with the baseline Faster R-CNN, corresponding improvements of the mAPs are, respectively, 6.8% and 6.4%. It can be seen that the 2D geometric property provides extra auxiliary discrimination.

Figure 4 shows some detection results on the SUN2012 test set. The implementation models are Faster16 and GP-Faster16 (53.9% mAP). A score threshold of 0.6 is used to draw the detection bounding boxes. The blue and red colors, respectively, show the detections launched by Faster16 and GP-Faster16.

Figure 4 demonstrates that the 2D geometric property is helpful for indoor object detection. On the one hand, some positive objects are undetected by Faster16, but they are detected by GP-Faster16, such as "painting" and "door" in Figure 4(a), "desk lamp" in Figure 4(c), "window" and "door" in Figure 4(d), "door," "plant," and "chair" in Figure 4(e), "painting" and "curtain" in Figure 4(i), "pillow" in Figure 4(j). On the other hand, GP-Faster16 corrects some false detection of Faster16, such as "chair" in Figure 4(b), "person" in Figure 4(f), "desk lamp" in Figure 4(g), "table" and "chair" in Figure 4(h). The

Figure 4. (a–j) Detection examples of Faster16 and GP-Faster16 on the SUN2012 test set. A score threshold of 0.6 is used to draw the detection bounding boxes. The blue and red colors, respectively, show the detections launched by Faster16 and GP-Faster16.

detection results suggest that GP-Faster is more powerful than Faster R-CNN on indoor object detection.

Experiments and results on NYUv2

In this section, we implement our proposed method on the NYUv2 dataset. The standard split of 795 training images and 654 testing images is employed for experiments in this work. To compare with the state-of-the-art methods[12,36,37] on the NYUv2 dataset, 19 classes are extracted for experiments. After the images are rescaled such that their shorter side is 600 pixels, Figure 5 shows the 2D constraints of the 19 classes. The scale parameter s_{an} on NYUv2 is set to $(4, 8, 16, 32)$. We initialize a learning rate of 0.001 to train the model. The total iterations and the step size of the learning rate are, respectively, set to 20 k and 30 k.[36] The geometric parameter T_g for GP-Faster on the dataset is set the same as that on SUN2012.

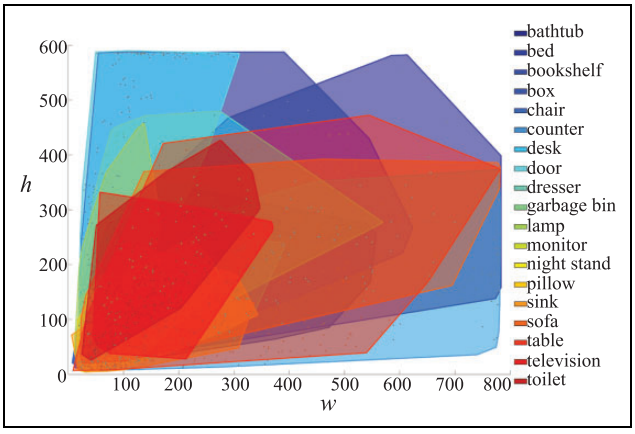

Figure 5. The 2D geometric constraint of 19 indoor classes on NYUv2.

Since NYUv2 is composed of pairs of RGB and depth frames that have been synchronized and annotated with dense labels for every image, we take the depth information into account in this section. After running over the depth frames of the dataset, we first extract the depths of all the objects with the help of the dense annotation. The depth constraint of every class is a maximum depth on corresponding objects. To avoid nontarget invasion from the background in the bounding box of RPN-RoI, a similar box with a quarter area centered in the RPN-RoI box is then cropped, and the depth values on the horizontal and vertical lines that are centered in the cropped box are used to approximate the object depth. The depth constraint is employed to refine RPN-RoI. In detail, the approximated object depth is required to be not greater than its corresponding class depth. Figure 6 shows our depth constraint. Figure 6(a) shows a depth frame. Figure 6(b) shows the depth of the "table" in Figure 6(a). Figure 6(c) shows an RPN-RoI of the "table" that overlaps with the "table" in Figure 6(a) with a certain IoU overlap. The color bar shows the depth values in meters in Figure 6(b) and (c). In Figure 6(c), the depth values on the red and white lines in the black box are employed to approximate the depth of the "table."

After combining 2D geometry and depth constraints to refine RPN-RoIs, we propose 3D geometric property-based Faster R-CNN (DGP-Faster) in this section. DGP-Faster16 and DGP-Faster101 are, respectively, implemented with the backbone networks of VGG16 and ResNet101. For DGP-Faster, the geometric parameter T_g is set to 2×10^{-4}. In addition, $N_{\text{NMS}}^{\text{pre}} = 12\,\text{k}$, $N_{\text{NMS}}^{\text{post}} = 2\,\text{k}$, $N_{\text{RR}} = 512$, and $N_{\text{RoI}} = 256$ are used for GP-Faster16 and DGP-Faster16. As for GP-Faster101 and DGP-Faster101, the four parameters are, respectively, set to be 24 k, 4 k, 512, and 256. Table 3 provides our detection results on the test set of NYUv2.

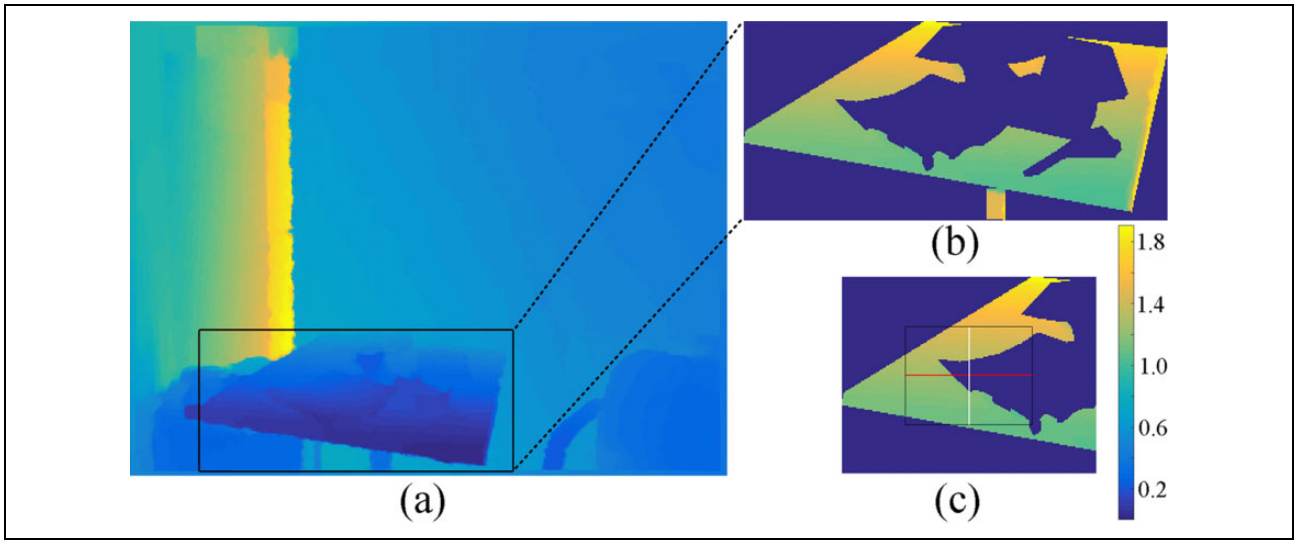

Figure 6. (a–c) Illustration of the depth constraint.

Table 3. Detection results on the NYUv2 test set (%).

Object	RCNN[3,36]	D-RCNN[36]	D-MH[37]	D-ROCK[12]	Faster16	GP-Faster16	DGP16	Faster101	GP-Faster101	DGP101
Bathtub	16.9	44.4	16.8	23.5	41.4	23.7	39.1	42.2	29.5	36.4
Bed	45.3	71.0	62.3	61.8	54.6	59.5	59.6	72.5	69.9	71.2
Bookshelf	28.5	32.9	41.8	43.0	35.1	42.1	37.7	42.7	45.6	45.3
Box	0.7	1.4	2.1	1.5	3.0	10.0	10.1	10.1	6.9	11.8
Chair	25.9	43.3	37.3	51.8	41.3	41.6	41.5	45.8	49.9	50.1
Counter	30.4	44.0	43.4	42.5	32.6	36.1	32.0	47.0	48.8	51.3
Desk	9.7	15.1	15.4	19.5	14.9	15.8	11.3	24.2	21.4	21.9
Door	16.3	24.5	24.4	35.7	23.0	27.9	30.1	35.2	31.6	32.8
Dresser	18.9	30.4	39.1	22.9	32.0	38.7	39.3	43.4	44.3	49.5
Garbage bin	15.7	39.4	22.4	39.0	28.6	31.9	31.0	38.4	40.0	36.6
Lamp	27.9	36.5	30.3	39.8	32.1	31.0	36.4	39.4	43.2	45.5
Monitor	32.5	52.6	46.6	40.0	39.6	45.5	42.4	42.6	49.2	50.5
Night stand	17.0	40.0	30.9	37.7	23.6	32.1	32.2	42.5	46.8	46.2
Pillow	11.1	34.8	27.0	38.5	26.8	33.0	31.7	33.1	39.3	36.8
Sink	16.6	36.1	42.9	36.6	32.3	28.9	31.4	42.5	44.0	46.8
Sofa	29.4	53.9	46.2	49.8	46.1	46.8	41.3	58.3	56.6	58.8
Table	12.7	24.4	22.2	22.0	22.5	21.4	22.6	26.9	30.8	29.3
TV	27.4	37.5	34.1	47.1	35.5	40.7	43.9	38.0	42.9	43.0
Toilet	44.1	46.8	60.4	53.1	58.2	52.2	60.4	54.5	56.0	59.5
mAP	22.5	37.3	34.0	37.1	32.8	34.7	35.5	41.0	41.9	43.3

mAP: mean average precision; DGP16: DGP-Faster16; DGP101: DGP-Faster101.

As presented in Table 3, we compare our method with the state-of-the-art models. After implementing Faster16 and Faster101 on the dataset of NYUv2, we obtain the baseline results, which are, respectively, 32.8% and 41.0%. As given in Table 3, GP-Faster16 and GP-Faster101, respectively, achieve mAPs of 34.7% and 41.9%. Compared with the baseline results, corresponding improvements on mAP are, respectively, 1.9% and 0.9%. It can be seen that the 2D constraints are also helpful for indoor object detection on the NYUv2 dataset. In addition, DGP-Faster16 and DGP-Faster101, which involve depth constraints in training, achieve mAPs of 35.5% and 43.3%, respectively. Compared with 2D geometric property-based detectors, DGP-Faster16 improves the mAP by 0.8% and DGP-Faster101 improves the mAP by 1.4%. DGP-Faster101 achieves the greatest mAP and outperforms all the state-of-the-art detectors in mAP. It can be seen that the depth constraint provides extra auxiliary discrimination. Overall, both the 2D geometry and depth constraints are helpful for indoor object detection.

We evaluate the inference time of our detectors on NYUv2. Although geometric constraints are not used for model test, our detection parameters $Nd_{\mathrm{NMS}}^{\mathrm{pre}}$ and $Nd_{\mathrm{NMS}}^{\mathrm{post}}$ are, respectively, set to 24 k and 1200, in which their default values are, respectively, 12 k and 600 in standard Faster R-CNN. The detection speed of Faster16 is approximately 10 fps. The detection speed of GP-Faster and DGP-Faster is approximately 4 fps. The main contribution of the inferiority may be the number of RoIs that are doubled. Although our proposed detectors are slower than the standard Faster R-CNN, geometric constraints do not directly contribute to the detection inferiority.

Discussion

To improve the performance of the indoor mobile robot, we proposed GP-Faster and DGP-Faster, which incorporates geometric property in Faster R-CNN to improve the detection performance. Faster R-CNN chooses RPN-RoIs to train the proposals. However, on the one hand, some outliers in RPN-RoIs are chosen as candidates. On the other hand, some small indoor objects cannot be covered by anchors generated by the standard Faster R-CNN. In this study, we employed the shape of the bounding box as a universal property and used geometric constraint to refine RPN-RoIs. In addition, we use mesh grids to generate appropriate anchors for indoor objects with the help of direct and inverse proportion functions. The comparison experiments implemented on the SUN2012 and NYUv2 datasets showed that GP-Faster improved the performance of the mAP. The experiments on NYUv2 showed that DGP-Faster achieved a further step in performance. It suggests that both the 2D geometric property and depth information are helpful for mobile robot to detect indoor object. However, the depth information is not always available, DGP-Faster may be limited for implementation in some applications.

Conclusions

In this article, a geometric property-based Faster R-CNN is proposed for indoor object detection. With the help of direct and inverse proportion functions, we first use mesh grids to generate appropriate anchors for indoor objects. After the anchors are regressed to RPN-RoIs, we then use the geometric constraints to refine the RPN-RoIs, in which

the geometric constraints may contain 2D size and depth information. The geometric constraints can remove some outliers in RPN-RoIs. With the help of geometric property, our proposed GP-Faster and DGP-Faster increase the mAP performance.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Anhui Provincial Natural Science Foundation (1808085MF171, 1908085MA07), and the National Natural Science Foundation of China (61672039, 61972439).

ORCID iD


Xintao Ding https://orcid.org/0000-0003-3325-3306

References

1. Liu L, Ouyang W, Wang X, et al. Deep learning for generic object detection: a survey. *Int J Comput Vis* 2020; 128(2): 261–318.
2. Krizhevsky A, Sutskever I, and Hinton GE. ImageNet classification with deep convolutional neural networks. In: Bartlett PL, Pereira FCN, Burges CJC, et al. (eds) *Advances in neural information processing systems 25*. Lake Tahoe, Nevada, USA: Curran Associates, Inc., 2012, pp. 1097–1105.
3. Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. Columbus, Ohio, USA, 23–28 June 2014, pp. 580–587. IEEE.
4. Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* 2017; 39(6): 1137–1149.
5. Redmon J and Farhadi A. Yolo9000: better, faster, stronger. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. Honolulu, Hawaii, USA, 21–26 July 2017, pp. 7263–7271. IEEE.
6. He K, Gkioxari G, Dollár P, et al. Mask R-CNN. In: *Proceedings of the IEEE international conference on computer vision*. Venice, Italy, 22–29 October 2017, pp. 2961–2969. IEEE.
7. Everingham M, Van Gool L, Williams CK, et al. The pascal visual object classes (VOC) challenge. *Int J Comput Vis* 2010; 88(2): 303–338.
8. Lin TY, Maire M, Belongie S, et al. Microsoft COCO: common objects in context. In: *European conference on computer vision*, part V (eds D Fleet, T Pajdla, B Schiele, et al.), Zurich, Switzerland, 6–12 September, 2014, pp. 740–755. Cham: Springer.
9. Xiao J, Hays J, Ehinger KA, et al. Sun database: large-scale scene recognition from abbey to zoo. In: *2010 IEEE computer society conference on computer vision and pattern recognition*. San Francisco, California, USA, 13–18 June 2010, pp. 3485–3492. IEEE.
10. Silberman N, Hoiem D, Kohli P, et al. Indoor segmentation and support inference from RGBD images. In: *European conference on computer vision*, part V (eds A Fitzgibbon, S Lazebnik, P Perona, et al.), Florence, Italy, 7–13 October 2012, pp. 746–760. Berlin, Heidelberg: Springer.
11. Gopan L and Aarthi R. A vision based DCNN for identify bottle object in indoor environment. In: Hemanth DJ and Smys S (eds) *Computational vision and bio inspired computing*. Cham: Springer, 2018, pp. 447–456.
12. Mordan T, Thome N, Henaff G, et al. Revisiting multi-task learning with ROCK: a deep residual auxiliary block for visual detection. In: *32nd conference on neural information processing systems (NeurIPS)* (eds S Bengio, HM Wallach, H Larochelle, et al..), Montréal, 2018, pp. 1317–1329. Canada: Curran Associates, Inc.
13. Zheng C, Wang J, Chen W, et al. Multi-class indoor semantic segmentation with deep structured model. *Visual Comput* 2018; 34(5): 735–747.
14. Ammirato P, Fu CY, Shvets M, et al. Target driven instance detection. *arXiv preprint arXiv: 1803.04610*, 2018.
15. Ehsan TZ and Nahvi M. Violence detection in indoor surveillance cameras using motion trajectory and differential histogram of optical flow. In: *Proceedings of the international conference on computer and knowledge engineering (ICCKE)*. Mashhad, Iran, 25–26 October 2018, pp. 153–158. IEEE.
16. Zhou F, Hu Y, and Shen X. MFDCNN: a multimodal fusion DCNN framework for object detection and segmentation. In: *Pacific rim conference on multimedia* (eds R Hong, WH Cheng, T Yamasaki, et al.), Hefei, China, 21–22 September, 2018, pp. 3–13. Springer International Publishing.
17. Reyes E, Gómez C, Norambuena E, et al. Near real-time object recognition for pepper based on deep neural networks running on a backpack. In: Holz D, Genter K, and Saad M (eds) *Robot world cup*. Cham: Springer, pp. 287–298.
18. Zhu J, Li Q, Cao R, et al. Indoor topological localization using a visual landmark sequence. *Remote Sens* 2019; 11(1): 73.
19. Jiang S, Yao W, Hong Z, et al. A classification-lock tracking strategy allowing a person-following robot to operate in a complicated indoor environment. *Sensors* 2018; 18(11): 3903.
20. Loghmani MR, Planamente M, Caputo B, et al. Recurrent convolutional fusion for RGB-D object recognition. *IEEE Robot Autom Lett* 2019; 4(3): 2878–2885.
21. Sampedro C, Rodriguez-Ramos A, Bavle H, et al. A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques. *J Intell Robot Syst* 2019; 95(2): 601–627.
22. Kowalewski S, Maurin AL, and Andersen JC. Semantic mapping and object detection for indoor mobile robots. In: *IOP conference series: materials science and engineering, proceedings of the 2nd international conference on*

robotics and mechantronics. Singapore, 9–11 November 2018, p. 012012. IOP Publishing.

23. Wu WY and Wang MJJ. Elliptical object detection by using its geometric properties. *Pattern Recognit* 1993; 26(10): 1499–1509.
24. Batool N and Chellappa R. Fast detection of facial wrinkles based on Gabor features using image morphology and geometric constraints. *Pattern Recognit* 2015; 48(3): 642–658.
25. Ismail A, Seifelnasr M, and Guo H. Understanding indoor scene: spatial layout estimation, scene classification, and object detection. In: *Proceedings of the 3rd international conference on multimedia systems and signal processing*. Shenzhen, China, April 2018, pp. 64–70. New York, United States: ACM.
26. Pham TT, Do TT, Sünderhauf N, et al. SceneCut: joint geometric and object segmentation for indoor scenes. In: *Proceedings of the IEEE international conference on robotics and automation (ICRA)*. Brisbane, QLD, Australia, 21–25 May 2018, pp. 1–9. IEEE.
27. Mizginov V and Danilov S. Synthetic thermal background and object texture generation using geometric information and GAN. *Int Arch Photogramm Remote Sens Spat Inform Sci* 2019; 42: 149–154.
28. Cai J, Hou J, Lu Y, et al. Improving CNN-based planar object detection with geometric prior knowledge. In: *2020 IEEE international symposium on safety, security, and rescue bobotics (SSRR)*, Abu Dhabi, UAE, United Arab Emirates, 4–6 November 2020, pp. 387–393. IEEE.
29. Amin S and Galasso F. Geometric proposals for Faster R-CNN. In: *14th IEEE international conference on advanced video and signal based surveillance (AVSS)*. Lecce, Italy, 29 August 2017–1 September 2017, pp. 1–6. IEEE.
30. Preparata FP and Shamos MI. *Computational geometry: an introduction*. Berlin: Springer Science & Business Media, 2012.
31. Simonyan K and Zisserman A. Very deep convolutional networks for large-scale image recognition. In: *International conference on learning representations*. San Diego, CA, USA, 7–9 May 2015, pp. 1–14.
32. Quattoni A and Torralba A. Recognizing indoor scenes. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. Miami, Florida, USA, 20–25 June 2009, pp. 413–420. IEEE.
33. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. Las Vegas, Nevada, USA, 27–30 June 2016, pp. 770–778. IEEE.
34. Jia Y, Shelhamer E, Donahue J, et al. Caffe: convolutional architecture for fast feature embedding. In: *Proceedings of the 22nd ACM international conference on multimedia*. Orlando, Florida, USA, November 2014, pp. 675–678. New York, United States: ACM.
35. Abadi M, Barham P, Chen J, et al. TensorFlow: a system for large-scale machine learning. In: *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*. Savannah, Georgia, USA, November 2016, pp. 265–283.
36. Gupta S, Girshick R, Arbeláez P, et al. Learning rich features from RGB-D images for object detection and segmentation. In: *European conference on computer vision*, part VII (eds D Fleet, T Pajdla, B Schiele, et al.), Zurich, Switzerland, 6–12 September 2014, pp. 345–360. Cham: Springer.
37. Hoffman J, Gupta S, and Darrell T. Learning with side information through modality hallucination. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. Las Vegas, Nevada, USA, 27–30 June 2016, pp. 826–834.