Depth Privileged Object Detection in Indoor Scenes via Deformation Hallucination

Zhijie Zhang, Yan Liu, Junjie Chen, Li Niu*, Liqing Zhang*
MoE Key Lab of Artificial Intelligence,
Department of Computer Science and Engineering,
Shanghai Jiao Tong University
{zzj506506, loseover, chen.bys, ustcnewly}@sjtu.edu.cn, zhang-lq@cs.sjtu.edu.cn

Abstract

RGB-D object detection has achieved significant advance, because depth provides complementary geometric information to RGB images. Considering that depth images are unavailable in some scenarios, we focus on depth privileged object detection in indoor scenes, where the depth images are only available in the training stage. Under this setting, one prevalent research line is modality hallucination, in which depth image and depth feature are common hallucination targets. In contrast, we choose to hallucinate depth deformation, which benefits a lot from rich geometric information in depth data. Specifically, we employ the deformable convolutional layer with augmented offsets to perform geometric deformation, because the offsets enable flexibly sampling over the object and transforming to a canonical shape for ease of object detection. In addition, we design a quality-based weighted transfer loss to avoid negative transfer of depth deformation. Experimental results on NYUDv2 and SUN RGB-D demonstrate the effectiveness of our method against the state-of-the-art methods for depth privileged object detection.

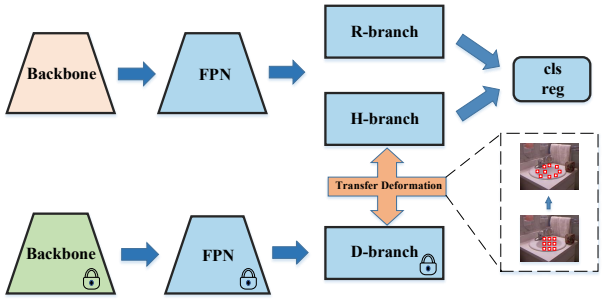


Figure 1: The overview of our framework. In the training stage, the geometric deformation is transferred from D-branch to H-branch. In the testing stage without depth, H-branch hallucinates the depth deformation to complement R-branch. D-branch is pre-trained and frozen. The red squares represent sampling locations.

Introduction

Object detection in indoor scenes is a fundamental yet challenging step towards scene understanding. Up to now, cluttered objects remain difficult to be detected due to the large variations (*e.g.*, occlusion and illumination) in appearances and boundaries. Fortunately, depth images provide color-insensitive information in representing objects and boundaries, and the advantage of depth images for object detection has been demonstrated in (Cadena and Košecka 2015; Gupta et al. 2014; Li et al. 2018). However, depth data is not always available because depth sensors are much less prevalent than RGB capturing devices. To this end, we consider depth privileged object detection, in which depth (*i.e.*, the privileged information (Vapnik and Vashist 2009)) is only available in the training stage yet unavailable in the testing stage.

As far as we know, there are only a few works on depth-privileged object detection. One research line is multi-task learning. For example, ROCK (Mordan et al. 2018) simultaneously performed object detection and depth prediction, so that the intermediate features are enriched. Another research line is modality hallucination. For instance, Hoffman *et al.* (2016) used RGB input to hallucinate intermediate features learned from depth modality, which can function as the unavailable depth modality during testing. Analogously, Cao *et al.* (2016) directly hallucinated depth images to complement RGB images. Compared with modality hallucination, multi-task learning implicitly distills knowledge from depth images and depends on the relevance of multiple tasks, so we tend to follow the research line of modality hallucination.

Modality hallucination requires an appropriate target to hallucinate. The depth image chosen in (Cao, Shen, and Shen 2016) and the depth feature chosen in (Hoffman, Gupta, and Darrell 2016) have many redundancies and may be difficult to hallucinate. In contrast, in this paper, we propose to hallucinate an elegant yet informative target: geometric deformation. On the one hand, the geometric deformation explicitly fits the object shape and transforms to a canonical shape, which has shown a great advantage in the object detection task (Dai et al. 2017; Zhu et al. 2019). On the other hand, the geometric deformation learned from RGB images is often confused by cluttered colors, which can be mitigated by exploiting the rich geometric information from depth images. We refer to the deformation learned from RGB (*resp.*, depth) input as RGB (*resp.*, depth) defor-

*Corresponding author

mation. Depth deformation is supposed to capture geometric cues more precisely and thus be complementary to RGB deformation. However, we only have RGB images in the testing stage, so our goal is to hallucinate depth deformation from RGB modality.

To this end, as shown in Figure 1, our network consists of three branches: RGB branch (R-branch), hallucination branch (H-branch), and depth branch (D-branch), all equipped with deformation modules. All three branches are basic detection models, in which R-branch and H-branch take RGB images as input while D-branch takes depth images as input. In practice, R-branch and H-branch can share the backbone resulting in a more compact architecture compared with (Cao, Shen, and Shen 2016; Hoffman, Gupta, and Darrell 2016), probably because geometric deformation is easier to hallucinate. In the training stage, besides typical detection losses, the deformation module in H-branch is further forced to hallucinate the mature deformation generated by the pre-trained D-branch. In the testing stage, we leverage both RGB deformation and hallucinated deformation by fusing the outputs from R-branch and H-branch, without using D-branch.

For the deformation module, we opt for deformable convolutional layer (Dai et al. 2017), referred to as DeformConv, which augments the sampling locations of standard convolutional kernel with offsets and produces a flexible sampling grid. The reasons for choosing DeformConv are twofold. On the one hand, we can integrate DeformConv in any elaborate architecture such as ResNeXt (Xie et al. 2017) without modification. On the other hand, the deformation in DeformConv is light-weighted, which is an ideal hallucination target. Two examples are provided to show the effectiveness of DeformConv. In Figure 2(a), the RGB deformation concentrates on the region with the similar color. While in Figure 2(b), the depth deformation can adapt the sampling locations to the shape of the object by being aware of depth boundaries. For instance, the RGB deformation of the door is confused by the background with similar color, while the depth deformation can fit the door frame. Comparing the offsets between RGB and depth, we find that depth deformation focuses on geometric cues and thus can complement RGB deformation to some extent.

Considering that the depth deformation is not always reliable, the model may be misled by the hallucinated inaccurate deformation. To avoid such negative transfer, we propose a weighted transfer loss by using the quality of depth deformation to control the amount of transfer. We conduct extensive experiments on NYUDv2 dataset and SUN RGB-D dataset. The results and analyses demonstrate that deformation hallucination can greatly benefit depth privileged object detection. In summary, our main contributions are as follows:

- We propose to hallucinate deformation for depth-privileged object detection, which has never been investigated before.
- We design a weighted transfer loss to avoid negative transfer of depth deformation.
- Our method outperforms all state-of-the-art methods on both NYUDv2 dataset and SUN RGB-D dataset.

(a) RGB Deformation (b) Depth Deformation

Figure 2: Illustration of RGB deformation (a) and depth deformation (b). The red squares represent the 81 (9×9) sampling locations for an output location (blue square) by tracing back two DeformConvs with 3×3 convolutional kernels. We also show the ground-truth bounding box of the object corresponding to the output location.

Related work

RGB-D Object Detection

Over the years, there have been considerable works on RGB-D object detection (Ye and Malik 2013; Gupta et al. 2014, 2015), which integrated depth representation into RGB representation and achieved significant improvement over RGB-only methods. Many methods (Bo et al. 2011; Blum et al. 2012) incorporated the information of two different modalities (*i.e.*, RGB and depth) at the input level. Recently, most works (Cao, Shen, and Shen 2016; Xu et al. 2017; Rahman et al. 2019) focused on the fusion of features extracted from two modalities. For example, Hoffman *et al.* (2016) proposed to fuse RGB and depth mid-level features, leading to better performance with little additional annotation effort. Gated information fusion network (Kim et al. 2018) weighed the contribution of each modality according to the input feature maps. Contrary to these approaches, our work focuses on learning an RGB only model that hallucinates complementary deformation from depth images in the training stage.

Learning Using Privileged Information

Learning Using Privileged Information (LUPI) was first introduced in (Vapnik and Vashist 2009), where the privileged information is only available in the training stage but not available in the testing stage. LUPI has been explored in a wide range of applications such as image classification (Yan et al. 2016), semantic segmentation (Lee et al. 2019), object localization (Feyereisl et al. 2014), image aesthetic assessment (Pan, Wang, and Jiang 2019), text clustering (Marcacini and Rezende 2013), and *etc.* Recently, many works (Hoffman, Gupta, and Darrell 2016; Mordan et al. 2018;

Cao, Shen, and Shen 2016) have investigated using depth as privileged information for object detection. Specifically, ROCK (Mordan et al. 2018) conducted multi-task learning (*i.e.*, depth prediction and object detection) to inject intermediate auxiliary depth representation into the primary RGB representation. HallucinationNet (Hoffman, Gupta, and Darrell 2016) hallucinated the mid-level depth features using RGB images to make up for the absence of depth features in the testing stage, while Cao *et al.* (2016) chose to hallucinate the depth images estimated by DCNF model (Liu et al. 2015). In contrast, we hallucinate geometric deformation learned from depth images, which benefits from rich geometric information in depth images.

Knowledge Distillation

Hinton *et al.* (2015) introduced the new idea of knowledge distillation (KD), where the knowledge distilled from a teacher-net is transferred to a student-net to achieve higher performance with lower complexity. Generally, KD can be categorized into single teacher-net distillation (Yang et al. 2019), multi teacher-nets distillation (Shen et al. 2019), online distillation (Zhang et al. 2018), cross-modal distillation (Do et al. 2019), self distillation (Xu and Liu 2019), and *etc*. Among them, cross-modal distillation is most related to our work, which has been used in pose estimation (Zhao et al. 2018), VQA (Do et al. 2019), tracking (Gan et al. 2019), segmentation (Dou et al. 2020), scene recognition(Du et al. 2019), and *etc*. For object detection, Hoffman *et al.* (2016) distilled the mid-level features from the RGB branch to the depth branch and then fused the features extracted by these two branches. Gupta *et al.* (2016) transferred the supervision from the labeled modality (*i.e.*, RGB) to the unlabeled modality (*i.e.*, depth). Li *et al.* (2017) used student-net to mimic the feature map extracted by teacher-net. Unlike the above methods, what we distill and transfer is the geometric deformation extracted by the deformable convolutional layer (Dai et al. 2017), which is much more correlated to the geometric cues offered by depth.

Background

The deformable convolutional layer (Dai et al. 2017) lays the foundation of our framework, which is built by augmenting the standard convolutional layer with offsets, as shown in Figure 3.

For the standard convolution, we denote the pixel-wise vector at the location i (2D spatial location (i_x, i_y)) on the input and output feature map as $x(i)$ and $y(i)$, respectively. The standard convolution can be formulated as

$$y(i) = \sum_{k=1}^{K^2} v_k \cdot x(i + l_k), \tag{1}$$

where v_k denotes the weight vector at the k-th sampling location of convolutional kernel, l_k is the offset from kernel center corresponding to the k-th sampling location, and K is the size of convolutional kernel. For instance, for a 3×3 convolutional kernel with dilation 1, we have $K = 3$ and $l_k \in \{(-1,-1),(-1,0),\ldots,(1,1)\}$ for $k = 1,\ldots,9$.

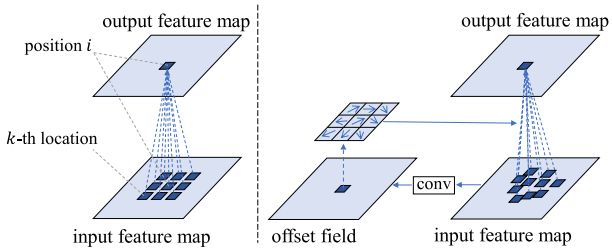


Figure 3: Comparison between the standard convolution (left) and the deformable convolution (right).

Then, deformable convolutional layer applies extra 2D offsets to shift the sampling locations to the regions of interest, which can be formulated as

$$y(i) = \sum_{k=1}^{K^2} v_k \cdot x(i + l_k + \Delta l_{i,k}), \tag{2}$$

Compared with (1), deformable convolution (2) introduces extra offset $\Delta l_{i,k}$, which denotes the offset of the k-th sampling location when performing convolution at location i. Bilinear interpolation is employed to compute the value of $x(i+l_k+\Delta l_{i,k})$, because $i+l_k+\Delta l_{i,k}$ may not be an integer. As shown in the right of Figure 3, the offsets are learned by a separate convolutional layer, which is applied over the same input feature map x and outputs $2 \times K^2$ (*i.e.*, 18) channels for each location, which correspond to 2-D offsets of 3×3 sampling locations.

Methodology

The proposed framework is illustrated in Figure 4, which consists of three branches: RGB branch (R-branch), hallucination branch (H-branch), and depth branch (D-branch). The H-branch learns to hallucinate the geometric deformation generated by D-branch, which is fused with R-branch to accomplish the detection task. The details of network architecture and our method will be introduced as follows.

Network Architecture

In this section, we first introduce the basic detection branch, which forms the basis of our R-branch, H-branch, and D-branch. Then, we introduce our whole hallucination framework with three branches.

The Detection Branch We choose the fully convolutional one-stage object detection (FCOS) (Tian et al. 2019) as our basic detection branch due to its simplicity and popularity, as shown in each branch of Figure 4. Firstly, the backbone extracts the feature pyramid containing multiple scales of feature maps, which accounts for detecting various scales of objects. Secondly, the detection head detects objects on each feature map in a fully-convolutional manner. In this way, there is one predicted object at each spatial position on each feature map. Specifically, at position (s, i), *i.e.*, the i-th spatial position on the s-th scale of feature map, the detection head outputs three types of values for a potential object:

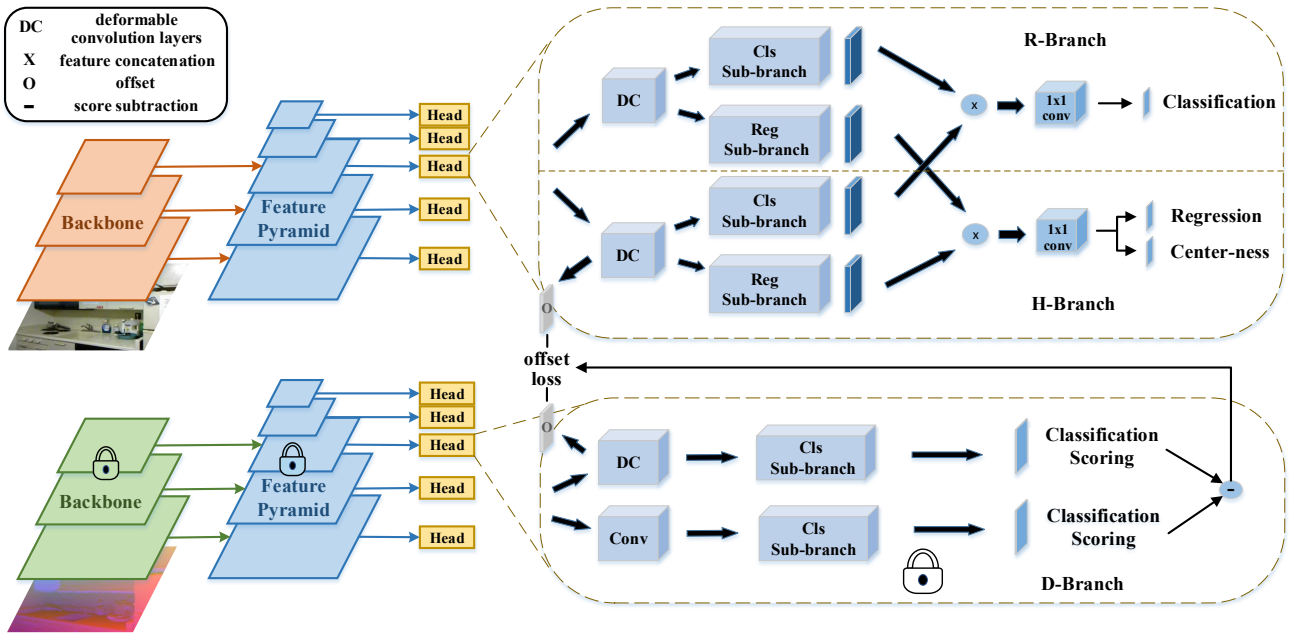


Figure 4: Detailed architecture of our framework. The DeformConvs in H-branch are forced to hallucinate the deformation generated in D-branch. The hallucination procedure is weighted by the quality measurement in D-branch to avoid negative transfer. The D-branch is pre-trained and frozen.

1) the predicted category $c_{s,i}$; 2) the predicted centerness $d_{s,i}$, which is the normalized distance from (s, i) to the center of object (Tian et al. 2019); 3) the predicted bounding box $\{x^1_{s,i}, y^1_{s,i}, x^2_{s,i}, y^2_{s,i}\}$, which predicts the coordinates of bounding box.

In the training stage, for the position (s, i), if its neighborhood (determined by the Adaptive Training Sample Selection approach (Zhang et al. 2020)) contains a ground-truth object with a category $c^*_{s,i}$, a centerness $d^*_{s,i}$ and a bounding box $\{x^{1*}_{s,i}, y^{1*}_{s,i}, x^{2*}_{s,i}, y^{2*}_{s,i}\}$, we refer to this position as a positive position, which is supervised to predict accurate category, centerness, and bounding box. Otherwise, the position (s, i) is referred to as a negative position, for which only the category $c_{s,i}$ is supervised by the "background" category. The overall loss of detection branch is summarized as

$$\mathcal{L}_d = \sum_{s=1}^{S} \sum_{i=1}^{h_s \times w_s} [L_{focal}(c_{s,i}, c^*_{s,i}) + \alpha P_{s,i} L_{BCE}(d_{s,i}, d^*_{s,i}) + \beta P_{s,i} L_{GIoU} \left(\{x^1_{s,i}, y^1_{s,i}, x^2_{s,i}, y^2_{s,i}\}, \{x^{1*}_{s,i}, y^{1*}_{s,i}, x^{2*}_{s,i}, y^{2*}_{s,i}\}\right)], \quad (3)$$

where S is the total number of scales, $h_s \times w_s$ is the feature map size of the s-th scale, $P_{s,i}$ is an indicator function ($P_{s,i} = 1$ if (s, i) is a positive position and 0 otherwise). L_{focal} is the focal loss (Lin et al. 2017), L_{GIoU} is the generalized intersection over union loss (Rezatofighi et al. 2019), and L_{BCE} is binary cross entropy (BCE) loss. Following (Zhang et al. 2020), two trade-off parameters α and β are set as 1.0 and 2.0, respectively.

For more details, please refer to FCOS (Tian et al. 2019) and ATSS (Zhang et al. 2020).

The Hallucination Framework Inspired by (Hoffman, Gupta, and Darrell 2016), we employ three detection branches to hallucinate deformation. All three branches are built upon the abovementioned basic detection branch with the same architecture: backbone, FPN, and detection heads. As illustrated in Figure 4, R-branch and H-branch share the backbone and FPN but have different detection heads.

Each scale of feature map in FPN is associated with a detection head which consists of a classification sub-branch and a regression sub-branch, in which the former predicts category while the latter predicts bounding box and centerness. We enforce two sub-branches to share the first M convolutional layers, which are replaced by M DeformConvs. For the detection heads attached to the same scale of the feature map, we concatenate the classification (*resp.*, regression) sub-branch outputs of R-branch and H-branch. The concatenated output passes through a 1×1 convolutional layer to reduce the channels for classification (*resp.*, regression).

In the training stage, we first pre-train and freeze the D-branch for stable and mature deformation guidance for H-branch. In the testing stage, we only use R-branch and H-branch. Because each detection head has M DeformConvs, in the following sections, we introduce another index m to indicate the m-th DeformConv.

Deformation Hallucination

To fulfil deformation hallucination, we force the offsets generated by H-branch to mimic the offsets generated by D-branch with the transfer loss:

$$\mathcal{L}_o = \sum_{s=1}^{S}\sum_{m=1}^{M}\sum_{i=1}^{h_s\times w_s}\sum_{k=1}^{K^2}\left\|\Delta l^D_{s,m,i,k} - \Delta l^H_{s,m,i,k}\right\|_2^2, \quad (4)$$

where $\Delta l^H_{s,m,i,k}$ (*resp.*, $\Delta l^D_{s,m,i,k}$) indicates the 2D offset for the k-th sampling location at position (s,i) in the m-th DeformConv of H-branch (*resp.*, D-branch).

Besides, another intuitive alternative is to conduct deformation hallucination only on positive positions, because better deformation information may be learned on positive positions due to stronger supervision (see (3)). Similar to (3), we use $P_{s,m,i}$ to indicate whether the position (s,i) in the m-th DeformConv is a positive position. For the last DeformConv ($m=M$), the positive positions ($P_{s,m,i}=1$) are determined in the same way as in (3). Then, we calculate the positive positions in previous DeformConvs by tracking their contributions to the supervision of object detection. The details are left to the Supplementary.

After defining positive positions in each DeformConv, we rewrite the loss (4) as

$$\mathcal{L}^p_o = \sum_{s=1}^{S}\sum_{m=1}^{M}\sum_{i=1}^{h_s\times w_s}\sum_{k=1}^{K^2} P_{s,m,i}\left\|\Delta l^D_{s,m,i,k} - \Delta l^H_{s,m,i,k}\right\|_2^2. \quad (5)$$

The experimental results (see Table 1) show that hallucination on positive positions yields better performance than hallucination on all positions.

Avoid Negative Transfer

Considering that the depth deformation is not always reliable, transferring noisy deformation is likely to degrade the performance. Thus, we design a weighted transfer loss to avoid such negative transfer, in which the weights are determined by deformation quality.

We conjecture that the performance can be improved only by reliable deformation, so we compute the performance gain brought by deformation to measure the deformation quality. Then, we assign the deformation transfer losses (4) at different positive positions with different weights, which are calculated based on the deformation quality at each positive position. Specifically, we add another classification sub-branch in D-branch, which is almost the same as the original classification sub-branch except replacing DeformConv with standard conv, as shown in Figure 4. We denote the loss weight assigned at position (s,i) in the m-th DeformConv as $w_{(s,m,i)}$.

Firstly, the loss weights in the last DeformConv ($m=M$) can be calculated by

$$w_{s,m,i} = exp(\delta\cdot(f^{(w)}_{s,i} - f^{(w/o)}_{s,i})), \quad (6)$$

where $f^{(w)}_{s,i}$ (*resp.*, $f^{(w/o)}_{s,i}$) is the classification score corresponding to the ground-truth category $c^*_{s,i}$ at position (s,i) predicted by the classification sub-branch with (*resp.*, without) deformation. δ is a hyper-parameter controlling the intensity of avoiding negative transfer and set as 0.25 via cross-validation.

Secondly, we back-propagate the loss weights from the last DeformConv to previous DeformConvs ($m<M$). Intuitively, we calculate the loss weights at different positions in previous DeformConvs by tracking their contributions to the classification score improvement. The details of computing $w_{s,m,i}$ are left to Supplementary.

After obtaining all weights $w_{s,m,i}$, we can formulate the weighted transfer loss $\mathcal{L}^{pw}_o$ as

$$\mathcal{L}^{pw}_o = \sum_{s=1}^{S}\sum_{m=1}^{M}\sum_{i=1}^{h_s\times w_s}\sum_{k=1}^{K^2} w_{s,m,i}P_{s,m,i}\left\|\Delta l^D_{s,m,i,k} - \Delta l^H_{s,m,i,k}\right\|_2^2. \quad (7)$$

The Full Objective Function

By taking both object detection loss and weighted transfer loss into consideration, the overall objective function can be formulated as

$$\mathcal{L}_{full} = \mathcal{L}_d + \mu\mathcal{L}^{pw}_o, \quad (8)$$

where μ is a trade-off parameter and set as 0.1 via corss-validation.

Experiment

Datasets

NYU Depth V2 NYU Depth V2 (NYUDv2) (Silberman et al. 2012) consists of 1449 paired RGB-D images. The dataset is split into training (795 images) and test (654 images) sets. Following previous works (Hoffman et al. 2016; Mordan et al. 2018), we train and evaluate our model based on the 19 most common categories.

SUN RGB-D SUN RGB-D (Song, Lichtenberg, and Xiao 2015) is composed of an official train/test split with 5285 and 5050 images, respectively. We train and evaluate our model based on 19 common categories (the same as NYUDv2) under the standard setting in (Song, Lichtenberg, and Xiao 2015).

Implementation Details

We use ResNeXt-101 (Xie et al. 2017) pretrained on ImageNet (Deng et al. 2009) as backbone. We set the number of DeformConvs M to 2 by default. Following (Gupta et al. 2014), we adopt the HHA encoding (Horizontal disparity, Height above ground, Angle with gravity) for depth input. We train our model using the SGD optimizer for 50k iterations for D-branch pre-training and the whole model training. The basic learning rate is initialized to 1×10^{-3} and reduced to 1×10^{-4} when the iterations reach 40k. The weight decay and momentum are set to 5×10^{-4} and 0.9, respectively. The random seed is set to 222. Besides, we conduct a significant test using 10 different random seeds and analyse hyper-parameters (*i.e.,* δ in (6) and μ in (8)), which are reported in the Supplementary. All experiments are conducted on Ubuntu 18.04 with two 8GB GeForce RTX 2080 SUPER,

$\mathcal{L}_d$	$\mathcal{L}_o$	$\mathcal{L}_o^p$	$\mathcal{L}_o^{pw}$	mAP(%)	
				NYUDv2	SUN RGB-D
✓				44.01	53.93
✓	✓			46.26	56.15
✓		✓		46.50	56.47
✓			✓	46.88	56.84

Table 1: Ablation studies on loss terms on NYUDv2 and SUN RGB-D. mAP represents the mean Average Precision of 19 categories.

16GB Intel 9700K, and PyTorch 1.2.0 on Python 3.7. In this paper, we use Average Precision (AP) with IoU threshold 0.5 as our evaluation metric.

Ablation Studies

We conduct ablation studies on NYUDv2 and SUN RGB-D datasets to investigate the impact of different loss terms. The results are summarized in Table 1.

The Effect of Deformation Hallucination In row 2, we apply the transfer loss to all positions without using weights (see (4)). By using the transfer loss, the deformation generated by H-branch is forced to mimic the depth deformation. By comparing row 2 with row 1, we can see that hallucinating depth deformation with RGB input dramatically increases the performance, *i.e.*, 46.26% *v.s.* 44.01% on NYUDv2 dataset and 56.15% *v.s.* 53.93% on SUN RGB-D dataset.

The Effect of Positive Position Only In row 3, we only apply the transfer loss to positive positions without using weights (see (5)). By comparing row 3 with row 2, the results (46.50% *v.s.* 46.26%, and 56.47% *v.s.* 56.15%) indicate that the deformation hallucination merely targeted at positive positions could make the transferred deformation more useful.

The Effect of Avoiding Negative Transfer The results in row 4 are from our full method (see (8)), which employs weighted transfer loss to avoid negative transfer. By comparing row 4 with row 3, the results are further improved (46.88% *v.s.* 46.50% and 56.84% *v.s.* 56.47%), which verifies that unreliable partial deformation is prevented from being transferred.

Comparison with State-Of-The-Art

In this section, we compare our method with the state-of-the-art approaches on both datasets, including HallucinationNet (Hoffman, Gupta, and Darrell 2016), ROCK (Mordan et al. 2018), and Cao *et al.* (2016). Note that all the methods above adopt different detection architectures as well as different backbones. For a fair comparison, we reproduce them using the same detection network with the same backbone (*i.e.* FCOS+ATSS with ResNeXt-101 backbone) as ours. Besides, we report the result of the basic detection network (*i.e.* FCOS+ATSS) in Table 2 as a benchmark.

For HallucinationNet, we hallucinate the depth feature extracted by the FPN. For ROCK, based on a detection network in RGB modality, we insert the residual auxiliary block between FPN and each detection head, in which the auxiliary task-specific (depth prediction) features are fused by element-wise addition in residual style. For (Cao, Shen, and Shen 2016), we first adopt DCNF (Liu et al. 2015) with RGB images to estimate depth images, and then we learn deep depth features from estimated depth images, which are combined with RGB features and fed into the detection head.

Method	mAP(%)	
	NYUDv2	SUN RGB-D
FCOS+ATSS	42.73	52.94
HallucinationNet	45.22	55.35
ROCK	44.89	55.14
Cao *et al.* (2016)	44.96	55.27
Ours	**46.88**	**56.84**

Table 2: Comparison with state-of-the-art approaches on NYUDv2 and SUN RGB-D. The best results are denoted in boldface.

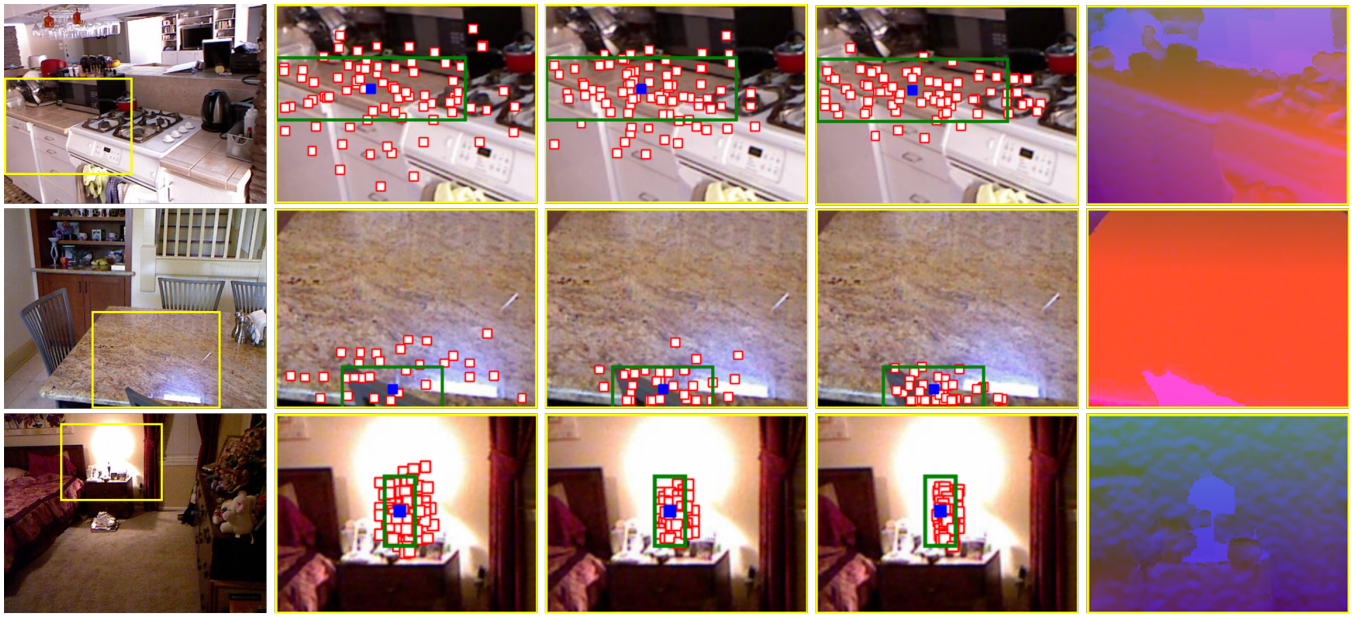
Configuration	mAP(%)	
	$\mathcal{L}_d$	$\mathcal{L}_d + \mu\mathcal{L}_o$
B_1	43.66	45.34
B_2	44.17	45.93
B_3	44.43	46.19
$H1$	43.35	45.36
$H2$	44.01	46.26
$H3$	44.21	46.30
$B_3 + H_2$	44.25	46.32

Table 3: Results using different DeformConv configurations. In terms of the location to use DeformConvs, we denote backbone as 'B' and detection head as 'H'. The subscript (1, 2, 3) means the number of DeformConvs.

All results are summarized in Table 2. It is noticeable that all methods can achieve better performance than the basic detection network. Furthermore, we can find that our method outperforms all the baselines on NYUDv2 and SUN RGB-D datasets. Specifically, our method beats the best baseline on two datasets (*i.e.*, 46.88% *v.s.* 45.22% and 56.84% *v.s.* 55.35%), which demonstrates the superiority of deformation hallucination.

Configuration of DeformConvs

In this section, we explore various configurations of Deformable Convolutional Layers (DeformConvs) from two main aspects: location and number. By default, we choose the detection head to replace standard convs with DeformConvs. However, it is also feasible to choose the last few layers of backbone as in (Dai et al. 2017). For both locations, we consider replacing an appropriate number (*i.e.*, up to 3) of standard convolutional layers. Here we only use the loss $\mathcal{L}_d + \mu\mathcal{L}_o$ because it is hard to define positive positions or measure the deformation quality for the backbone location. By taking the NYUDv2 dataset as an example, all results are summarized in Table 3.



(a) RGB images (b) RGB Deformation (c) Hallucinated Deformation (d) Depth Deformation (e) HHA-encoded images

Figure 5: Illustration of deformation visualization with two DeformConvs. (a) shows the RGB images with the bounded area to be visualized in the follow-up images. (b), (c), and (d) show the shifted sampling locations learned from R-branch, H-branch, and D-branch, respectively. (e) shows the corresponding HHA-encoded depth images.

As shown in Table 3, on the one hand, increasing the number of DeformConvs ($1 \rightarrow 3$) in either location can boost performance. By comparing $\mathcal{L}_d + \mu\mathcal{L}_o$ and $\mathcal{L}_d$, deformation hallucination can consistently bring performance gain when using different numbers of DeformConvs in either location. On the other hand, replacing standard convs in detection head outperforms that in the backbone, probably because the deformation in detection head is more mature to represent the geometric information of individual object. For $\mathcal{L}_d + \mu\mathcal{L}_o$, replacing standard convs in both locations only achieves slight improvement ($B_3 + H_2$ *v.s.* H_2), so we adopt the option H_2 unless otherwise specified, considering the trade-off between performance and model complexity.

Deformation Visualization

To better explore how DeformConvs capture geometric information and how the deformation hallucination can benefit object detection, we visualize RGB, hallucinated, and depth deformation by using two DeformConvs in detection head (H_2 in Table 3) in Figure 5.

RGB deformation has a strong representation ability for color information, which mainly focuses on the recognizable area with the similar color around the kernel center. In contrast, the depth deformation is more sensitive to geometric information and thus can capture the complementary contour, edge, and shape information. Hallucinated deformation is forced to mimic depth deformation to capture more geometric information. By taking the second row of Figure 5 as an example, RGB deformation cannot focus on the detected chair because its color is similar to the background. However, depth deformation can easily focus on the chair due to the depth gap between the chair and background. Hallucinated deformation is similar to depth deformation and fits the shape of the object. In conclusion, depth deformation offers additional complementary geometric information to RGB images, so that our method can improve the performance by hallucinating depth information with RGB input.

Fusion Strategy Analyses

There are mainly two aspects for feature fusion: fusion location and fusion operator. In our case, we explore the fusion of features from R-branch and H-branch for various fusion locations and various fusion operators. Due to space limitation, the results and analyses are presented in Supplementary.

Evaluation on RGB-Only Dataset

Following (Hoffman, Gupta, and Darrell 2016), we investigate the potential of the proposed method on standard object detection dataset PASCAL VOC (Everingham et al. 2010), which is not associated with depth images. The details are left to the Supplementary.

Conclusion

In this work, we have proposed a novel deformation hallucination framework. The presented framework could hallucinate the depth deformation using RGB input, which complements the RGB deformation for ease of object detection. Experiments on NYUDv2 and SUN RGB-D datasets have demonstrated that our method outperforms state-of-the-art for depth privileged object detection.

Acknowledgements

The work is supported by the National Key R&D Program of China (2018AAA0100704) and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and is partially sponsored by National Natural Science Foundation of China (Grant No.61902247) and the Shanghai Science and Technology R&D Program of China (20511100300).

References

Blum, M.; Springenberg, J. T.; Wülfing, J.; and Riedmiller, M. 2012. A learned feature descriptor for object recognition in RGB-D data. In *IEEE International Conference on Robotics and Automation*, 1298–1303.

Bo, L.; Lai, K.; Ren, X.; and Fox, D. 2011. Object recognition with hierarchical kernel descriptors. In *Proceedings of the IEEE conference on Cmputer Vision and Pattern Recognition*, 1729–1736.

Cadena, C.; and Košecka, J. 2015. Semantic parsing for priming object detection in indoors RGB-D scenes. *The International Journal of Robotics Research* 34(4-5): 582–597.

Cao, Y.; Shen, C.; and Shen, H. T. 2016. Exploiting depth from single monocular images for object detection and semantic segmentation. *IEEE Transactions on Image Processing* 26(2): 836–846.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 764–773.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Do, T.; Do, T.-T.; Tran, H.; Tjiputra, E.; and Tran, Q. D. 2019. Compact trilinear interaction for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, 392–401.

Dou, Q.; Liu, Q.; Heng, P.; and Glocker, B. 2020. Unpaired multi-modal segmentation via knowledge distillation. *IEEE Transactions on Medical Imaging* .

Du, D.; Wang, L.; Wang, H.; Zhao, K.; and Wu, G. 2019. Translate-to-recognize networks for rgb-d scene recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 11836–11845.

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

Feyereisl, J.; Kwak, S.; Son, J.; and Han, B. 2014. Object localization based on structural SVM using privileged information. In *Advances in Neural Information Processing Systems*, 208–216.

Gan, C.; Zhao, H.; Chen, P.; Cox, D.; and Torralba, A. 2019. Self-supervised moving vehicle tracking with stereo sound. In *Proceedings of the IEEE International Conference on Computer Vision*, 7053–7062.

Gupta, S.; Arbeláez, P.; Girshick, R.; and Malik, J. 2015. Indoor scene understanding with RGB-D images: Bottom-up segmentation, object detection and semantic segmentation. *International Journal of Computer Vision* 112(2): 133–149.

Gupta, S.; Girshick, R.; Arbeláez, P.; and Malik, J. 2014. Learning rich features from RGB-D images for object detection and segmentation. In *Proceedings of the European Conference on Computer Vision*, 345–360. Springer.

Gupta, S.; Hoffman, J.; and Malik, J. 2016. Cross modal distillation for supervision transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2827–2836.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *Computer Science* 14(7): 38–39.

Hoffman, J.; Gupta, S.; and Darrell, T. 2016. Learning with side information through modality hallucination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 826–834.

Hoffman, J.; Gupta, S.; Leong, J.; Guadarrama, S.; and Darrell, T. 2016. Cross-modal adaptation for RGB-D detection. In *IEEE International Conference on Robotics and Automation*, 5032–5039.

Kim, J.; Koh, J.; Kim, Y.; Choi, J.; Hwang, Y.; and Choi, J. W. 2018. Robust deep multi-modal learning based on gated information fusion network. In *Asian Conference on Computer Vision*, 90–106. Springer.

Lee, K.-H.; Ros, G.; Li, J.; and Gaidon, A. 2019. SPIGAN: Privileged adversarial learning from simulation. In *International Conference on Learning Representations*.

Li, G.; Gan, Y.; Wu, H.; Xiao, N.; and Lin, L. 2018. Cross-modal attentional context learning for RGB-D object detection. *IEEE Transactions on Image Processing* 28(4): 1591–1601.

Li, Q.; Jin, S.; and Yan, J. 2017. Mimicking very efficient network for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6356–6364.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2980–2988.

Liu, F.; Shen, C.; Lin, G.; and Reid, I. 2015. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Tansactions on Pattern Analysis and Machine Intelligence* 38(10): 2024–2039.

Marcacini, R. M.; and Rezende, S. O. 2013. Incremental hierarchical text clustering with privileged information. In *Proceedings of the ACM symposium on Document engineering*, 231–232.

Mordan, T.; Thome, N.; Henaff, G.; and Cord, M. 2018. Revisiting multi-task learning with ROCK: A deep residual

auxiliary block for visual detection. In *Advances in Neural Information Processing Systems*, 1310–1322.

Pan, B.; Wang, S.; and Jiang, Q. 2019. Image aesthetic assessment assisted by attributes through adversarial learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 679–686.

Rahman, M. M.; Tan, Y.; Xue, J.; Shao, L.; and Lu, K. 2019. 3D object detection: Learning 3D bounding boxes from scaled down 2D bounding boxes in RGB-D images. *Information Sciences* 476: 147–158.

Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; and Savarese, S. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 658–666.

Shen, C.; Xue, M.; Wang, X.; Song, J.; Sun, L.; and Song, M. 2019. Customizing student networks from heterogeneous teachers via adaptive knowledge amalgamation. In *Proceedings of the IEEE International Conference on Computer Vision*, 3504–3513.

Silberman, N.; Hoiem, D.; Kohli, P.; and Fergus, R. 2012. Indoor segmentation and support inference from RGBD images. In *Proceedings of the European Conference on Computer Vision*, 746–760.

Song, S.; Lichtenberg, S. P.; and Xiao, J. 2015. Sun RGB-D: A RGB-D scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 567–576.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. FCOS: Fully convolutional one-stage object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 9627–9636.

Vapnik, V.; and Vashist, A. 2009. A new learning paradigm: Learning using privileged information. In *International Joint Conference on Neural Networks*, volume 22, 544–557.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1492–1500.

Xu, T.-B.; and Liu, C.-L. 2019. Data-distortion guided self-distillation for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 5565–5572.

Xu, X.; Li, Y.; Wu, G.; and Luo, J. 2017. Multi-modal deep feature learning for RGB-D object detection. *Pattern Recognition* 72: 300–313.

Yan, Y.; Nie, F.; Li, W.; Gao, C.; Yang, Y.; and Xu, D. 2016. Image classification by cross-media active learning with privileged information. *IEEE Transactions on Multimedia* 18(12): 2494–2502.

Yang, C.; Xie, L.; Su, C.; and Yuille, A. L. 2019. Snapshot distillation: Teacher-student optimization in one generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2859–2868.

Ye, E. S.; and Malik, J. 2013. Object detection in RGB-D indoor scenes. *Technical Report of University of California at Berkeley* .

Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; and Li, S. Z. 2020. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9759–9768.

Zhang, Y.; Xiang, T.; Hospedales, T. M.; and Lu, H. 2018. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4320–4328.

Zhao, M.; Li, T.; Abu Alsheikh, M.; Tian, Y.; Zhao, H.; Torralba, A.; and Katabi, D. 2018. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7356–7365.

Zhu, X.; Hu, H.; Lin, S.; and Dai, J. 2019. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9308–9316.