Cylindrical Convolutional Networks for Joint Object Detection and Viewpoint Estimation

Sunghun Joung[1], Seungryong Kim[2,3], Hanjae Kim[1], Minsu Kim[1],
Ig-Jae Kim[4], Junghyun Cho[4], and Kwanghoon Sohn[1,*]
[1]Yonsei University [2]École Polytechnique Fédérale de Lausanne (EPFL)
[3]Korea University [4]Korea Institute of Science and Technology (KIST)
{sunghunjoung,incohjk,minsukim320,khsohn}@yonsei.ac.kr
seungryong_kim@korea.ac.kr, {drjay,jhcho}@kist.re.kr

Abstract

Existing techniques to encode spatial invariance within deep convolutional neural networks only model 2D transformation fields. This does not account for the fact that objects in a 2D space are a projection of 3D ones, and thus they have limited ability to severe object viewpoint changes. To overcome this limitation, we introduce a learnable module, cylindrical convolutional networks (CCNs), that exploit cylindrical representation of a convolutional kernel defined in the 3D space. CCNs extract a view-specific feature through a view-specific convolutional kernel to predict object category scores at each viewpoint. With the view-specific feature, we simultaneously determine objective category and viewpoints using the proposed sinusoidal soft-argmax module. Our experiments demonstrate the effectiveness of the cylindrical convolutional networks on joint object detection and viewpoint estimation.

1. Introduction

Recent significant success on visual recognition, such as image classification [33], semantic segmentation [24], object detection [12], and instance segmentation [13], has been achieved by the advent of deep convolutional neural networks (CNNs). Their capability of handling geometric transformations mostly comes from the extensive data augmentation and the large model capacity [19, 15, 31], having limited ability to deal with severe geometric variations, e.g., object scale, viewpoints and part deformations. To realize this, several modules have been proposed to explicitly handle geometric deformations. Formally, they transform the input data by modeling spatial transformation [16, 3, 20], e.g., affine transformation, or by learning the offset of sampling locations in the convolutional operators [42, 4]. However, all of these works only use a visible feature to handle geometric deformation in the 2D space, while viewpoint variations occur in the 3D space.

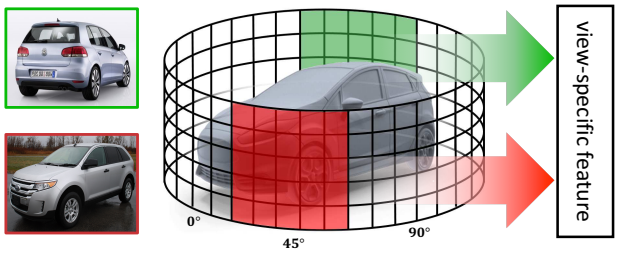

Figure 1. Illustration of cylindrical convolutional networks (CCNs) : Given a single image of objects, we apply a view-specific convolutional kernel to extract the shape characteristic of object from different viewpoints.

To solve the problems of viewpoint variations, joint object detection and viewpoint estimation using CNNs [36, 35, 26, 6] has recently attracted the interest. This involves first estimating the location and category of objects in an image, and then predicting the relative rigid transformation between the camera coordinate in the 3D space and each image coordinate in the 2D space. However, category classification and viewpoint estimation problems are inherently contradictory, since the former requires a *view-invariant* feature representation while the latter requires a *view-specific* feature representation. Therefore, incorporating viewpoint estimation networks to a conventional object detector in a multi-task fashion does not help each other, as demonstrated in several works [26, 7].

Recent studies on 3D object recognition have shown that object viewpoint information can improve the recognition performance. Typically, they first represent a 3D object

This research was supported by R&D program for Advanced Integrated-intelligence for Identification (AIID) through the National Research Foundation of KOREA (NRF) funded by Ministry of Science and ICT (NRF-2018M3E3A1057289).
*Corresponding author

with a set of 2D rendered images, extract the features of each image from different viewpoints, and then aggregate them for object category classification [34, 1, 37]. By using multiple features with a set of predefined viewpoints, they effectively model shape deformations with respect to the viewpoints. However, in real-world scenarios, they are not applicable because we cannot access the invisible side of an object without 3D model.

In this paper, we propose cylindrical convolutional networks (CCNs) for extracting view-specific features and using them to estimate object categories and viewpoints simultaneously, unlike conventional methods that share representation of feature for both object category [30, 23, 21] and viewpoint estimation [35, 26, 6]. As illustrated in Fig. 1, the key idea is to extract the view-specific feature conditioned on the object viewpoint (i.e., azimuth) that encodes structural information at each viewpoint as in 3D object recognition methods [34, 1, 37]. In addition, we present a new and differentiable argmax operator called sinusoidal soft-argmax that can manage sinusoidal properties of the viewpoint to predict continuous values from the discretized viewpoint bins. We demonstrate the effectiveness of the proposed cylindrical convolutional networks on joint object detection and viewpoint estimation task, achieving large improvements on Pascal 3D+ [41] and KITTI [10] datasets.

2. Related Work

2D Geometric Invariance. Most conventional methods for visual recognition using CNNs [33, 12, 24] provided limited performance due to geometric variations. To deal with geometric variations within CNNs, spatial transformer networks (STNs) [16] offered a way to provide geometric invariance by warping features through a global transformation. Lin and Lucey [20] proposed inverse compositional STNs that replace the feature warping with transformation parameter propagation, but it has a limited capability of handling local transformations. Therefore, several methods have been introduced by applying convolutional STNs for each location [3], estimating locally-varying geometric fields [42], and estimating spatial transformation in a recursive manner [18]. Furthermore, to handle adaptive determination of scales or receptive field for visual recognition with fine localization, Dai *et al*. [4] introduced two new modules, namely, deformable convolution and deformable ROI pooling that can model geometric transformation for each object. As all of these techniques model geometric deformation in the projected 2D image only with visible appearance feature, there is a lack of robustness to viewpoint variation, and they still only rely on extensive data augmentation.

Joint Category and Viewpoint Estimation. Since viewpoint of 3D object is a continuous quantity, a natural way to estimate it is to setup a viewpoint regression problem. Wang *et al*. [38] tried to directly regress viewpoint to manage the periodic characteristic with a mean square loss. However, the regression approach cannot represent the ambiguities well that exist between different viewpoints of objects with symmetries or near symmetries [26]. Thus, other works [36, 35] divide the angles into non-overlapping bins and solve the prediction of viewpoint as a classification problem, while relying on object localization using conventional methods (i.e. Fast R-CNN [11]). Divon and Tal [6] further proposed a unified framework that combines the task of object localization, categorization, and viewpoint estimation. However, all of these methods focus on accurate viewpoint prediction, which does not play a role in improving object detection performance [26].

Another main issue is a scarcity of real images with accurate viewpoint annotation, due to the high cost of manual annotation. Pascal 3D+ [41], the largest 3D image dataset still is limited in scale compare to object classification datasets (e.g. ImageNet [5]). Therefore, several methods [35, 38, 6] tried to solve this problem by rendering 3D CAD models [2] into background images, but they are unrealistic and do not match real image statistics, which can lead to domain discrepancy.

3D Object Recognition. There have been several attempts to recognize 3D shapes from a collection of their rendered views on 2D images. Su *et al*. [34] first proposed multi-view CNNs, which project a 3D object into multiple views and extract view-specific features through CNNs to use informative views by max-pooling. GIFT [1] also extracted view-specific features, but instead of pooling them, it obtained the similarity between two 3D objects by view-wise matching. Several methods to improve performance have been proposed, by recurrently clustering the views into multiple sets [37] or aggregating local features through bilinear pooling [43]. Kanezaki *et al*. [17] further proposed RotationNet, which takes multi view images as an input and jointly estimates object's category and viewpoint. It treats the viewpoint labels as latent variables, enabling usage of only a partial set of multi-view images for both training and testing.

3. Proposed Method

3.1. Problem Statement and Motivation

Given a single image of objects, our objective is to jointly estimate object category and viewpoint to model viewpoint variation of each object in the 2D space. Let us denote N_c as the number of object classes, where the class C is determined from each benchmark and N_v is determined by the number of discretized viewpoint bins. In particular, since the variation of elevation and tilt is small on real-scenes [41], we focus on estimation of the azimuth.

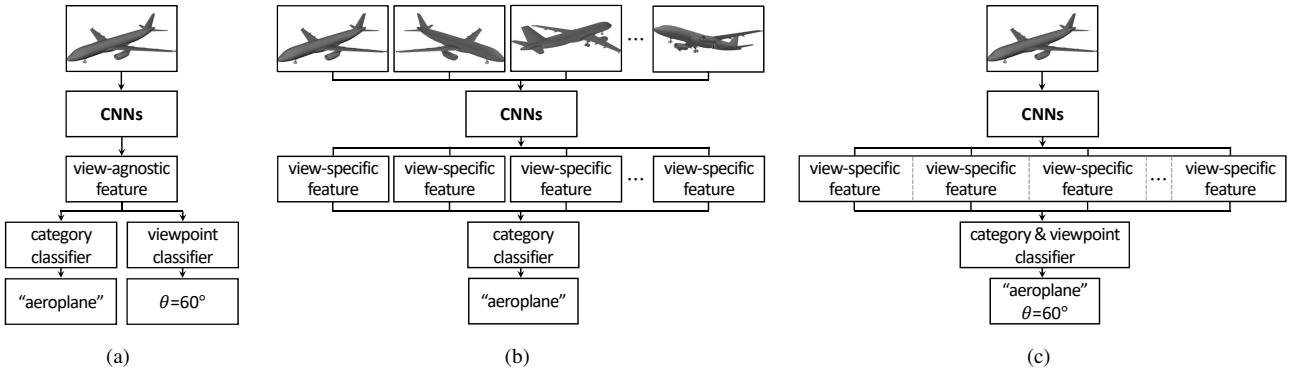

Figure 2. Intuition of cylindrical convolutional networks: (a) joint category and viewpoint estimation methods [26, 6] using single-view image as an input, (b) 3D object recognition methods [34, 1] using multi-view image as an input, and (c) cylindrical convolutional networks, which take the advantages of 3D object recognition methods by extracting view-specific features from single-view image as an input.

Object categorization requires a view-agnostic representation of an input so as to recognize the object category regardless of viewpoint variations. In contrast, viewpoint estimation requires a representation that preserves shape characteristic of the object in order to distinguish their viewpoint. Conventional CNNs based methods [26, 6] extract a *view-agnostic* feature, followed by task-specific subnetworks, i.e., object categorization and viewpoint estimation, as shown in Fig. 2 (a). They, however, do not leverage the complementary characteristics of the two tasks, thus showing a limited performance. Unlike these methods, some methods on 3D object recognition have shown that *view-specific* features for each viewpoint can encode structural information [34, 1], and thus they use these feature to facilitate the object categorization task as shown in Fig. 2 (b). Since they require multi-view images of pre-defined viewpoints, their applicability is limited to 3D object recognition (i.e. ModelNet 40 [39]).

To extract the *view-specific* features from a single image, we present cylindrical convolutional networks that exploit a cylindrical convolutionial kernel, where each subset is a view-specific kernel to capture structural information at each viewpoint. By utilizing view-specific feature followed by object classifiers, we estimate an object category likelihood at each viewpoint and select a viewpoint kernel that predicts to maximize object categorization probability.

3.2. Cylindrical Convolutional Networks

Let us denote an intermediate CNN feature map of Region of Interest (ROI) [13] as $\mathrm{x} \in \mathbb{R}^{k \times k \times \mathrm{ch}_i}$, with spatial resolution $k \times k$ and ch_i channels. Conventional viewpoint estimation methods [26, 6] apply a $k \times k$ view-agnostic convolutional kernel in order to preserve position sensitive information for extracting feature $F \in \mathbb{R}^{\mathrm{ch}_o}$, where ch_o is the number of output channels. Since the structural information of projected images varies with different viewpoints, we aim to apply a view-specific convolutional kernel at a predefined set of N_v viewpoints. The most straightforward way for realizing this is to define N_v variants of $k \times k$ kernel. This strategy, however, cannot consider structural similarity between nearby viewpoints, and would be inefficient.

We instead model a cylindrical convolutional kernel with weight parameters $W_{\mathrm{cyl.}} \in \mathbb{R}^{k \times N_v \times \mathrm{ch}_i \times \mathrm{ch}_o}$ as illustrated in Fig. 3. Each $k \times k$ kernel extracted along horizontal axis on $W_{\mathrm{cyl.}}$ in a sliding window fashion can be seen as a view-specific kernel W_{v}. We then obtain N_v variants of a view-specific feature $F_v \in \mathbb{R}^{\mathrm{ch}_o}$ as

$$F_v = \sum_{\mathrm{p} \in \mathcal{R}} W_{\mathrm{v}}\left(\mathrm{p}\right) \cdot \mathrm{x}\left(\mathrm{p}\right) = \sum_{\mathrm{p} \in \mathcal{R}} W_{\mathrm{cyl.}}\left(\mathrm{p} + \mathrm{o}_v\right) \cdot \mathrm{x}\left(\mathrm{p}\right), \tag{1}$$

where o_v is an offset on cylindrical kernel $W_{\mathrm{cyl.}}$ for each viewpoint v. The position p varies within in the $k \times k$ window $\mathcal{R}$. Different from view-specific features on Fig. 2 (b) extracted from multi-view images, our view-specific feature benefit from structural similarity between nearby viewpoints. Therefore, each view-specific kernel can be trained to discriminate shape variation from different viewpoints.

3.3. Joint Category and Viewpoint Estimation

In this section, we propose a framework to jointly estimate object category and viewpoint using the view-specific features F_v. We design convolutional layers $f\left(\cdot\right)$ with parameters W_{cls} to produce $N_v \times \left(N_c + 1\right)$ score map such that $S_{v,c} = f\left(F_v; W_{\mathrm{cls}}\right)$. Since each element of $S_{v,c}$ represents the probability of object belong to each category c and viewpoint v, the category and viewpoint can be predicted by just finding the maximum score from $S_{v,c}$. However, it is not differentiable along viewpoint distribution, and only predicts discretized viewpoints. Instead, we propose sinusoidal soft-argmax function, enabling the network to predict continuous viewpoints with periodic properties. To obtain the probability distribution, we normalize $S_{v,c}$ across the viewpoint axis with a softmax operation $\sigma\left(\cdot\right)$ such that

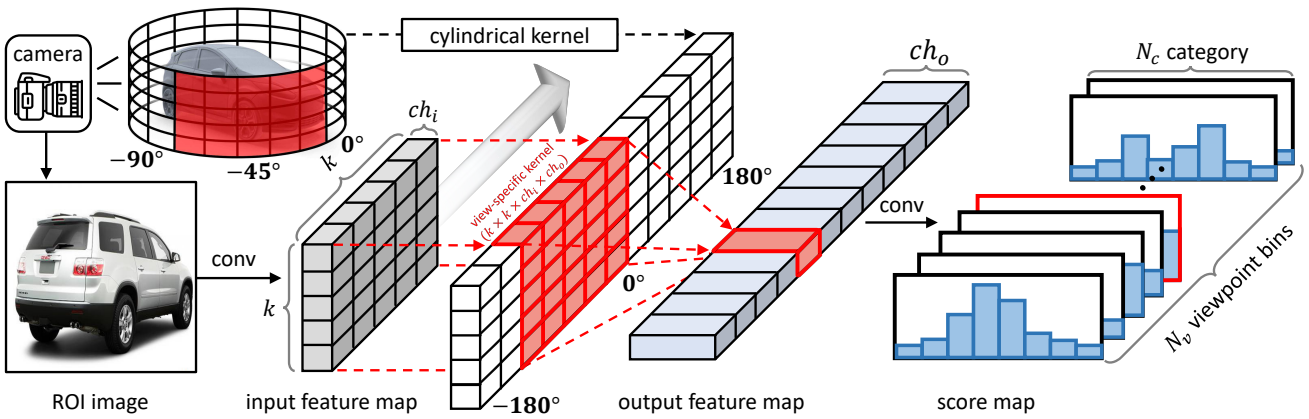

Figure 3. Key idea of cylindrical convolutional networks. Input feature maps from fully convolutional networks are fed into the cylindrical convolutional kernel to obtain N_v variants of view-specific feature. Then, each view-specific feature is used to identify its category likelihood that object category classification and viewpoint estimation can be jointly estimated.

$P_{v,c} = \sigma(S_{v,c})$. In the following, we describe how we estimate object categories and viewpoints.

Category Classification. We compute the final category classification score using a weighted sum of category likelihood for each viewpoint, $S_{v,c}$, with viewpoint probability distribution, $P_{v,c}$, as follows:

$$S_c = \sum_{v=1}^{N_v} S_{v,c} \cdot P_{v,c}, \tag{2}$$

where S_c represents an final classification score along category c. Since the category classification is essentially viewpoint invariant, the gradient from S_c will emphasize correct viewpoint's probability, while suppressing others as attention mechanism [16]. It enables the back-propagation of supervisory signal along N_v viewpoints.

Viewpoint Estimation. Perhaps the most straightforward way to estimate a viewpoint within CCNs is to choose the best performing view-specific feature from predefined viewpoints to identify object category. In order to predict the continuous viewpoint with periodic properties, we further introduce a sinusoidal soft-argmax, enabling regression from $P_{v,c}$ as shown in Fig. 4.

Specifically, we make use of two representative indices, $\sin(i_v)$ and $\cos(i_v)$, extracted by applying sinusoidal function to each viewpoint bin i_v (i.e. 0°, 15°,... for $N_v = 24$). We then take sum of each representative index with its probability, followed by atan2 function to predict object viewpoint for each class c as follows:

$$\theta_c = \text{atan2}\left(\sum_{v=1}^{N_v} P_{v,c} \sin(i_v), \sum_{v=1}^{N_v} P_{v,c} \cos(i_v)\right), \tag{3}$$

which takes advantage of classification-based approaches [36, 35] to estimate posterior probabilities, enabling better training of deep networks, while considering the periodic characteristic of viewpoints as regression-based approaches [38]. The final viewpoint estimation selects θ_c with corresponding class c through category classification (2).

Bounding Box Regression. To estimate fine-detailed location, we apply additional convolutional layers for bounding box regression with W_{reg} to produce $N_v \times N_c \times 4$ bounding box offsets, denoted as $t_{v,c} = f(F_v; W_{\text{reg}})$. Each set of 4 values encodes bounding box transformation parameters [12] from initial location for one of the $N_v \times N_c$ sets. This leads to use different sets of boxes for each category and viewpoint bin, which can be shown as an extended version of class-specific bounding box regression [11, 30].

Loss Functions. Our total loss function defined on each feature is the summation of classification loss L_{cls}, bounding box regression loss L_{reg}, and viewpoint estimation loss L_{view} as follows:

$$L = L_{\text{cls}}(c, \hat{c}) + [\hat{c} \geq 1]\{L_{\text{reg}}(t_{v,c}, \hat{t}) + [\hat{\theta} \neq \emptyset] L_{\text{view}}(\theta_c, \hat{\theta})\}, \tag{4}$$

using ground-truth object category $\hat{c}$, bounding box regression target $\hat{t}$ and viewpoint $\hat{\theta}$. Iverson bracket indicator function $[\cdot]$ evaluates to 1 when it is true and 0 otherwise. For background, $\hat{c} = 0$, there is no ground-truth bounding box and viewpoint, hence L_{reg} and L_{view} are ignored. We train the viewpoint loss L_{view} in a semi-supervised manner, using the sets with ground-truth viewpoint ($\hat{\theta} \neq \emptyset$) for supervised learning. For the datasets without viewpoint annotation ($\hat{\theta} = \emptyset$), L_{view} is ignored and viewpoint estimation task is trained in an unsupervised manner. We use cross-entropy for L_{cls}, and smooth L1 for both L_{reg} and L_{view}, following conventional works [11, 30].

3.4. Implementation and Training Details

For cylindrical kernel $W_{\text{cyl.}}$, we apply additional constraint to preserve a reflectinoal symmetry of 3D ob-

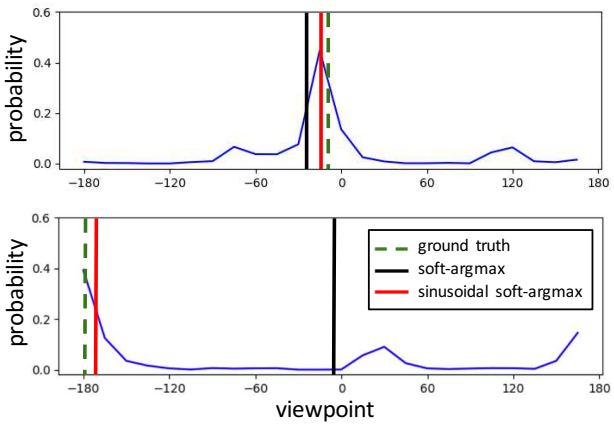

Figure 4. Illustration of sinusoidal soft-argmax: for probability distribution of discretized viewpoint bins, sinusoidal soft-argmax enables to regress periodic viewpoint signal, while conventional soft-argmax cannot be applied.

jects. We first divide the parameters into four groups as front, rear, left-side, and right-side, and make the parameters of left-side and the right-side to be reflective using horizontal flip operation $h(\cdot)$ such that $W_{\text{cyl.}} = [W_{\text{side}}, W_{\text{front}}, h(W_{\text{side}}), W_{\text{rear}}]$, where parameters of each groups are concatenated horizontally. We set the spatial resolution of W_{front} and W_{back} as $k \times 1$, and W_{side} as $k \times (N_v - 2)/2$. Therefore, $W_{\text{cyl.}}$ can preserve horizontal reflectional symmetry and saves the network memory.

In order to make $W_{\text{cyl.}}$ defined on a 3D space to be implemented in a 2D space, periodicity along the azimuth has to be preserved. Therefore, we horizontally pad $k \times \lfloor k/2 \rfloor$ of parameters from the left end to the right side using flip operation, and vice versa, where $\lfloor \cdot \rfloor$ denotes floor function that outputs the greatest integer less than or equal to input. It allows $W_{\text{cyl.}}$ to be used as periodic parameters.

We adopt two stage object detection framework, Faster R-CNN [30] that first processes the whole image by standard fully convolutional networks [15, 21], followed by Region Proposal Network (RPN) [30] to produce a set of bounding boxes. We then use ROI Align [13] layer to extract fixed size feature x for each Region of Interest (ROI).

In both training and inference, images are resized so that the shorter side is 800 pixels, using anchors of 5 scales and 3 aspect ratios with FPN, and 3 scales and 3 aspect ratios without FPN are utilized. 2k and 1k region proposals are generated using non-maximum suppression threshold of 0.7 at both training and inference respectively. We trained on 2 GPUs with 4 images per GPU (effective mini batch size of 8). The backbones of all models are pretrained on ImageNet classification [5], and additional parameters are randomly initialized using *He initialization* [14]. The learning rate is initialized to 0.02 with FPN, 0.002 without FPN, and decays by a factor of 10 at the 9th and 11th epochs. All models are trained for 12 epochs using SGD with a weight decay of 0.0001 and momentum of 0.9, respectively.

Method		Cateogory		Viewpoint	
N_v	CCNs	top-1	top-3	$\text{Acc}_{\pi/6}$	Mederr
24		0.91	0.97	0.56	23.5
18	✓	0.95	0.99	0.63	17.3
24	✓	**0.95**	**0.99**	**0.66**	**15.5**
30	✓	0.94	0.98	0.63	17.7

Table 1. Joint object category and viewpoint estimation performance with ground truth box on Pascal 3D+ dataset [41].

4. Experiments

4.1. Experimental Settings

Our experiments are mainly based on maskrcnn-benchmark [25] using PyTorch [27]. We use the standard configuration of Faster R-CNN [30] based on ResNet-101 [15] as a backbone. We implement two kinds of network, with and without using FPN [21]. For the network without using FPN, we remove the last pooling layer to preserve spatial information of each ROI feature. We set $k = 7$ following conventional works, and set $N_v = 24$ unless stated otherwise. The choice of other hyper-parameters keeps the same with the default settings in [25].

We evaluate our joint object detection and viewpoint estimation framework on the Pascal 3D+ [41] and KITTI dataset [10]. The Pascal 3D+ dataset [41] consists of images from Pascal VOC 2012 [8] and images of subset from ImageNet [5] for 12 different categories that are annotated with its viewpoint. Note that the bottle category is omitted, since it is often symmetric across different azimuth [41]. On the other hand, the KITTI dataset [10] consists of 7,481 training images and 7,518 test images that are annotated with its observation angle and 2D location. For KITTI dataset, we focused our experiment on the Car object category.

Pascal 3D+ dataset. In this experiment, we trained our network using the training set of Pascal 3D+ [41] (*training* set of Pascal VOC 2012 [8] and ImageNet [5]) for supervised learning only, denoted as CCNs, and semi-supervised learning with additional subset of *trainval35k* with overlapping classes of COCO dataset [22], denoted as CCNs*. The evaluation is done on the *val* set of Pascal 3D+ [41] using Average Precision (AP) metric [8] and Average Viewpoint Precision (AVP) [41], where we focus on AVP24 metric. Furthermore, we also evaluate our CCNs using *minival* split of COCO dataset [22] using COCO-style Average Precision (AP) @ $[0.5 : 0.95]$ and Average Recall (AR) metric [22] on objects of small, medium, and large sizes.

KITTI dataset. In this experiment, we followed *train/val* setting of Xiang *et al*. [40], which guarantees that images from the training and validation set are from different videos. For evaluation using KITTI dataset [10], we use Average Precision (AP) metric with 70% overlap threshold (AP@IOU0.7), and Average Orientation Similarity (AOS)

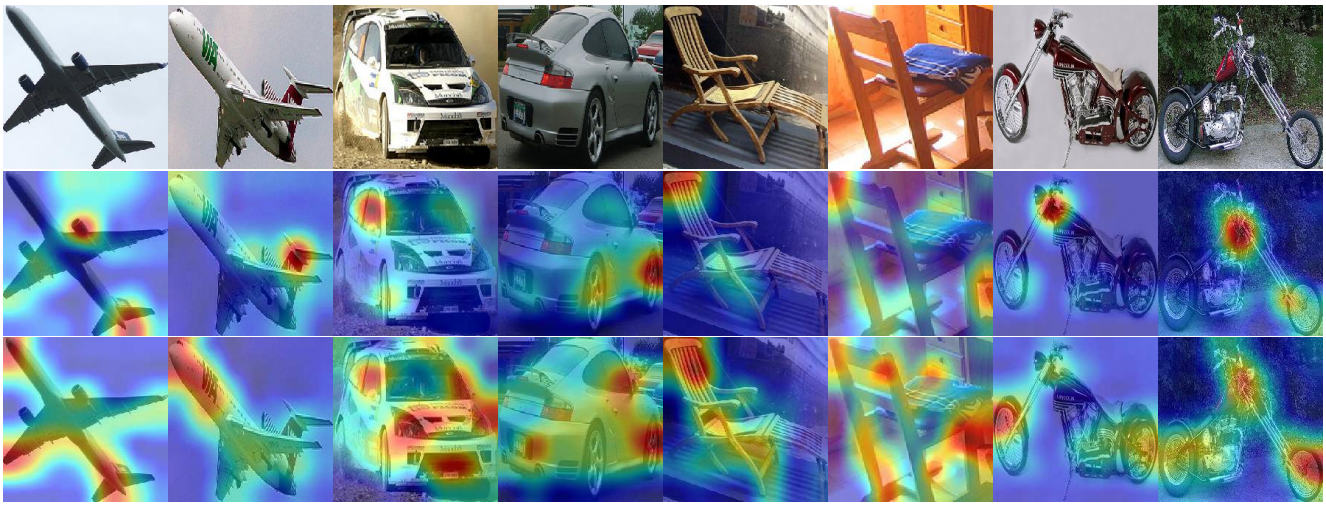

Figure 5. Visualization of learned deep feature through Grad-CAM [32]: (from top to bottom) inputs, attention maps trained without CCNs, and with CCNs. Note that red color indicates attentive regions and blue color indicates suppressed regions.

[10]. Results are evaluated based on three levels of difficulty: Easy, Moderate, and Hard, which are defined according to the minimum bounding box height, occlusion, and truncation grade.

4.2. Ablation Study

Analysis of the CCNs components. We analyzed our CCNs with the ablation evaluations with respect to various setting of N_v and the effectiveness of the proposed view-specific convolutional kernel. In order to evaluate performance independent of factors such as mis-localization, we tackle the problem of joint category classification and viewpoint estimation with ground-truth bounding box using ResNet-101 [15]. For a fair comparison, a $k \times k$ view-agnostic convolutional kernels are implemented for joint object category classification and viewpoint estimation, which outputs $N_c \times N_v$ score map following conventional work [6]. In order to compare the viewpoint estimation accurately, we applied sinusoidal soft-argmax to regress the continuous viewpoint. We evaluated the top-1 and top-3 error rates for object category classification performance, and use median error (MedErr) and $\mathrm{Acc}_{\pi/6}$ for viewpoint estimation performance [36].

As shown in Table 1, CCNs have shown better performance in both object category classification and viewpoint estimation compared to the conventional method using view-agnostic kernel. The result shows that view-specific kernel effectively leverage the complementary characteristics of the two tasks. Since the result with $N_v = 24$ has shown the best performance in both category classification and viewpoint estimation, we set $N_v = 24$ for remaining experiments. Note that the number of parameters in cylindrical kernel is $k \times \{(N_v - 2)/2 + 2\} \times ch_i = 7 \times 13 \times ch_i$, while the baseline uses $k \times k \times ch_i = 7 \times 7 \times ch_i$. The number of additional parameters is marginal ($\sim 0.01\%$) compared to the total number of network parameters, while performance is significantly improved.

Network visualization. For the qualitative analysis, we applied the Grad-CAM [32] to visualize attention maps based on gradients from output category predictions. We compared the visualization results of CCNs with view-specific kernel and baseline with view-agnostic kernel. In Fig. 5, the attention map of the CCNs covers the overall regions in target object, while conventional category classifier tends to focus on the discriminative part of an object. From the observations, we conjecture that the view-specific convolutional kernel leads the network to capture the shape characteristic of object viewpoint.

4.3. Results

Pascal 3D+ dataset. In the following, we evaluated our CCNs and CCNs* in comparison to the state-of-the-art methods. Object detection methods are compared such as DPM [9], RCNN [12], Faster R-CNN [30] with ResNet-101 [15] and FPN [21]. Joint object detection and viewpoint estimation methods are also compared, including hand-crafted modules such as VDPM [41], DPM-VOC+VP [28], methods using off-the-shelf 2D object detectors for viewpoint estimation such as Su *et al*. [35], Tulsani and Malik [36], Massa *et al*. [26], and unified methods such as Poirson *et al*. [29], Divon and Tal [6].

As shown in Table 2 and Table 3, our CCNs* with FPN [21] outperformed conventional methods in terms of both object detection (mAP) and joint object detection and viewpoint estimation (mAVP) on Pascal 3D+ dataset [41]. It is noticeable that conventional methods for joint object detection and viewpoint estimation actually lowered the classification performance at [26], while ours improved the per-

Method	aero	bike	boat	bus	car	chair	dtable	mbike	sofa	train	tv	mAP
DPM [9]	42.2	49.6	6.0	54.1	38.3	15.0	9.0	33.1	18.9	36.4	33.2	29.6
VDPM [41]	42.2	44.4	6.0	53.7	36.3	12.6	11.1	35.5	17.0	32.6	33.6	29.5
DPM-VOC+VP [28]	36.0	45.9	5.3	53.9	42.1	8.0	5.4	34.8	11.0	28.2	27.3	27.1
RCNN [12]	72.4	68.7	34.0	73.0	62.3	33.0	35.2	70.7	49.6	70.1	57.2	56.9
Massa *et al*. [26]	77.1	70.4	51.0	77.4	63.0	24.7	44.6	76.9	51.9	76.2	64.6	61.6
Poirson *et al*. [29]	76.6	67.7	42.7	76.1	59.7	15.5	51.7	73.6	50.6	77.7	60.7	59.3
Faster R-CNN w/ [15]	79.8	78.6	64.4	79.6	75.9	48.2	51.9	80.5	49.8	77.9	79.2	69.6
Faster R-CNN w/ [21]	82.7	78.3	**71.8**	78.7	76.0	**50.8**	**53.3**	83.3	50.7	82.6	77.2	71.4
CCNs w/ [15]	82.5	79.2	64.4	80.3	76.7	49.4	50.9	81.4	48.2	79.5	78.9	70.2
CCNs* w/ [15]	82.9	81.4	63.7	86.6	79.7	43.6	51.7	81.6	52.5	81.0	82.1	71.5
CCNs w/ [21]	82.6	80.6	69.3	84.9	78.8	50.9	50.7	83.4	50.3	82.2	80.0	72.2
CCNs* w/ [21]	**83.7**	**82.8**	71.4	**88.1**	**81.2**	46.3	51.1	**85.9**	**52.7**	**83.8**	**84.0**	**73.7**

Table 2. Comparison of object detection on Pascal 3D+ dataset [41]. Average Precision (AP) @IOU 0.5 is evaluated.

Method	aero	bike	boat	bus	car	chair	dtable	mbike	sofa	train	tv	mAVP24
VDPM [41]	8.0	14.3	0.3	39.2	13.7	4.4	3.6	10.1	8.2	20.0	11.2	12.1
DPM-VOC+VP [28]	9.7	16.7	2.2	42.1	24.6	4.2	2.1	10.5	4.1	20.7	12.9	13.6
Su *et al*. [35]	21.5	22.0	4.1	38.6	25.5	7.4	11.0	24.4	15.0	28.0	19.8	19.8
Tulsani & Malik [36]	37.0	33.4	10.0	54.1	40.0	17.5	19.9	34.3	28.9	43.9	22.7	31.1
Massa *et al*. [26]	43.2	39.4	16.8	61.0	44.2	13.5	29.4	37.5	33.5	46.6	32.5	36.1
Poirson *et al*. [29]	33.4	29.4	9.2	54.7	35.7	5.5	23.0	30.3	27.6	44.1	34.3	28.8
Divon & Tal [6]	**46.6**	41.1	23.9	72.6	53.5	**22.5**	**42.6**	42.0	**44.2**	54.6	44.8	44.4
CCNs w/ [15]	39.0	45.9	22.6	74.5	54.7	19.6	38.9	44.2	41.5	55.3	46.8	43.9
CCNs* w/ [15]	39.4	47.0	23.2	76.6	55.5	20.3	39.5	44.5	41.8	56.1	45.5	44.5
CCNs w/ [21]	45.1	47.4	23.1	77.8	55.2	19.9	39.6	45.3	43.4	58.0	47.8	45.7
CCNs* w/ [21]	46.1	**48.8**	**24.2**	**78.0**	**55.9**	20.9	41.0	**45.3**	43.7	**59.5**	**49.0**	**46.6**

Table 3. Comparison of joint object detection and viewpoint estimation on Pascal 3D+ dataset [41]. Average Precision with 24 discretized viewpoint bins (AVP24) is evaluated, where true positive stands with correct bounding box localization and viewpoint estimation.

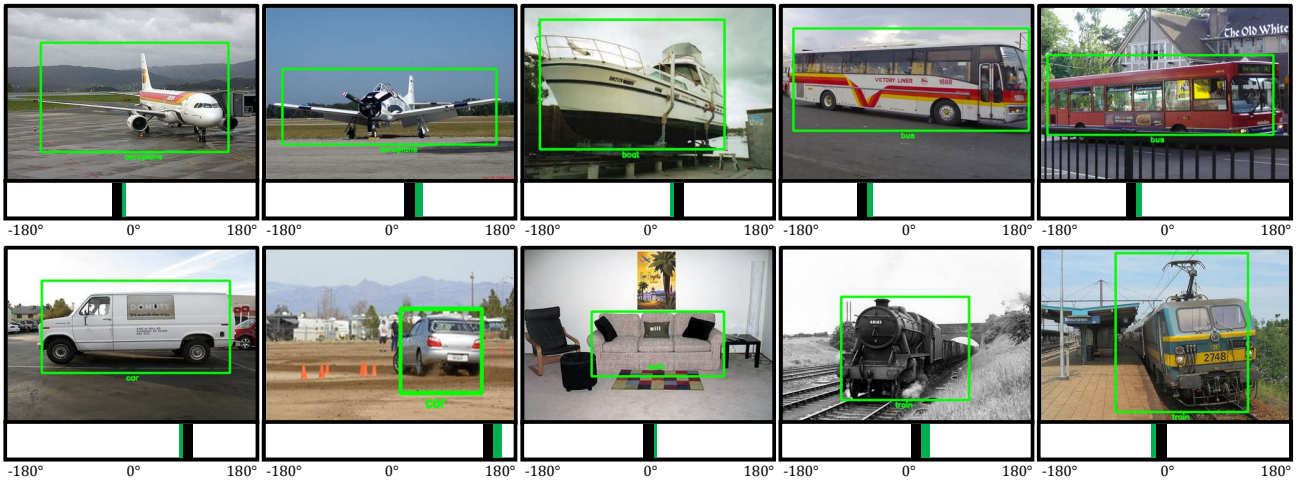

Figure 6. Qualitative examples of joint object detection and viewpoint estimation on Pascal3D+ dataset [41]. The bar below each image indicates the viewpoint prediction, in green, and the ground-truth in black.

formance compared to the original Faster R-CNN [30, 21]. Furthermore, our semi-supervised learning scheme using real datasets [22] shows performance improvement, indicating that (2) enables the supervisory signal for viewpoint estimation. Note that other viewpoint estimation methods used synthetic images with ground-truth viewpoint annotation [35, 26, 6] or keypoint annotation [36]. In Fig. 6, we show the examples of our joint object detection and view-

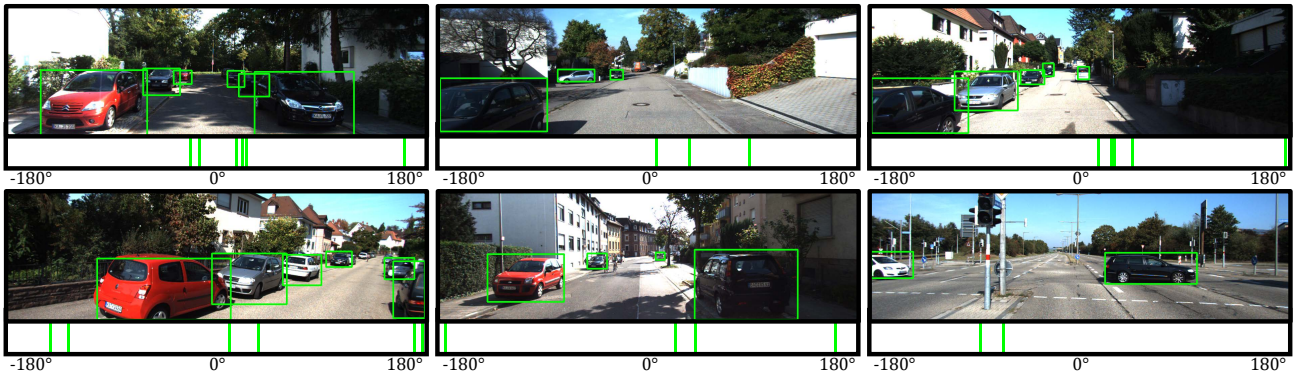

Figure 7. Qualitative examples of joint object detection and viewpoint estimation on KITTI dataset [10]. The bar below each image indicates the viewpoint prediction of corresponding object in green.

Metric	Network	CCNs	All	S	M	L
AP	ResNet [15]		34.3	15.5	28.9	47.3
		✓	**36.6**	**17.5**	**30.2**	**49.6**
	FPN [21]		40.7	22.1	36.2	52.1
		✓	**41.8**	**24.2**	**38.8**	**52.9**
AR	ResNet [15]		47.2	21.6	42.9	63.6
		✓	**49.6**	**22.7**	**44.1**	**66.0**
	FPN [21]		54.1	32.6	51.3	66.5
		✓	**56.3**	**33.9**	**53.1**	**68.3**

Table 4. Comparison of object detection on subset of COCO dataset [22]. The COCO-style Average Precision (AP) @IOU$\in [0.5, 0.95]$ and Average Recall (AR) are evaluated on objects of small (S), medium (M), and large (L) sizes.

Metric	Methods	Easy	Moderate	Hard
AP	Faster-RCNN [30]	82.97	77.83	66.25
	w/o CCNs	81.74	76.23	64.19
	CCNs	**86.17**	**80.19**	**67.14**
AOS	Faster-RCNN [30]	-	-	-
	w/o CCNs	79.46	72.92	59.63
	CCNs	**85.01**	**79.13**	**63.56**

Table 5. Comparison of joint object detection and viewpoint estimation on *val* set of KITTI dataset [10] for cars. Average Precision (AP) @IOU 0.7 is evaluated for object detection, and Average Orientation Similarity (AOS) for viewpoint estimation.

point estimation on Pascal 3D+ dataset [10].

Table 4 validates the effect of CCNs on the standard object detection dataset. Compared to the baseline without using CCNs, object detection performance (AP) has increased by applying view-specific convolutional kernel. Furthermore, the localization performance (AR) has also increased, indicating that our view-specific convolutional kernel can effectively encode structural information of input objects.

KITTI dataset. We further evaluated our CCNs in KITTI object detection benchmark [10]. Since the other methods aim to find 3D bounding boxes from monocular image, we conducted the experiment to validate the effectiveness of CCNs. As shown in Table 5, our CCNs have shown better results compare to original Faster-RCNN [30] by adapting view-specific convolutional kernel. On the other hand, joint training of object detection and viewpoint estimation without using CCNs actually lowered the object detection performance. This results share the same properties as previous studies [26, 7], indicating that proper modeling of geometric relationship is to be determined. In Fig. 7, we show the examples of our joint object detection and viewpoint estimation on KITTI dataset [10].

4.4. Discussion

Estimating the viewpoint of deformable categories is an open problem. We thus experimented our cylindrical convolutional networks for visual recognition on rigid categories only [41]. However, our key idea using view-specific convolutional kernel can be generalized with suitable modeling of deformable transformation (e.g., deformable convolution [4]) at the kernel space. We believe that the modeling pose or keypoint of non-rigid categories (e.g., human pose estimation) with our CCNs can be alternative to the current limitation, and leave it as future work.

5. Conclusion

We have introduced cylindrical convolutional networks (CCNs) for joint object detection and viewpoint estimation. The key idea is to exploit view-specific convolutional kernels, sampled from a cylindrical convolutional kernel in a sliding window fashion, to predict an object category likelihood at each viewpoint. With this likelihood, we simultaneously estimate object category and viewpoint using the proposed sinusoidal soft-argmax module, resulting state-of-the-art performance on the task of joint object detection and viewpoint estimation. In the future, we aim to extend view-specific convolutional kernel into non-rigid categories.

References

[1] Song Bai, Xiang Bai, Zhichao Zhou, Zhaoxiang Zhang, and Longin Jan Latecki. Gift: A real-time and scalable 3d shape search engine. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5023–5032, 2016. 2, 3

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. 2015. 2

[3] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In *Advances in Neural Information Processing Systems*, pages 2414–2422, 2016. 1, 2

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 764–773, 2017. 1, 2, 8

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2, 5

[6] Gilad Divon and Ayellet Tal. Viewpoint estimation—insights & model. In *Proceedings of the European Conference on Computer Vision*, pages 252–268, 2018. 1, 2, 3, 6, 7

[7] Mohamed Elhoseiny, Tarek El-Gaaly, Amr Bakry, and Ahmed Elgammal. A comparative analysis and study of multiview cnn models for joint object categorization and pose estimation. In *International Conference on Machine learning*, pages 888–897, 2016. 1, 8

[8] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015. 5

[9] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2009. 6, 7

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 2, 5, 6, 8

[11] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 2, 4

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. 1, 2, 4, 6, 7

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017. 1, 3, 5

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015. 5

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 5, 6, 7, 8

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015. 1, 2, 4

[17] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5010–5019, 2018. 2

[18] Seungryong Kim, Stephen Lin, Sang Ryul Jeon, Dongbo Min, and Kwanghoon Sohn. Recurrent transformer networks for semantic correspondence. In *Advances in Neural Information Processing Systems*, pages 6126–6136, 2018. 2

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 1

[20] Chen-Hsuan Lin and Simon Lucey. Inverse compositional spatial transformer networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2568–2576, 2017. 1, 2

[21] 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. 2, 5, 6, 7, 8

[22] 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 *Proceedings of the European Conference on Computer Vision*, pages 740–755, 2014. 5, 7, 8

[23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*, pages 21–37, 2016. 2

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1, 2

[25] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. `https://github.com/facebookresearch/maskrcnn-benchmark`, 2018. 5

[26] Francisco Massa, Renaud Marlet, and Mathieu Aubry. Crafting a multi-task cnn for viewpoint estimation. In *Proceedings*

of the British Machine Vision Conference, pages 91.1–91.12, 2016. 1, 2, 3, 6, 7, 8

[27] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[28] Bojan Pepik, Michael Stark, Peter Gehler, and Bernt Schiele. Teaching 3d geometry to deformable part models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3362–3369, 2012. 6, 7

[29] Patrick Poirson, Phil Ammirato, Cheng-Yang Fu, Wei Liu, Jana Kosecka, and Alexander C Berg. Fast single shot detection and pose estimation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 676–684, 2016. 6, 7

[30] 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. 2, 4, 5, 6, 7, 8

[31] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017. 1

[32] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017. 6

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv preprint arXiv:1409.1556*, 2014. 1, 2

[34] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–953, 2015. 2, 3

[35] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2686–2694, 2015. 1, 2, 4, 6, 7

[36] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1519, 2015. 1, 2, 4, 6, 7

[37] Chu Wang, Marcello Pelillo, and Kaleem Siddiqi. Dominant set clustering and pooling for multi-view 3d object recognition. In *Proceedings of the British Machine Vision Conference*, pages 61.4–61.12, 2017. 2

[38] Yumeng Wang, Shuyang Li, Mengyao Jia, and Wei Liang. Viewpoint estimation for objects with convolutional neural network trained on synthetic images. In *Pacific Rim Conference on Multimedia*, pages 169–179, 2016. 2, 4

[39] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 3

[40] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. Subcategory-aware convolutional neural networks for object proposals and detection. In *IEEE Winter Conference on Applications of Computer Vision*, pages 924–933, 2017. 5

[41] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, pages 75–82, 2014. 2, 5, 6, 7, 8

[42] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *Proceedings of the European Conference on Computer Vision*, pages 467–483, 2016. 1, 2

[43] Tan Yu, Jingjing Meng, and Junsong Yuan. Multi-view harmonized bilinear network for 3d object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 186–194, 2018. 2