Recognizing Objects from Any View with Object and Viewer-Centered Representations

Sainan Liu Vincent Nguyen Isaac Rehg Zhuowen Tu
University of California, San Diego
{sal131, vvn012, irehg, ztu}@ucsd.edu

Abstract

In this paper, we tackle an important task in computer vision: any view object recognition. In both training and testing, for each object instance, we are only given its 2D image viewed from an unknown angle. We propose a computational framework by designing object and viewer-centered neural networks (OVCNet) to recognize an object instance viewed from an arbitrary unknown angle. OVCNet consists of three branches that respectively implement object-centered, 3D viewer-centered, and in-plane viewer-centered recognition. We evaluate our proposed OVCNet using two metrics with unseen views from both seen and novel object instances. Experimental results demonstrate the advantages of OVCNet over classic 2D-image-based CNN classifiers, 3D-object (inferred from 2D image) classifiers, and competing multi-view based approaches. It gives rise to a viable and practical computing framework that combines both viewpoint-dependent and viewpoint-independent features for object recognition from any view.

1. Introduction

Objects are three-dimensional in the physical world, but the recognition tasks in computer vision have been primarily performed on 2D natural images [9]. Despite the great success of the deep convolutional neural networks (CNNs) [18, 43, 38, 14, 49], a standard CNN model that represents images in the 2D image space only tends to suffer from a "mental rotation" [36] like effect [3], as shown in Figure 3. Namely, when training a network with a limited number of views of an object instance, it may have a hard time recognizing the same object instance from an unseen viewpoint. There are two schools of thought regarding object representations. For biological vision systems, there has been a long-time debate [26] in cognitive psychology about whether objects are fundamentally encoded by *object-centered* or *viewer-centered* representations [44, 13]. In David Marr's pioneering vision paradigm [27], object recognition is carried out primarily in an object-centered manner in which objects are represented either by explicit 3D primitives (*e.g.* cylinders) [4] or by features that are invariant to viewpoint changes [2]. However, the theory of object-centered representation has been challenged in the past. Psychophysical and computational neural studies have shown evidence that viewer-centered representations [35, 25, 10] play a significant role in object recognition.

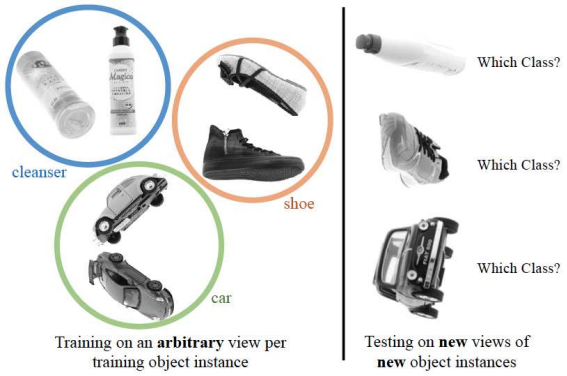


Figure 1. Problem illustration. Our task is to recognize an object from any view. In both training and testing, we only see 2D images without knowing the viewing angles and depth.

Implementations of both viewpoint-independent [20, 22] and viewpoint-dependent [3, 1] systems are present in computer and machine vision literature. An object-centered system typically encodes and stores a representation with viewpoint-independent (object-centric) features [17] that are invariant to viewpoint changes. During test time, representations with viewpoint-independent features are computed for a query object under a novel view to match with the stored features. A viewer-centered system instead stores a set of viewpoint-dependent features from typical viewing angles. During testing, a given view of an object instance is matched to the saved features to the specific viewpoints.

An object-centered representation has the advantage of maintaining rotation-invariant features that are insensitive to viewpoint changes; however, it relies on the presence of faithful 3D reconstructions or effective invariant features that are usually difficult to obtain from a single view image [13]. Conversely, a viewer-centered representation typically

stores features that are sensitive to the viewpoint changes; viewpoint-dependent features are usually straightforward to compute and learn.

Studies that combine both object-centered and viewer-centered representations also exist [28, 5, 29]. However, there has been limited success in the computer vision literature to build a hybrid system [19]. Additionally, systematic novel-view evaluation metrics are rarely used to evaluate the new state-of-the-art recognition systems.

Inspired by the theories of object-centered and viewer-centered object recognition [27, 26] as well as recent deep learning approaches for object recognition [40, 8], we propose a new algorithm: object and viewer-centered neural networks (OVCNet) for object recognition from any view. OVCNet has several attractive properties: 1) It adopts a pre-trained Generalizable Reconstruction (GenRe) model [51] to reconstruct 3D images from a single view image. We take advantage of the property of GenRe generalizing well to unseen object classes beyond the three classes ("plane", "car", and "chair") that it was trained on. Hence, we are able to infer the shape of a novel instance without additional object-specific 3D shape information. 2) OVCNet consists of three object recognition branches/modules by respectively implementing object-centric, 3D viewer-centric, and in-plane viewer-centric recognition to better perform the task. 3) We show that by adding sparse viewer-centered representations, we can further assist feature learning in the object-centered sub-module through spherical CNNs [8]. The resulting OVCNet is an integrated framework that learns viewpoint-independent and viewpoint-dependent features from an arbitrary view, and it can recognize novel views from both seen (familiar) and novel object instances.

In cognitive psychology, Marr initially proposed the definition [27] of **object-centered** and **viewer-centered** representation for object recognition. Since then, further interpretations are provided in [13, 26, 26, 44] emphasizing that a viewer-centered representation captures shapes at a particular view, whereas an object-centered representation represents the intrinsic 3D shape. Inspired by these cognitive psychology findings, we ask for the following properties for an *object-centered* module in our network design: 1) *3D model based* (*e.g.* volumetric, mesh, point-cloud or spherical maps); 2) *rotation invariant*; 3) *absent pose alignment*. Here, we characterize some of the methods [46, 31, 41, 16, 8] referred in this paper in Table 1. Although these individual approaches in comparison have their own merits, our experiments show that each method alone does not produce satisfactory recognition result on 3D-reconstruction derived from an arbitrary view image.

To evaluate OVCNet, we use a real object grayscale multi-view dataset [16], a virtual object grayscale multi-view dataset generated from ShapeNet [7], and a natural-colored dataset (a subset of the Pascal VOC dataset [12]). We split the views of different object instances into training and testing. In training, the dataset consists of one 2D image per object instance from an unspecified viewing angle; in testing, we perform classification on two sets of images from novel viewpoints of both seen (familiar) and novel object instances, respectively. Compared to a 2D image-based object recognition system such as AlexNet [18] and ResNet [14] as well as several 3D object recognition methods [8, 31, 46] following a single-view reconstruction module, OVCNet shows its clear advantage in the performance observed, especially on the relatively larger dataset, gMIVO. Furthermore, we also show that our algorithm outperforms standard ResNet18 by a large margin on a subset of Pascal VOC natural images.

In comparison with standard image classification tasks such as ImageNet [9], their metrics concern with generalization to novel instances, whereas our paradigm introduces generalization to novel views as well. Our contributions are listed as follows.

- We tackle the problem of object recognition from any view (single-arbitrary-view training and novel-view-novel-object-instance testing) by developing an algorithm that jointly encodes object-centered and viewer-centered representations.
- We create an object and viewer-centered network (OVCNet) with three branches, each specializing in either object-centered, viewer-centered (3D), or viewer-centered (2D) learning. The proposed OVCNet consists of a combination of spherical CNNs, ResNet, and attention structures.
- Between object-centered and viewer-centered 3D branches, we develop a new network structure that enables integrated learning of both object-centered and viewer-centered representations with a communicating pathway between the two.
- We provide a new multi-view dataset generated from a subset of models of ShapeNetCoreV2 3D models.

2. Related work

In this section, we briefly discuss the existing literature and methods related to object-centered and viewer-centered object recognition.

Method	3D model based	Rotation-invariant	No pose alignment
3DShapeNet [46]	✓		
PointNet [31]	✓	✓	
MVCNN [41]		✓	✓
RotationNet [16]		✓	
Spherical CNNs [8]	✓	✓	✓

Table 1. Properties as an **object-centered** representation for different methods.

3D object recognition. With various 3D object datasets [7, 46, 47, 8] being created and becoming increasingly popular, 3D object recognition [48, 42, 41, 16, 32, 46, 31, 50, 33, 40, 8] has become a highly discussed topic in computer vision. Existing systems rely on given ground-truth 3D data in the form of either volumetric shapes [46], point-cloud sets [31], spherical maps [8], or multi-view images

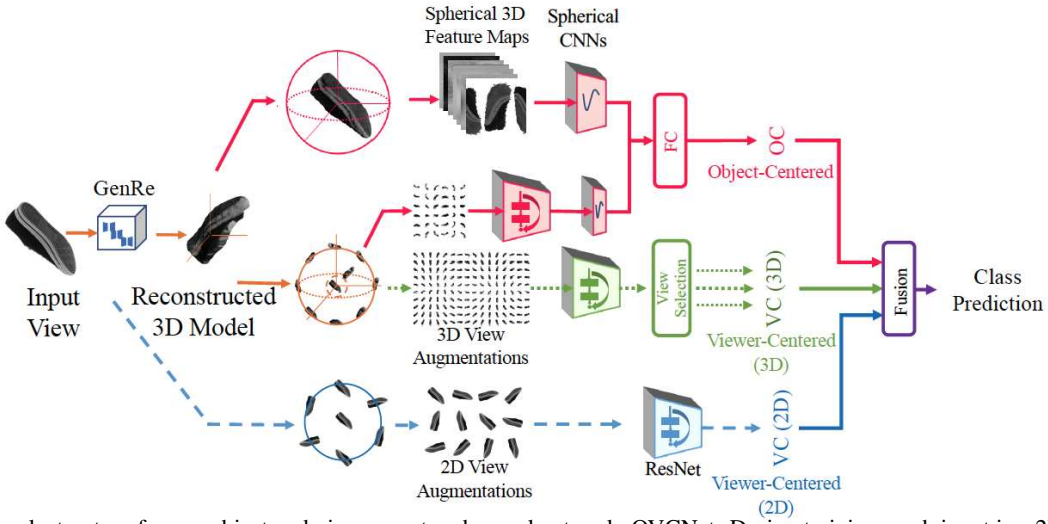


Figure 2. Network structure for our object and viewer-centered neural network, OVCNet. During training, each input is a 2D image of an object instance. OVCNet consists of 3 branches. For the top two branches, single-view 3D reconstruction using GenRe [51] is performed first. The first branch (Object-Centered) builds a representation using spherical maps [8]; the second branch (Viewer-Centered (3D)) builds a 2D CNN classifier with data augmentation using novel-view image syntheses. The third branch (Viewer-Centered (2D)) executes 2D based image classification with in-plane rotation for data augmentation. The final fusion layer provides a weighted sum of the outputs from the three branches/modules. Please see Section 4 for details about the three branches/modules, as well as the fusion layer.

[42, 41, 32, 16]. In contrast, we utilize these network structures as our recognition module following a single-view 3D reconstruction module.

2D Image-based object recognition. Viewer-centered feature learning has previously been addressed [3]. Broadly speaking, the recent common practice of data-augmentation can be considered viewer-centered feature learning where no new views are generated since the augmentation is mainly implemented in the 2D image plane.

Hybrid 2D and 3D object recognition. SPLATNet [40] is a hybrid system that integrates both 2D and 3D features for object classification and segmentation and is closely related to ours. However, SplatNet takes two modalities of inputs: a point-cloud based 3D shape and 2D multi-view images. Hence the scope of SplatNet is very different from ours.

Data-augmentation for transfer learning. There have been recent works in transfer learning [39, 34, 21, 11, 24] where data-augmentation is performed subject to certain domain adaption and regularization. These approaches address a fairly different problem compared to ours. We focus on the basic problem for 3D single image classification instead of a multi-task prediction problem.

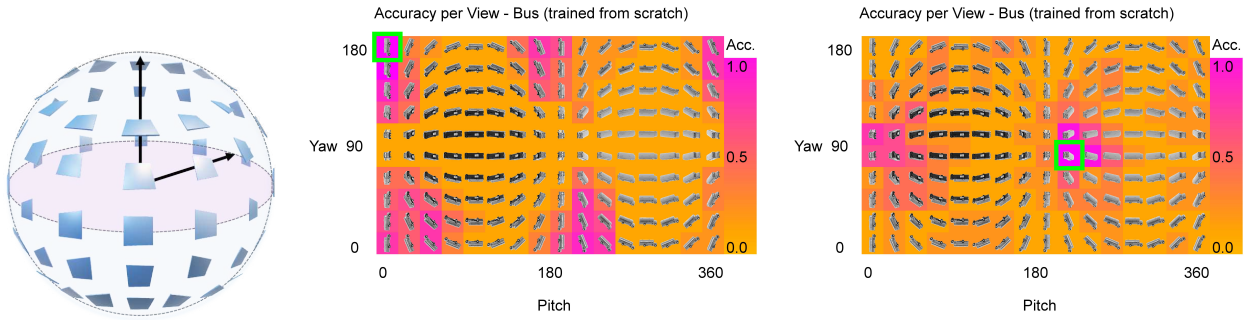
Single-view 3D reconstruction. In the field of single-view 3D reconstruction, an object-centered network outputs 3D information in a canonical view of the object. In contrast, a viewer-centered network's 3D output is relative to the input view [37, 45]. This definition is significantly different from what we define previously for recognition tasks. Nonetheless, for better reconstruction, Shin *et al*. and Tatarchenko *et al*. have shown that using 3D-supervision in a viewer-centered coordinate system tends to generalize better against unseen classes. Better generalization for unseen categories allows us to acquire 3D shape priors for new instances in an image without any 3D shape information during training. We adopt the state-of-the-art method for unseen class reconstruction, GenRe [51], to reconstruct 3D shape from a 2D single image, but GenRe itself does not perform image recognition.

Spherical CNNs. We build our chosen object-centered representation based on spherical CNNs [8], which is an effective and efficient way to obtain 3D shape representation for the 3D object classification tasks. Spherical CNNs themselves do not perform object recognition from any view, and a 3D input is required to generate the spherical map that spherical CNNs need.

To summarize, we focus on a challenging problem setting for object recognition from any view using object and viewer -centered representations.

3. Problem formulation

In this section, we focus on the any view object classification task. During training, the input is an arbitrary single view per training object instance, and the output is the ground truth class label. Every object instance is seen only once. We evaluate the effectiveness of OVCNet in two aspects: 1) SeenInstances: the ability to recognize novel views of seen (familiar) object instances (instances that are used in training) and 2) NovelInstances: the ability to recognize arbitrary views of novel/unseen object instances (instances absent from the training set). We present results from two experiments corresponding to these two aspects.



(a) (b) trained on view 1 (c) trained on view 90

Figure 3. (a) is an example of viewpoints used for generating viewpoint dependent images for the Viewer-Centered (3D) module (Section 4.3) similar to [16]. (b) and (c) show the classification accuracies across all viewpoints for a ResNet18 model trained only on view 1 (b) and view 90 (c) (highlighted) of the objects, respectively, on the MIRO dataset [16]. Without seeing other views, classic 2D CNNs have unsatisfactory performances on novel views.

4. Network architecture

4.1. Single-view shape prior

Given a single view of an object instance, we first use a state-of-the-art algorithm, GenRe [51], to generate 3D object reconstruction from a 2D image. GenRe separates reconstruction into three sub-tasks: depth estimation, spherical map inpainting, and voxel refinement. The separation of these tasks enables reasonable reconstruction for unseen objects/classes. Therefore, no additional object-specific information is needed. The pretrained GenRe model is only trained on three object classes ("plane", "car", and "chair") for reconstruction, but GenRe has shown great potential when it is evaluated on a wide variety of unseen object categories [51]. In our classification task on the gMIVO dataset, we include plane, car, chair, as well as other object classes such as lamp, pistol, motorbike, knife, laptop, guitar, and table. We adopt the trained GenRe model [51] directly to perform 3D reconstruction for a 2D image and add texture information to the final 3D model. We sample the texture information from the seen side with the nearest neighbor search algorithm using a k-d tree. This approach may result in different texture patterns due to different vertex ordering. A better texture filling approach should be explored in future studies.

4.2. Object-centered representation (OC module)

We utilize existing 3D recognition network structures as our classification module following GenRe's 3D shape estimation. We evaluate all three 3D shape-based recognition networks in Table 1: 3D CNNs, PointNet [31], and spherical CNNs [8], respectively. 3D CNNs is a 3D convolutional network inspired by 3DShapeNet [46] and built on top of [23]. Among them, spherical CNNs match the most with our object-centered definition for the following reasons.

First, spherical CNNs model is a 3D shape-based method. Object classification is carried out based on distance spherical maps along with cosine and sine of surface signals from 3D objects and their convex hulls. With spherical information of 3D models as input, the results of spherical CNNs on ShapeNet SHREC17 [7] are close to the state-of-the-art [8]. One can generate a spherical distance map by shooting a ray from the surface of a sphere (with a fixed radius) to the center of the object. The distance between the sphere surface and the object surface becomes the distance value captured by the spherical distance map [8]. *Second*, spherical CNNs use convolutions directly in the spherical harmonic domain, which keeps 3D rotation-equivariance of the spherical signals. See discussions about an empirical support for rotation-invariance in [8]. More discussion on its rotation-invariant capability is provided in the supplementary materials. *Third*, the network does not require any pose alignment.

In our overall model, we refer to the object-centered module branch with spherical CNNs as the **OC^b module**, where the superscript b indicates that it is a base module.

4.3. Viewer-centered representation (VC module)

For viewer-centered representations, different modules with two different inputs are used: 1) the original view **VC (2D) module**; 2) views re-projected using 3D viewpoint augmentation from the 3D output of the GenRe **VC (3D) module**. For both tasks, we find that ResNet18 works well as a 2D image classifier compared to other classic convolutional neural networks. To select augmented views, we implement three options for the view selection layer (discussed in detail in Section 4.4).

VC (2D) module. This module uses 2D augmentation with in-plane rotation. We evaluate ResNet18 with different angles of rotation augmentation, including intervals of 90, 30, 10, 5, and 1 degrees for gMIRO. We observe that the evaluation accuracy stops increasing as we provide denser angle augmentations. Rotation ablation studies (see the supplementary materials) show that ResNet18's accuracy plateaus when we augment the input view with 2D in-plane rotations at 30-degree intervals for the gMIRO dataset. In contrast, for gMIVO, the network performance plateaus with aug-

mentations of 90-degree intervals. We use these numbers in our later experiments. If trained under identical views, ResNet18, as shown in Figure 3.b and c, experiences difficulties recognizing images from new angles for the same set of objects. We refer to this effect as " mental rotation".

VC (3D) module. This module uses 2D augmentation from 3D viewpoints. We augment images with 10 evenly divided elevation angles and 16 evenly divided azimuth angles, yielding 160 views per object. The viewpoint augmentation setting is shown in Figure 3.a [16]. The viewpoint layout imitates the organization of object views in the dataset, starting from the input view. Additionally, we add in-plane (2D) rotation augmentations in 90-degree intervals to each augmented viewpoint.

For the VC (3D) module, we explore three types of view selection methods: 1) the nearest neighbor approach where the network only uses the augmented image that is closest to the input viewpoint for testing; 2) a simple selection layer where the network learns a set of weights for all augmented views; 3) an attention layer where the network learns a set of attention weights based on the input information. Option 1 is the most suitable for a dataset that has limited training views, such as gMIRO, and is the most efficient in terms of runtime. For options 2 and 3, we further divide the training views into a sub-training set1 and set2. We first use set1 for training ResNet18 and then use the set2 to train the selection network. We observe an improvement in average accuracy using a view selection network compared to a simple ensemble of all augmented views. However, given the limitation of the 3D reconstruction and size of the dataset, for the gMIRO dataset, using the input viewpoint alone outperforms the other options.

Other augmentations are also considered. We include 20 views taken from the 20 vertices of a dodecahedron around the object [41, 16] for a GenRe [51] + multi-view baseline. We also include 36 viewpoints from a sampling grid of spherical maps with a bandwidth of 3 [8] for a viewer-centered assisted object-centered module, OC (Section 4.4).

For the multi-view baseline, we include GenRe + multi-view CNN (MVCNN) [41] and GenRe + RotationNet [16]. A 20-view version of MVCNN is used due to memory constraint. The best performing backbones are VGG for MVCNN and ResNet18 for RotationNet. The results are encouraging for GenRe + MVCNN. However, MVCNN uses pretrained weights and requires 20-view augmentation during test. In contrast, we train our model with a single view and from scratch to avoid prior knowledge of unseen instances learned from the pretrained dataset.

4.4. Fused representation (OVCNet)

In summary, our overall network (Figure 2) includes 3 branches: OC^b branch ($GenRe^{text}$ + spherical CNNs [8]), VC (3D) branch ($GenRe^{text}$ + ResNet18 [14] + view selection), and VC (2D) branch (ResNet18).

To fuse the OC^b base module with the VC (3D) module, we create an OC module (Figure 2). In this module, in addition to the 160-view set, we use the information from 36 augmented views to reduce the number of views needed for training. We then organize the learned ResNet features into a grid and pass them into an ancillary spherical CNNs with an input bandwidth of 3. This new branch is then trained with the original OC^b base module fused by a fully connected layer as the final OC module. The result of gMIRO is shown in Table 3.

To fuse the output of OC and VC modules, we experiment with 3 options. The first option is to train a fully connected fusion layer with or without each module frozen. The second option is to learn an attention layer to fuse the three results. The third option is to use a set of weights found through a grid search using a validation set. Our experiment has shown that the third option works the best for the gMIRO dataset. Two reasons may contribute to this: 1) different branches have different learning rates due to diverse input and module modalities; 2) Even with the three branches frozen, the simpler fusion method adapts better when we have limited training information. We find the learned weights from option 3 are stable, *e.g.*, around 0.2, 0.3, and 0.5 for combining OC module, VC (3D) module, and VC (2D) module on both gMIRO and gMIVO datasets.

Please see the supplementary materials for details on runtime analysis.

5. Experiments

5.1. Baselines

Next, we report the results of various baseline classifiers as well as those by our OVCNet.

Traditional image classification networks. We learn 2D image classification using convolutional neural networks including AlexNet [18], ResNet18 [14], and ResNet152 [14] directly on the input views. For AlexNet and ResNet18, the batch size for training is 96. For ResNet152, a batch size of 32 is used due to memory constraint. We start with an initial learning rate of 0.01 and decay by 10 every 30 epochs. ResNet18 seems to generalize better and has more efficient memory usage.

3D shape-based classification networks. We convert the reconstructed 3D object from GenRe to voxels ($30\times30\times30$ or $128\times128\times128$), point sets (2500 point samples), and distance spherical maps in order to run 3D CNNs [23], PointNet [31], and spherical CNNs [8], respectively, without texture information.

Re-projected viewer-centered classification networks. For re-projections from GenRe's output, as a baseline for VC (3D) module, we evaluate ResNet18 with a different number of view augmentations. Although our algorithm only uses a single view during testing in the overall model, we also show our results with 20 views during evaluation

with GenRe [51] + RotationNet [16] and GenRe + MVCNN [41] as a multi-view module baseline.

Object and viewer-centered network. Since OVCNet combines three modules, for a fair comparison, we include two ensemble strategies of three VC (2D) modules and report the results in Table 2 (ResNet18$^{\text{rot30/90}}$ Ensemble I, II). To compare with the ensemble results, we randomly select six VC (2D) modules and report the average over two sets of ensemble results. For a fair comparison with OVCNet, we randomly select one VC (2D) module from the ensemble set to combine with our OC module and VC (3D) module. Ensemble I uses three equally weighted random models of the same type. Ensemble II trains additional fusing weights for the three random models.

	accuracy overall (%) SeenInstances	accuracy overall (%) NovelInstances
gMIRO		
AlexNet [18]	24.61 ± 3.02	27.40 ± 2.13
ResNet152 [14]	45.97 ± 1.08	43.68 ± 1.91
ResNet18 [14]	51.34 ± 0.52	44.04 ± 1.31
ResNet18*	45.08 ± 0.98	38.70 ± 2.09
ResNet18$^{\text{r30}}$(VC (2D))	68.34 ± 1.57	53.27 ± 0.89
ResNet18$^{\text{rot30}}$ (Ensemble I)	70.56 ± 0.56	54.91 ± 1.85
ResNet18$^{\text{rot30}}$ (Ensemble II)	70.91 ± 0.34	55.74 ± 2.52
GenRe [51] + PointNet [31]	27.33 ± 0.48	27.67 ± 0.80
GenRe + 3D CNNs [23]	30.26 ± 0.62	30.01 ± 0.75
GenRe^{tex} + RotationNet^{pre} [16]	46.55 ± 3.97	46.44 ± 4.54
GenRe^{tex} + MVCNN^{pre} [41]	58.68 ± 0.59	54.56 ± 0.41
OVCNet (ours)	**73.24** ± 0.08	**65.85** ± 0.14
gMIVO (ShapeNetCoreV2 subset)		
ResNet18$^{\text{rot90}}$ (VC (2D))	64.40 ± 0.45	64.86 ± 0.43
ResNet18$^{\text{rot90}}$ (Ensemble I)	65.70 ± 0.25	66.25 ± 0.59
ResNet18$^{\text{rot90}}$ (Ensemble II)	65.73 ± 0.18	66.27 ± 0.44
OVCNet (ours)	**79.24** ± 0.12	**75.03** ± 0.30

Table 2. **Results summary.** ResNet18*: a standard 2D image data augmentation [18]. ResNet18$^{\text{rot[d]}}$: 2D in-plane rotation augmentation with multiples of d degree rotation. GenRe^{tex}: texture is used for 3D viewpoint augmentation. RotationNet^{pre} and MVCNN^{pre}: using pretrained weights. Ensemble I uses an equally weighted ensemble of three models. Ensemble II includes learned fusing weights for the three random models. Two repeats for OVCNet and the ensembles.The proposed OVCNet performs the best here.

5.2. Datasets

We adopt the following three datasets: a grayscale version of the MIRO dataset [16] (**gMIRO**), our new dataset, grayscale multi-view images of virtual objects (**gMIVO**), and natural-colored images from Pascal VOC [12].

gMIRO. We use preprocessed grayscale images from the MIRO dataset [16] (gMIRO) as our primary dataset for ablation studies. This dataset contains 12 classes with 10 object instances for each class. For each object, there are 160 views (10 elevations $\times$ 16 azimuth angles) from real objects with empty backgrounds. We randomly select 80% of the instances as familiar object instances. For each object, we randomly select an arbitrary single view to use in the training set (12 $classes$ $\times$ 10 $objects$ $\times$ 80% $seen$ $split$ $\times$ 1 $view$ $=$ 96 $images$). We use the remaining views of the familiar instances as the first test set that evaluates how well the model generalizes towards unseen views of seen object instances (SeenInstances). The final test is done utilizing all the views from the remaining 20% new instances, where we can evaluate the generalization towards views from all 160 angles of the unseen object instances (NovelInstances).

gMIVO. gMIVO is a larger dataset with a similar setup as gMIRO. A subset of ShapeNetCore v2 is selected to generate this dataset. We do not use ModelNet [46] directly for this paper because an aligned ModelNet40 was not available at the time the project first started. Additionally, most of the objects are lacking material and texture information. ShapeNetCore v2 includes materials and textures and all objects are aligned [7]. We select a subset of the objects from ShapeNetCore v2 by referring to the 10 classes with the highest frequency from DensePoint [6] (which uses ShapeNetCore v2 objects with good material and texture information) and take 160 views of each object. This new dataset contains ten classes where each class has 110 objects. For each object, 160 views are generated using similar viewpoints from MIRO [16] as shown in Figure 3.a. Our rendering tool is built on top of the Stanford ShapeNet renderer. During training, we randomly select 80% of the objects for every class as the familiar objects. The two test sets, SeenInstances and NovelInstances, are set up similarly to gMIRO.

Pascal VOC. We use a subset of Pascal VOC images [12] to evaluate the capability of OVCNet with real color images with background. For training, to use GenRe, we obtain the masks for each object from [30]. For testing, an object mask is first obtained through a foreground segmentation algorithm using [15]. We choose images of aeroplane, bicycle, car, and motorbike because there are fewer occlusions in those images, which allows adequate 3D reconstructions. We randomly select 20% of the images from each category for training and the remaining for testing.

We start with grayscale images for gMIRO and gMIVO to illustrate the fundamental idea of OVCNet. We then experiment with colored inputs for MIRO and PASCAL images. Please see Section 6 for more details.

5.3. Metrics

For both the gMIRO and gMIVO datasets, we partition the data into familiar and novel instances with an $80\%/20\%$ train-test split. If not otherwise specified, we conduct three repeats for each experiment and averge the results. We report the overall class accuracy (the mean and standard deviation) for unseen views with seen objects (SeenInstances) and all views with unseen objects (NovelInstances).

6. Results and Discussions

Object-centered feature learning. For the object-centered branch, we compare the results of different representations of the 3D reconstruction using GenRe + 3D CNNs, GenRe + PointNet, and GenRe + spherical CNNs in Table 3. We

find that the performance for GenRe + 3D CNNs increases as the voxel resolution increases; however, the network size increases as well. For GenRe + spherical CNNs, the performance increases as bandwidth increases and plateaus at $bandwidth = 112$ for gMIRO. Overall, our OC branch outperforms other combinations in terms of overall accuracy for both SeenInstances and NovelInstances with comparable network size. Additionally, the OC module that further integrates information learned from the VC (3D) branch (bw=3 sgrid) can give OC^b baseline module an extra 10% boost on the gMIRO dataset.

Networks	accuracy overall (%)	
	SeenInstances	NovelInstances
GenRe + 3D CNNs [23] ($30 \times 30 \times 30$vx)	20.94 ± 0.41	21.74 ± 0.42
GenRe + 3D CNNs ($128 \times 128 \times 128$vx)	30.26 ± 0.62	30.01 ± 0.75
GenRe + PointNet [31] (2500pt)	27.33 ± 0.48	27.67 ± 0.80
GenRe + spherical CNNs [8] (bw=60)	40.79 ± 1.21	41.50 ± 0.44
GenRe + spherical CNNs (bw=112)	42.43 ± 1.24	40.80 ± 0.51
GenRe + spherical CNNs (bw=128)	40.94 ± 1.88	41.23 ± 0.77
$GenRe^{tex}$ + spherical CNNs (bw=112) (OC^b)	44.62 ± 0.58	44.65 ± 0.53
OC branch	**54.62** ± 0.73	**54.21** ± 0.54

Table 3. **Ablation study for object-centered network structure (OC) on gMIRO**. $GenRe^{tex}$ + spherical CNNs [8] (with additional approximated texture spherical map information) is chosen as our OC^b module in our OVCNet due to its relative performance advantage. *vx* indicates voxel representation, *pt* indicates point cloud representation, and *bw* indicates the bandwidth for spherical signals. The final OC model with an ancillary spherical pathway integrating the information learned from the VC (3D) module (bw=3 sgrid) performs the best.

Viewer-centered feature learning. For viewer-centered network structures with re-projected 2D images (VC (3D) module), we compare different 3D viewpoint augmentations during training, shown in Table 4. For GenRe + ResNet18, the performance increases as the number of training viewpoints increases. Once we introduce texture in the re-projection, both GenRe + MVCNN and VC (3D) outperform other methods. GenRe + MVCNN uses all 20 different viewpoints for testing. In contrast, VC (3D) only uses one viewpoint during the evaluation. Hence, it is more efficient than GenRe + MVCNN.

We also experiment with the attention structure as our view-selection layer (Not shown in tables). Compared to a simple ensemble of all 160 views at test time, we do notice a performance gain from the attention view selection layer in the Pascal dataset. This result suggests that a more complex view selection module during inference may boost the performance with increased training data.

For viewer-centered network structures with original 2D images (VC (2D) module), we conduct an ablation study on 2D rotation augmentation. In the supplementary materials, we show that, for gMIRO, the performance of ResNet18 plateaus with rotations of 30-degree intervals (12 augmented images per input). For gMIVO, we find that the performance of ResNet18 plateaus with rotations of 90-degree intervals (4 augmented images per input). These results may indicate that with increasing number of training instances, random viewing angles of similar instances increase. Hence, less in-plane rotation is needed to boost performance.

	3D-aug 1/160/640	accuracy overall (%) SeenInstances	accuracy overall (%) NovelInstances
GenRe + ResNet18	1	32.49 ± 0.68	32.95 ± 0.93
GenRe + ResNet18	160	45.15 ± 0.46	40.20 ± 0.51
GenRe + ResNet18	640	51.24 ± 0.23	47.57 ± 0.55
$GenRe^{tex}$ + $RotationNet^{pre}$ [16]	20	46.55 ± 3.97	46.44 ± 4.54
$GenRe^{tex}$ + $MVCNN^{pre}$ [41]	20	58.68 ± 0.59	54.56 ± 0.41
scratch VC (3D) (ours)	640	**65.70** ± 0.44	**58.27** ± 0.04

Table 4. **Ablation study for viewer-centered network structures with gMIRO** by using different types of data augmentations. **3D-aug**: the number of re-projected images used during training. Section 4.3 offers viewpoint details. $GenRe^{tex}$ + $MVCNN^{pre}$ and $GenRe^{tex}$ + $RotationNet^{pre}$ use fine-tuned weights with pretrained models and 20 views for evaluation, whereas other methods only use single view. The final VC (3D) model with $GenRe^{tex}$ and ResNet18 trained from scratch performs the best.

Object and viewer-centered network. Finally, we combine the results from both object (OC) and viewer -centered modules (VCs) for both gMIRO and gMIVO datasets. Through a simple grid search on the validation sets, the fusion layer outputs a weighted sum of probabilities from OC, VC (3D), and VC (2D) branches. The results are shown in Table 5. Our results show that the three models are complementary to each other for both datasets.

The advantage of OVCNet over the ensemble of ResNet18s appears to be more significant for gMIVO. The test accuracy improves by $\sim 13.5\%$ for unseen views of familiar object instances and $\sim 9\%$ for novel object instances in Table 2. It suggests that training with more arbitrary views of instances from the same category helps with classifying views from other viewpoints. Interestingly, for gMIVO in Table 5, the test accuracy of the VC (3D) branch alone is already higher than that of VC (2D); this further validates the importance of inferring 3D reconstruction through which our 3D view augmentation is realized.

We also evaluate the average class accuracy for OVCNet and the corresponding ensemble baseline (not shown in tables). For gMIVO, for all ten classes, the SeenInstances (other views from familiar instances) accuracy is raised by 13.41% from 65.89% to 79.36%. The NovelInstances (all views from novel instances) accuracy is raised by 8.68% from 66.65% to 75.33% (we list these numbers here in the text directly).

Given that we use a pretrained GenRe model that is trained on three classes from ShapeNet and our gMIVO dataset is also a subset of ShapeNet, we additionally test on gMIVO after removing the three classes that are overlapping between the two datasets. Our model shows a slightly greater improvement compared to using all ten classes. The final OVCNet model outperforms the ensemble of VC (2D) by 14.45% for unseen views of seen objects and 9.3% for

Experiments	OC	VC (3D)	VC (2D)	SeenInstances accuracy (%)	NovelInstances accuracy (%)
gMIRO					
(1)	✓			52.65	53.02
(2)		✓		65.70	58.31
(3)			✓	69.74	54.11
(4)	✓	✓		67.24	61.48
(5)		✓	✓	72.47	62.99
(6)	✓		✓	72.04	58.57
OVCNet	✓	✓	✓	**73.25**	**65.99**
gMIVO (ShapeNetCoreV2 subset)					
(1)	✓			52.83	50.49
(2)		✓		77.00	70.53
(3)			✓	63.66	64.50
(4)	✓	✓		77.60	71.23
(5)		✓	✓	77.71	74.50
(6)	✓		✓	67.83	67.63
OVCNet	✓	✓	✓	**79.36**	**75.33**

Table 5. **Ablation study over different model integrations**. gMIRO uses an OC module (see Section 4.4), whereas gMIVO uses an OC^b module (see Section 4.2). For the VC (3D) branch (see Section 4.3), gMIRO uses textured reconstructed 3D models from GenRe to generate 640 3D viewpoint augmentations per input view, whereas gMIVO uses 160 viewpoints. For the VC (2D) branch (see Section 4.3), gMIRO uses 30-degree intervals whereas gMIVO uses 90-degree intervals. The three modules are shown to be complementary to each other on both datasets.

all views of unseen objects. We demonstrate that the effectiveness of OVCNet does not depend on the training classes from GenRe. The improvement may be due to the removed classes being harder to classify.

	$80\% - 20\%$	$50\% - 50\%$	$20\% - 80\%$
	test accuracy for SeenInstances (%)		
VC (2D)	68.34 ± 1.57	64.42 ± 0.43	64.53 ± 0.84
OVCNet*	69.95 ± 0.35	67.24 ± 0.08	69.13 ± 0.75
	test accuracy for NovelInstances (%)		
VC (2D)	53.27 ± 0.89	47.36 ± 0.83	36.66 ± 0.54
OVCNet*	59.57 ± 0.28	50.99 ± 0.31	42.09 ± 0.06

Table 6. **Ablation study with different train-test split percentages**. Each column corresponds to a different train-test split for the gMIRO dataset. OVCNet* uses a less optimal configuration compared to the OVCNet used in Table 2. Under varying training sizes, the trend of OVCNet w.r.t. VC (2D) is consistent as in Table 5 and Table 2.

Ablation study for train-test split percentages. To evaluate our model's performance on the varying training data size, we experiment with two more train-test splits. In addition to the original split (80% familiar instances vs. 20% new instances), we also test $50\%/50\%$ and $20\%/80\%$ train-test splits. Table 6 shows the means and standard deviations for the test accuracies on seen instances and novel instances under multiple repeats. As the number of familiar instances decreases, the overall classification accuracy also declines, which is typical when trained on fewer data. However, we see a similar improvement as that in Table 5 and Table 2 for OVCNet w.r.t. VC (2D) module. These experiments are tested with an earlier version of OVCNet for gMIRO that uses a less optimal configuration than what is used in Table 5 and Table 2.

Color and Natural Images.

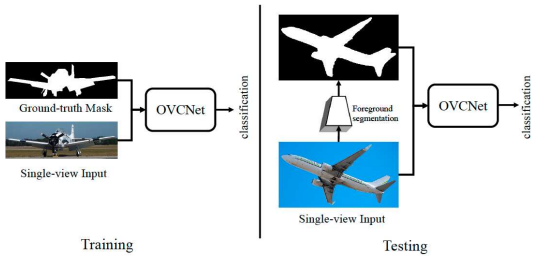


Figure 4. Algorithm pipeline for the PASCAL experiment.

Our experiments in Table 3 show the results of combining approximated texture information with the grayscale input. In a similar spirit, we also provide results for colored input as follows (not shown in tables). We use color images from MIRO to train the VC (2D) module (ResNet18 with in-plane rotations) as a baseline; we keep OC and VC (3D) the same since they mostly concern with shape. Nevertheless, our results show that, for gMIRO, OC, and VC (3D) modules still provide a consistent boost to the VC (2D) baseline trained with color images from MIRO. The accuracy improves from 73.23% to **75.64**$\%$ for SeenInstances (unseen views from familiar instances) and from 54.53% to **67.66**$\%$ for NovelInstances (unseen instances). This improvement validates the benefit of having an object- and viewer-centered representation for colored images as well.

	test accuracy (%)
OC^b (bw=112)	80.08
VC (3D) (160)	82.35
VC (2D)	72.84
VC (2D) (*Ensemble I*)	75.49
VC (2D) (*Ensemble II*)	75.91
OVCNet	**85.24**

Table 7. Test accuracy for Pascal VOC subset images for the aeroplane, bicycle, car, and motorbike classes.

An evaluation of natural-colored images with a background (a subset of Pascal VOC) also shows encouraging results. Experimental results are reported in Table 7. We see a 10% improvement over the baseline. Random rotation does not improve the performance for VC (2D) here.

7. Conclusion

We have developed a new algorithm for any view object recognition that is inspired by the object and viewer-centered recognition theories. The resulting OVCNet is an integrated framework that learns viewpoint-independent and viewpoint-dependent features for an image from an unknown view, and it can be used to recognize novel instances from novel views. We show a clear advantage of OVCNet over the object-centered and viewer-centered baselines in Table 2 and 5. We also report results on natural-colored images in Table 7.

Acknowledgment. This work is supported by NSF IIS-1717431 and NSF IIS-1618477. We thank Wenlong Zhao, Justin Lazarow, Hao Su, and Jiajun Wu for the help and valuable feedbacks.

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 *CVPR*, 2016. 1

[2] Dana H. Ballard and Daniel Sabbah. Viewer independent shape recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):653–660, 1983. 1

[3] Ronen Basri. Viewer-centered representations in object recognition: A computational approach. In *Handbook of pattern recognition and computer vision*, pages 863–882. World Scientific, 1993. 1, 3

[4] Irving Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94(2):115, 1987. 1

[5] E. Darcy Burgund and Chad J. Marsolek. Invariant and viewpoint-dependent object recognition in dissociable neural subsystems. *Psychonomic Bulletin & Review*, 7(3):480–489, 2000. 2

[6] Xu Cao and Katashi Nagao. Point cloud colorization based on densely annotated 3d shape dataset. In *International Conference on Multimedia Modeling (MMM)*, 2019. 6

[7] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 2, 4, 6

[8] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. In *ICLR*, 2018. 2, 3, 4, 5, 7

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 1, 2

[10] Vaibhav A. Diwadkar and Timothy P. McNamara. Viewpoint dependence in scene recognition. *Psychological Science*, 8(4):302–307, 1997. 1

[11] Mandar Dixit, Roland Kwitt, Marc Niethammer, and Nuno Vasconcelos. Aga: Attribute-guided augmentation. In *CVPR*, 2017. 3

[12] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 2, 6

[13] William G. Hayward. Whatever happened to object-centered representations? *Perception*, 41(9):1153–1162, 2012. 1, 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 1, 2, 5, 6

[15] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 6

[16] Asako Kanezaki. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *CVPR*, 2018. 2, 3, 4, 5, 6, 7

[17] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3 d shape descriptors. In *Symposium on Geometry Processing (SGP)*, volume 6, 2003. 1

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012. 1, 2, 5, 6

[19] Takio Kurita and Takashi Takahashi. Viewpoint independent face recognition by competition of the viewpoint dependent classifiers. *Neurocomputing*, 51:181–195, 2003. 2

[20] Simon Lacey, Andrew Peters, and Krish Sathian. Cross-modal object recognition is viewpoint-independent. *PLoS One*, 2(9):e890, 2007. 1

[21] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014. 3

[22] Joerg Liebelt, Cordelia Schmid, and Klaus Schertler. Viewpoint-independent object class detection using 3d feature maps. In *CVPR*, 2008. 1

[23] Riccardo Lincetto. https://github.com/riclincio/3d-shape-classification. 4, 5, 6, 7

[24] Bo Liu, Xudong Wang, Mandar Dixit, Roland Kwitt, and Nuno Vasconcelos. Feature space transfer for data augmentation. In *CVPR*, 2018. 3

[25] Nikos K. Logothetis and Jon Pauls. Psychophysical and physiological evidence for viewer-centered object representations in the primate. *Cerebral Cortex*, 5(3):270–288, 1995. 1

[26] Nikos K. Logothetis, Jon Pauls, and Tomaso Poggio. Shape representation in the inferior temporal cortex of monkeys. *Current Biology*, 5(5):552–563, 1995. 1, 2

[27] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., USA, 1982. 1, 2

[28] Patricia A. McMullen and Martha J. Farah. Viewer-centered and object-centered representations in the recognition of naturalistic line drawings. *Psychological Science*, 2(4):275–278, 1991. 2

[29] Branka Milivojevic. Object recognition can be viewpoint dependent or invariant–it's just a matter of time and task. *Frontiers in Computational Neuroscience*, 6:27, 2012. 2

[30] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014. 6

[31] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 2, 4, 5, 6, 7

[32] Charles R. Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *CVPR*, 2016. 2, 3

[33] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, 2017. 2

[34] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 3

[35] Irvin Rock and Joseph DiVita. A case of viewer-centered object perception. *Cognitive psychology*, 19(2):280–293, 1987. 1

[36] Roger N. Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971. 1

[37] Daeyun Shin, Charless C. Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In *CVPR*, 2018. 3

[38] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[39] Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems*, 2013. 3

[40] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. SPLATNet: Sparse lattice networks for point cloud processing. In *CVPR*, 2018. 2, 3

[41] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *ICCV*, 2015. 2, 3, 5, 6, 7

[42] Hao Su, Min Sun, Li Fei-Fei, and Silvio Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *ICCV*, 2009. 2, 3

[43] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 1

[44] Michael J. Tarr and Quoc C. Vuong. Visual object recognition. *Stevens' handbook of experimental psychology*, 2002. 1, 2

[45] Maxim Tatarchenko*, Stephan R. Richter*, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *CVPR*, 2019. 3

[46] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015. 2, 4, 6

[47] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. Objectnet3d: A large scale database for 3d object recognition. In *ECCV*, 2016. 2

[48] Jianxiong Xiao, Bryan Russell, and Antonio Torralba. Localizing 3d cuboids in single-view images. In *Advances in Neural Information Processing Systems*, 2012. 2

[49] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 1

[50] Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. Attentional shapecontextnet for point cloud recognition. In *CVPR*, 2018. 2

[51] Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Josh Tenenbaum, Bill Freeman, and Jiajun Wu. Learning to reconstruct shapes from unseen classes. In *Advances in Neural Information Processing Systems*, 2018. 2, 3, 4, 5, 6