IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2018.DOI*

# Volumetric Object Recognition Using 3D CNNs on Depth Data

**ALI CAGLAYAN[1, 2], (Student Member, IEEE) and AHMET BURAK CAN[1], (Member, IEEE)**

[1]Department of Computer Engineering, Hacettepe University, Beytepe Campus, 06800, Ankara, Turkey
[2]Department of Computer Engineering, Bingol University, 12000, Bingol, Turkey

Corresponding author: Ahmet Burak Can (e-mail: abc@cs.hacettepe.edu.tr).

We gratefully acknowledge the support of NVIDIA Corporation with the donation of graphics card. We also thank the anonymous reviewers for their valuable comments.

**ABSTRACT** Recognizing 3D objects has a wide range of application areas from autonomous robots to self-driving vehicles. The popularity of low-cost RGB-D sensors has enabled a rapid progress in 3D object recognition in the recent years. Most of the existing studies use depth data as an additional channel to the RGB channels. Instead of this approach, we propose two volumetric representations to reveal rich 3D structural information hidden in depth images. We present a 3D Convolutional Neural Network (CNN) based object recognition approach, which utilizes these volumetric representations and single and multi-rotational depth images. The 3D CNN architecture trained to recognize single depth images produces competitive results with the state-of-the-art methods on two publicly available datasets. However, recognition accuracy increases further when multiple rotations of objects are brought together. Our multi-rotational 3D CNN combines information from multiple views of objects to provide rotational invariance and improves the accuracy significantly comparing to the single-rotational approach. The results show that utilizing multiple views of objects can be highly informative for 3D CNN based object recognition.

**INDEX TERMS** 3D object recognition, convolutional neural networks, volumetric representations

## I. INTRODUCTION

As the computer vision area tries to discover the external world from images, utilizing 3D information in recognition systems has been one of the earliest interests in the computer vision literature [1]–[4]. Since the real world is three dimensional, discovering new 3D representations of objects has been an important problem for the object recognition area. In the recent years, the advent of low-cost RGB-D sensors such as the Microsoft Kinect has created a new trend in 3D object recognition. These sensors provide depth and RGB data together, which enables to build reliable 3D object representations. As depth data is relatively robust to illumination, color, and viewpoint changes, depth information plays an important role in 3D object recognition solutions.

Despite considerable progress, object recognition on depth data is still an open research area. Most existing efforts in the field (e.g. [5], [6]) have concentrated on using the depth image as an extra channel in addition to the RGB channels. However, RGB and depth images have different characteristics. While RGB data have color and rich texture information, depth data are strongly characterized by 3D structural information of objects and insensitive to changes in lighting conditions. Especially in the field of robotics, this kind of data is needed to increase a robot's interaction capabilities in darkness, mitigating some issues concerning low light conditions [7]. Moreover, some misclassification problems in RGB images caused by viewpoint changes can also be reduced by utilizing depth data. To take full advantage of depth data, depth-specific approaches are needed in object recognition. In this context, compact 3D object representations have a key role in devising the best performing algorithms.

Historically, a typical object recognition system has relied on a hand-crafted feature extraction step followed by an efficient classifier. However, these methods lack good generic representations and require the design of domain-specific solutions. In the last years, Convolutional Neural Networks (CNNs) [8] have presented a cutting-edge research in computer vision. CNNs with the ability of automatic feature learning have surpassed object recognition approaches based on traditional hand-crafted features. More recently, CNN architectures have been successfully extended to the 3D data domain (e.g. [9], [10]). The key to these approaches are to handle 3D data in an adequately representative way. To this end, we propose a method to classify object categories using raw depth data. The method uses two types of volumetric representations for depth data and a 3D CNN model, which exploits 3D geometric cues of objects in a data-driven manner to predict an object category from the volumetric representations. Objects can be recognized using a single depth image as well as multiple images from different viewpoints. The latter is especially important for a robot, which dynamically interacts with its environment to overcome viewpoint-related ambiguities. As the robot moves to new viewpoints, recognition success can increase drastically by bringing together the cues gained from different perspectives. In this paper, we extend the early version of this work [11] in the following aspects. Firstly, we develop a more robust 3D CNN architecture based on [10] to prevent overfitting and to help the model generalize better. Secondly, we explore the use of a single depth image as well as the role of multi rotational views of objects together. In this way, the information aggregated through the network from multiple rotations of objects can greatly increase performance. Moreover, we construct a new dataset as a subset of the Washington RGB-D dataset [12] using multiple rotations of objects. Finally, we present additional experiments on different scenarios to demonstrate the effectiveness of the proposed method. Experimental results on the Washington RGB-D and the 2D3D Object datasets show that while single-rotational approach produces competitive results with the state-of-the-art methods, the multi-rotational approach extends recognition accuracy further by providing rotational invariance.

## II. RELATED WORK

Object recognition approaches using depth images have become widespread due to the affordable RGB-D sensors. These approaches can be divided into three categories based on their techniques: hand-crafted feature based methods [12]–[16], 2.5D CNN based methods [5], [6], [17]–[20], and 3D CNN based methods [9]–[11], [21].

### A. HAND-CRAFTED FEATURE BASED METHODS

Hand-crafted techniques rely on conventional feature extraction and aggregation in a concise representation. In these methods, mostly standard local features such as SIFT [22], SURF [23], and HOG [24] in the RGB domain are applied to the depth domain. Lai *et al.* [12] introduce a large-scale RGB-D object dataset and extract a set of standard color and shape features from the data to classify. Depth kernel descriptors [13] are presented to capture features including size, shape, edges, and pixel orientations in a unified way. Hierarchical kernel descriptors [14] expand [13] in a layered fashion where pixel attributes are aggregated into patch-level features layer by layer until to the last object level. Alternatively, [12] and [13] use spin images [25] as 3D local shape descriptors in depth domain. Furthermore, depth domain-specific Histogram of Oriented Normal Vector [16] is introduced to capture 3D geometric characteristics of depth images and gives better results. Consequently, all of these aforementioned methods need a careful feature extraction labor. Recent advances in deep learning techniques such as CNNs have eliminated the need for hand-crafted feature representations.

### B. 2.5D CNN BASED METHODS

In 2.5D CNN based methods [5], [6], [17]–[20], depth data is used as an additional channel to the RGB channels. Blum *et al.* [5] present the convolutional k-means (CKM) descriptor which extracts patches around SURF interest points and learns features in an unsupervised way. Similarly, convolutional-recursive neural network (CNN-RNN) [6] is proposed to learn color and depth features separately. Both CKM and CNN-RNN feature learning methods concentrate on the random patch extraction to learn filters. To improve this approach, a subset based feature learning method is presented in [17]. Even though this method leads to an improvement in recognition performance compared to the random patch extraction, the authors note that they are not dividing the object images into subsets automatically. Cheng *et al.* [18] present a semi-supervised method to make use of unlabeled data with a co-training algorithm along with CNN-RNN model. [19] extends [18] by adapting CNN-RNN model for arbitrary size of images by replacing the first step of CNN-RNN with a spatial pyramid pooling (SPM) layer [26]. Grayscale images and surface normals are also utilized in addition to the RGB and depth images in this method. The proposed work by Zaki *et al.* in [20] is motivated by two main objectives. One is taking advantage of large annotated datasets like ImageNet [27] to overcome limitations of learning on relatively small RGB-D datasets. To do that, the authors represent depth data and point cloud data into the RGB domain in order to allow knowledge transfer from a pre-trained CNN model. Secondly, they utilize earlier layers of CNN model to encode locally-activated features. Consequently, these spatially-coherent features are combined with the semantic informative features in the last fully-connected layer, which results in a significant classification improvement.

### C. 3D CNN BASED METHODS

While 2.5D CNN based methods improve recognition success significantly compared to the hand-crafted methods, volumetric information hidden in depth data is not fully utilized in these methods. To this end, a 3D CNN for object recognition and a large-scale 3D CAD model dataset named ModelNet have been introduced by Wu *et al.* [9]. Later, Voxnet [10] approach increases accuracy of the ShapeNet with a smaller network. Our previous work [11] introduces the first volumetric object recognition on the popular Washington RGB-D dataset following the success of ShapeNets [9] and VoxNet [10]. In this paper, we extend [11] with a new 3D CNN architecture and utilize multiple rotations of objects in addition to single depth images. We also provide additional experiments on two publicly available datasets, the Washington RGB-D and the 2D3D Object datasets. Our multi-

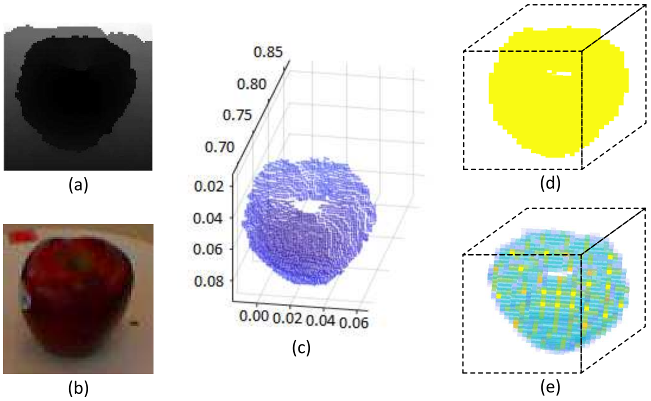

**FIGURE 1.** Volumetric representations of an apple sample. (a) The raw depth image. (b) Corresponding RGB image for visualization purpose. Our approach uses only depth images. (c) Related point cloud view. (d) Volumetric binary grid. (e) Volumetric intensity grid.

rotational approach increases category recognition accuracy significantly on both datasets, while our single-rotational approach produces promising results.

## III. PROPOSED APPROACH

The proposed approach uses only raw depth data obtained from the Microsoft Kinect. These raw depth data are noisy and have missing values caused by reflections, transparency of surfaces, etc [28]. Thus, we apply the denoising method of [29] as a pre-processing step to get rid of noise in the point cloud. Our approach consists of two main steps. First, we generate volumetric representations from raw depth images. Then, these volumetric data are fed to a 3D CNN to classify object categories.

### A. VOLUMETRIC REPRESENTATION

Since the perceptual structures of RGB and depth images are different, geometric silhouettes of objects hidden in depth data may not be fully revealed if depth data are used as an additional channel along with the RGB channels (e.g. [6], [13]). Furthermore, volumetric representations have advantages in CNN architectures. Unlike point clouds and meshes, their simplicity without the need of keeping spatial neighborhood relations and convenience to convolutional approaches are key factors in their use. Thus, we propose two straightforward and effective volumetric representations in this study. We construct these representations by projecting point cloud data to 3D matrix space in which each cell represents a voxel.

#### 1) Binary Grid

In binary grid model, each voxel has a binary state in which the existence of a surface point is represented. A voxel value of *1* means that a surface point exists in the representative space whereas *0* indicates the absence. For a given point cloud data $P = \{p_1, p_2, \ldots, p_m\}$, each point is represented as $p_n = \{x_n, y_n, z_n\}$ where $x_n, y_n, z_n$ values represent the 3D coordinates on $x, y, z$ axises in point cloud data respectively. $m$ denotes the number of points in the point cloud. Then, the transformation from point cloud data to volumetric grid is performed as follows:

$$\begin{aligned} x'_n &= \left(\frac{x_n - x_{min}}{(x_{max} - x_{min}) + \epsilon}\right)(t_{max} - t_{min}) + t_{min} \\ y'_n &= \left(\frac{y_n - y_{min}}{(y_{max} - y_{min}) + \epsilon}\right)(t_{max} - t_{min}) + t_{min} \quad (1) \\ z'_n &= \left(\frac{z_n - z_{min}}{(z_{max} - z_{min}) + \epsilon}\right)(t_{max} - t_{min}) + t_{min} \end{aligned}$$

Here, $x'_n, y'_n, z'_n$ represent the projected voxel position in the grid corresponding to $p_n$. The maximum and minimum values of $x$, $y$, $z$ axises of the point cloud data are represented as $(x_{max}, x_{min})$, $(y_{max}, y_{min})$, and $(z_{max}, z_{min})$ respectively. $t_{max}$ and $t_{min}$ are the maximum and minimum projection values of the volumetric grid, which are *30* and *1* respectively in this study. The constant $\epsilon \approx 0$ in denominator is used to prevent division by zero when max and min coordinate values on an axis of point cloud are equal. The values of $x'_n, y'_n, z'_n$ are rounded to closest integer values to obtain discrete values between $t_{min}$ and $t_{max}$. In order to preserve relative positions of points to each other, $x'_n, y'_n, z'_n$ values are separately calculated for each data file. Assume that $v_{ijk}$ represents the voxel of the grid on $i, j, k$ coordinates. Then, the value of $v_{ijk}$ is calculated as follows:

$$\tau_{ijk} = \begin{cases} 1, & \text{if there is a } p_n \text{ s.t. } i = x'_n, j = y'_n, k = z'_n \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

#### 2) Intensity Grid

Binary grid is a simple model which represents whether there is a surface point in a voxel. However, many point cloud values might fall into the same voxel, which cannot be represented in binary grids. Therefore, the volumetric intensity grid keeps how many points exist in a voxel, instead of keeping presence/absence of a point in a voxel. In each voxel, an intensity value is calculated with respect to the number of points projected into that voxel. Thus, more detailed information about object shapes can be obtained with this model. To this end, each voxel's value is initialized with zero and the number of points falling into a voxel is updated as in (3).

$$\tau_{ijk} = \begin{cases} \tau_{ijk} + 1, & \text{if } i = x'_n, j = y'_n, k = z'_n \text{ for } p_n \\ \tau_{ijk}, & \text{otherwise} \end{cases} \quad (3)$$

Fig. 1 illustrates the structures of volumetric representations . The values in the binary grid are represented in yellow. For the intensity grid, the colors become darker as the intensity values increase.

### B. 3D CNN

After the construction of volumetric grids, we employ a 3D CNN to classify object categories. CNNs effectively achieve classification of visual data by considering the spatial arrangement of the raw input data in a hierarchical manner. To

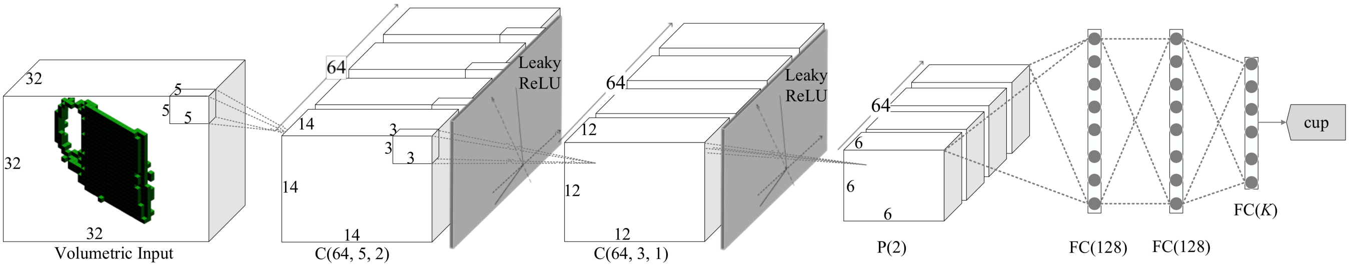

**FIGURE 2.** Network architecture of the proposed approach. The input layer accepts $32 \times 32 \times 32$ volumetric representations obtained from depth images. The convolutional layers have $64$ filters with $5 \times 5 \times 5$ and $3 \times 3 \times 3$ sizes followed by a leaky ReLU. The pooling layer downsamples the input volume by a factor of $2$ in each direction with max values. The last layers are fully-connected layers with $128$, $128$ and $K$ (number of class) unit numbers respectively.

this end, we propose a 3D CNN architecture which extends VoxNet [10]. In order to help the model generalize better and mitigate overfitting, we modify the baseline architecture by doubling the number of filters and adding dropout layers and one more fully-connected layer. We keep dropouts with the lower probabilities in the earlier layers to gain additional performance as suggested in [30]. We choose the proposed network among many different empirically evaluated alternatives by considering both the number of parameters and the accuracy results accomplished in our experiments. The overview of the proposed network architecture is illustrated in Fig. 2.

The basic components of the network are formed with; $(I)$ Convolution layer $C(k, w, s)$ applies $k$ filters of size $w \times w \times w$ with a stride of $s$ voxels. $(II)$ Pooling layer $P(w)$ summarizes statistics of a larger input in a lower dimension by a factor of $w$ along the spatial dimensions with their max values. $(III)$ Fully-connected layer $FC(K)$ has $K$ output neurons and each neuron has full connections to all activations of the previous layer. Apart from these, a dropout layer [30] with $p$ probability factor of dropping is used to prevent overfitting and leaky ReLU [31] is used as an activation function to obtain non-linearity after each convolution layer. Using these components, the CNN architecture is formed with two convolutional layers followed by the leaky ReLU, a pooling layer, and three fully-connected layers as final layers. The input layer accepts $32 \times 32 \times 32$ volumetric data including $1$ extra padding in each direction to reduce convolution artifacts. The convolution layers have $64$ filters of size $5 \times 5 \times 5$ and $3 \times 3 \times 3$, stride size $2$ and $1$ respectively. The initializations of convolutional layers are performed with the method of [32] and the outputs pass through a leaky ReLU with parameter $0.01$. The pooling layer summarizes the input volume by a factor of $2$ along the spatial dimensions with maximum values. Three fully connected layers are initialized from a zero-mean Gaussian with $\sigma = 0.01$ and have $128$, $128$ and $K = 51$ or $K = 14$ (number of classes) unit numbers respectively. Dropout layers are employed after the first convolution layer, the pooling layer, the first and the second FC layers with $0.2$, $0.3$, $0.4$, and $0.5$ dropping probability factors respectively. In order to minimize the objective function during training, Stochastic Gradient Descent (SGD) is used with $L2$ regularization and momentum optimization forms. The regularization and the momentum parameters are $0.001$ and $0.9$ respectively. The size of batch is 32. The learning rate is initialized with $0.001$ and has a decreasing policy by a factor of 10 over time. Finally, we augment the data during training by adding randomly mirrored and shifted instances as a common practice for CNNs.

### C. MULTI-ROTATIONAL APPROACH

In this work, we also bring multiple rotations of instances together to obtain rotational invariance as in the ModelNet [9]. However, unlike the 3D CAD models of ModelNet, we employ Kinect's depth images, which do not provide complete 3D models. We present an approach that compiles the information in multiple views of an object to provide rotational invariance. Fig. 3 shows the proposed multi-rotatinal approach at testing time. This approach is related to rotation augmentation where each jittered copy is added during training to learn rotational invariants. This can be seen as an implicit interpretation of expanding the local connectivity of filters along the rotations and sharing weights across the rotations. The activations of the output layer are pooled over the rotations at testing time. The network takes multiple rotations of objects and a final voting approach is performed on score values to identify the object class. Algorithm 1 briefly explains recognition of an object category using multiple rotations at testing time.

**Algorithm 1:** Object category recognition using multi-rotational depth data

```
input  : $I$ input rotations, $C$ categories
output: predicted category label
for each $I_r \in I$ do
    $S_r \leftarrow$ 3D-CNN($I_r$) {get score values for $I_r$}
end
for each category $c \in C$ do
    $F_c \leftarrow 0$ {holds the final score of the category $c$}
    for each rotation $I_r \in I$ do
        $F_c \leftarrow F_c + S_c(r)$
    end
end
return category $c$ of maximum score over $F$
```

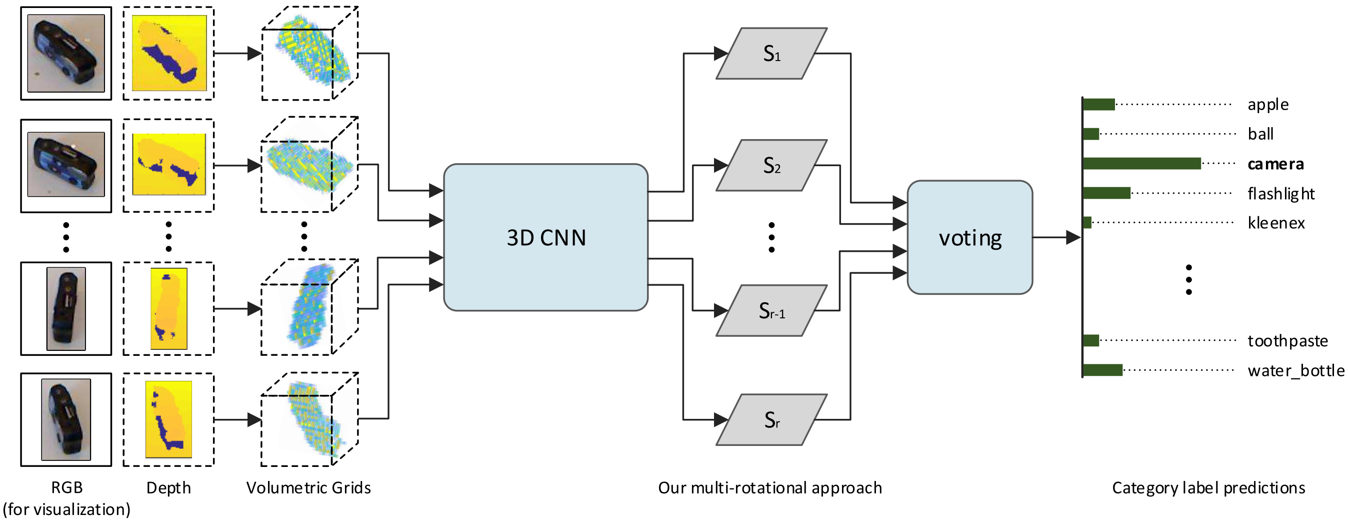

FIGURE 3. The proposed multi-rotational object recognition method. The network accepts multiple rotations of objects and produce $S_r$ score values for object categories. Then the category label is estimated with a final voting step (see the text for a detailed explanation). The left column shows RGB images for visualization purposes. The method uses only depth data.

## IV. EXPERIMENTAL EVALUATION

The proposed approach has been evaluated on two standard benchmarks, the Washington RGB-D [12] and the 2D3D Object [15] datasets. We first describe the datasets with experimental setups, then we evaluate experimental results with analysis. Finally, we compare the performance of the proposed approach with the previous methods that use depth information on both the Washington RGB-D and the 2D3D Object datasets. Results of the other methods are taken from the original publications.

### A. DATASET AND SETUP

#### 1) The Washington RGB-D Dataset

The Washington RGB-D has *51* object categories with a total of *207*, *662* images under *300* category instances. Each object contains frames of three video sequences recorded at three elevation angles (*30*°, *45*°, and *60*°). When evaluating different variations of our volumetric representations (Section IV-B1), we randomly divide the entire dataset into three parts: *60*% training, *20*% validation, and *20*% testing splits. For these experiments, we train our network up to *120* epochs.

When making comparisons with other studies, we follow the experimental setup in [12] and sub-sample the dataset by taking every *5*th depth image to have around *41*, *500* images. We use the best-performing representation of the first experiment with single rotation of each input instance. The *10* splits of Lai *et al.* [12] are used and the average result is reported. In each split, one instance of each category is selected for testing and the remaining instances are used for training. We train our network up to *820* epochs.

#### 2) The 2D3D Object Dataset

The 2D3D Object dataset consists of *155* instances organized into *14* categories. Each instance has a total of *36* views recorded in *10*° increments. We conduct experiments by following the settings in [15]. We subsample every *2*nd image of the dataset and reduce the dataset size to a total of *2790* images. Then, *6* instances per category are randomly selected for training and the remaining instances are used for testing. The only category with less than *6* instances is *Scissor* and for that we split *4* instances for training and *1* instance for testing. Thus, the training set consists of *82* instances with *1476* images and the testing set contains *73* instances with *1314* images[1]. We train the network up to *5000* epochs.

#### 3) Multi-rotational Setup

The above mentioned setups use a single depth image (single-rotational) for object category recognition. We also setup the datasets for our multi-rotational approach. The Washington RGB-D contains images of instances in different elevation angles, out of order. We construct a new dataset as a subset of the Washington RGB-D by bringing images from the *30*°, *45*°, and *60*° sequences for each instance into order. We choose *18* rotations for each sequence and a total of *16*, *200* images. Then, we follow the experimental setup in [12] by randomly leaving one object instance out from each category for testing, and train the network on the remaining $300 - 51 = 249$ instances at each trial. We use the *10* testing splits provided by Lai *et al.* [12] and report the average

[1] Browatzki *et al.* refers the splits as *82* and *74* for training and testing respectively. However, a total of *155* instances are provided on http://www.kyb.mpg.de/~browatbn. The corresponding author confirmed that there are *13* samples in the Cup category instead of *14* samples, and so there are *155* categories in total.

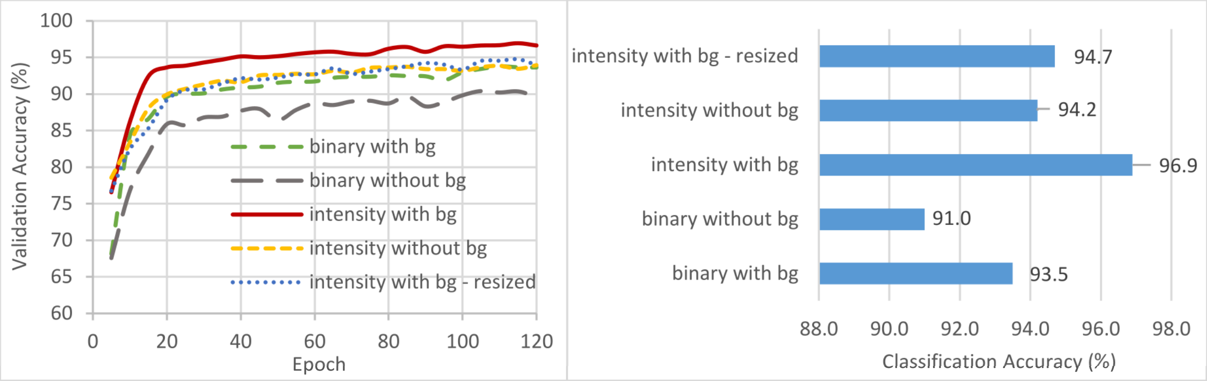

**FIGURE 4.** Effect of volumetric grids with various parameters on the Washington RGB-D dataset (bg is the abbreviation of background).

accuracy of the *10* splits. We train the network up to *820* epochs.

In the 2D3D Object dataset, images of each instance are already arranged in a sequential order with *10°* increments. Thus, we use this dataset as is for the multi-rotational approach. We use *18* rotations of each instance by subsampling every *2*nd image. As in the single-rotational approach, we use the setting in [15] and train the network up to *5000* epochs.

### B. RESULTS

#### 1) Volumetric Grid Variations

We first evaluate the effects of volumetric grids with and without object masks using the Washington RGB-D dataset. We also investigate the effect of resizing input images to see the potential performance degradation due to cropping and warping operations. Fig. 4 shows the results including validation accuracy and testing performance. We can see that the volumetric intensity grid performs better than the binary grid. This confirms that point intensity at a voxel gives more information for a better classification, rather than considering the presence/absence of a point projected into a voxel. We also show that our approach is able to deal with the background clutter by conducting experiments in which no object masks are used. Against expectations, we found that using object masks negatively affect classification performance. The most important reason is the imperfections of object masks provided with the dataset. Lastly, we evaluate the effect of resizing process used in CNNs on classification. Since CNNs require fixed-size input images, this may cause performance degradation by cropping and warping images as stated in [19]. For this purpose, we construct the volumetric grids from the resized input images with a scale of $148 \times 148$. We see that the classification performance falls as we expected. This shows that volumetric representations overcome the limitation of fixed-size inputs, since the projection operation does not require equal sized images. As a result of these experiments, we use the best performing volumetric intensity grid with the background combination in the rest of the experiments.

#### 2) Comparison on the Washington RGB-D Dataset

Table 1 reports the comparison of results on the Washington RGB-D dataset. Despite the fact we use the same setup in our multi-rotational and single-rotational approaches, we employ the multi-rotational approach to the constructed subset of the Washington RGB-D dataset. Thus, we give the result of our multi-rotational approach with $*$ label to indicate this. In addition, the proposed multi-rotational approach is the only method that takes multiple input images. Therefore, the result is given as a separate row in the table. The multi-rotational approach boosts recognition performance significantly by providing rotational invariance in spite of reduced data size. For the single-rotational approach, despite the view-based incomplete volumetric representation, the proposed approach achieves competitive performance and outperforms all the other methods except [19] and [20]. The state-of-the art result of Zaki *et al.* [20] is achieved by taking advantage of different data modalities among RGB images, depth maps, and point clouds to capture object features. However, their embedded point cloud representations utilize color information. Thus, this method does not use pure depth information unlike the other methods. This situation is denoted with the † indicator in the table. Their isolated experiment using depth images has 79.4% recognition accuracy. Cheng *et al.* [19] apply their CNN-SPM-RNN method to learn features not only from the depth images but also from the surface normals separately and concatenate them to represent the final depth view. In contrast, the proposed method only uses raw depth images.

Fig. 5 shows f-scores of the individual object categories in the Washington RGB-D dataset for both of the single-rotational and the multi-rotational approaches. The use of multiple rotations generally improves the results. Basically, there are several object categories with low results both in the single-rotational and the multi-rotational approaches. These are *camera*, *mushroom*, and *peach* object categories. The common problem of these object categories is that they contain the minimum number of instances with only three instances. This is quite small considering the other categories with up to 14 instances. This imbalance in category instances reduces the success for individual objects with small number

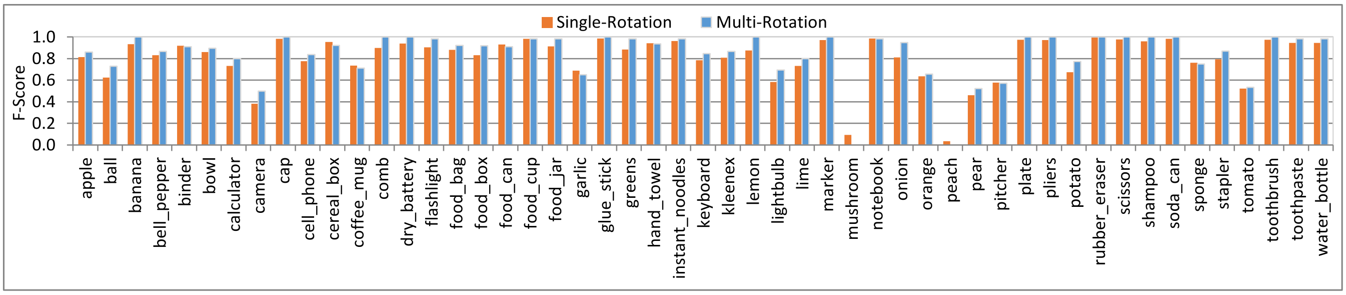

**FIGURE 5.** F-Scores for each category on the Washington RGB-D dataset.

**TABLE 1.** Accuracy comparison of category recognition on the Washington RGB-D dataset using depth data.

| Method | Accuracy(%) |
|---|---|
| Kernel SVM [12] | 64.7 ± 2.2 |
| HKDES [14] | 75.7 ± 2.6 |
| SSL [18] | 77.7 ± 1.4 |
| KDES [13] | 78.8 ± 2.7 |
| CNN-RNN [6] | 78.9 ± 3.8 |
| HMP [33] | 81.2 ± 2.3 |
| Subset-RNN [17] | 81.8 ± 2.6 |
| Volumetric [11] | 82.0 ± 2.3 |
| CNN-SPM-RNN [19] | 83.6 ± 2.3 |
| Hypercube [20] † | 85.0 ± 2.1 |
| **This work (single-rotational)** | 82.4 ± 2.2 |
| **This work (multi-rotational)** * | 85.9 ± 2.9 |

of instances as well as the overall success of the system. Because the imbalance of the dataset biases the learning with the categories having more number of instances. In addition to this, the other reasons for poor performance are intra-class variation and inter-class similarity in the dataset. On the other hand, samples with shiny surfaces such as *camera* may corrupt depth information since the depth sensors do not properly handle reflections from such surfaces.

Examples of misclassified object categories on the Washington RGB-D dataset are presented in Fig. 6. The first column (*a*) in the figure shows volumetric representations of the samples. The second column (*b*) shows the corresponding RGB images of the samples. In the last column (*c*), an example is given from the mispredicted object categories. RGB images in the figure are given for convenience (column *b* and *c*). We use only the volumetric grid obtained from depth images for classification, as depicted in the first column (*a*). As it can be seen from the figure, shapes of misclassified objects are actually very similar to each other, which is the main reason of misclassification. From top to bottom, *mushroom* classified as *ball*, *peach* as *orange*, *peach* as *apple*, *pear* as *apple*, *tomato* as *potato*, and *camera* as *sponge*.

### 3) Comparison on the 2D3D Object Dataset

Table 2 shows the accuracy comparison of the proposed approach with the previous works on the 2D3D Object dataset. We apply both the single-rotational and the multi-rotational approaches on this dataset with the same setup as described in Sec. IV-A. Despite the small size of the dataset, our method records a significant performance for both approaches. The multi-rotational approach achieves the best result by combining multiple rotations of an object. However, as we stated in Sec. IV-B2, the proposed multi-rotational approach is the only method that utilizes multiple input images. Hence, we present this result in a separate row.

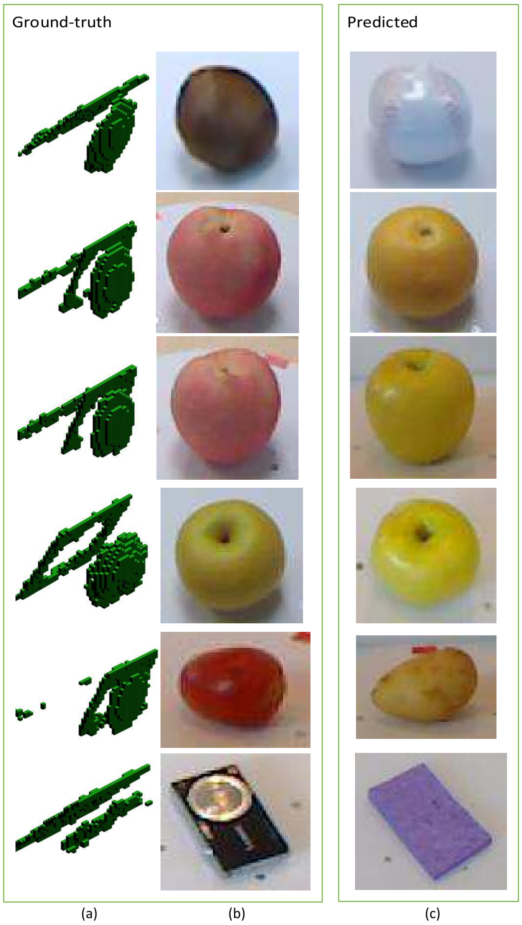

**FIGURE 6.** Misclassification examples on the Washington RGB-D dataset. RGB images are given for illustration purposes. (a) Volumetric representations of confused examples. (b) RGB views of the ground-truth examples. (c) Sample RGB images from the mispredicted object categories.

**TABLE 2.** Accuracy comparison of category recognition on the 2D3D Object dataset using depth data.

| Method | Accuracy(%) |
|---|---|
| Browatzki *et al.* [15] | 74.6 |
| HMP [33] | 87.6 |
| CNN-SPM-RNN [19] | 89.4 |
| Subset-RNN [17] | 90.2 |
| Hypercube [20] † | 91.6 |
| **This work (single-rotational)** | 90.1 |
| **This work (multi-rotational)** | 94.5 |

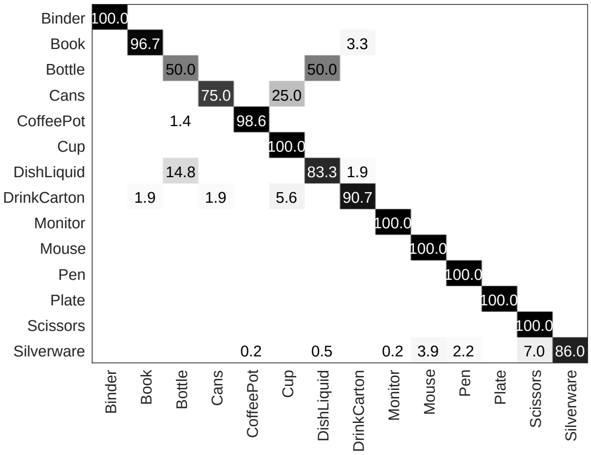

**FIGURE 7.** Confusion matrix of the single-rotational approach on the 2D3D Object dataset.

On the other hand, the proposed single-rotational approach achieves the highest recognition accuracy with Subset-RNN [17] after Hypercube [20]. Since the Hypercube [20] takes advantage of color information in the embedded point cloud, the result is presented with the † indicator. Therefore, an extra 1.5% success rate of this method is reasonable.

The confusion matrices of our single-rotational and multi-rotational approaches over the 14 categories of the 2D3D Object dataset are shown in Fig. 7 and Fig. 8 respectively. As shown in the figures, most categories are correctly classified. Considering the results in the confusion matrices, *Bottle* and *Cans* have the highest error rate. *Bottle*s are confused with *DishLiquid*s while *Cans* are confused with *Cup*s. As in the Washington RGB-D dataset, these misclassifications are mainly due to shape similarities.

## V. CONCLUSION

In this paper, we have proposed a volumetric object recognition approach based on 3D CNNs and two volumetric representations of depth data. Despite the limitations of incomplete 3D models of Kinect's depth images, we have shown the effectiveness of the proposed approach on two common benchmarks, the Washington RGB-D and the 2D3D Object datasets. While our multi-rotational approach produces the best results, our single-rotational approach competes with the state-of-the-art methods. The multi-rotational approach can be useful in robotics to overcome viewpoint-related ambiguities. A robot can change its viewpoints in the environment and collect multiple images of objects to increase recognition accuracy using the proposed approach. We believe the proposed method can be useful for robotic vision applications including semantic mapping and recognition tasks. In the future work, we plan to combine color information with depth information on 3D models to resolve ambiguities caused by similar shaped objects and increase classification accuracy.

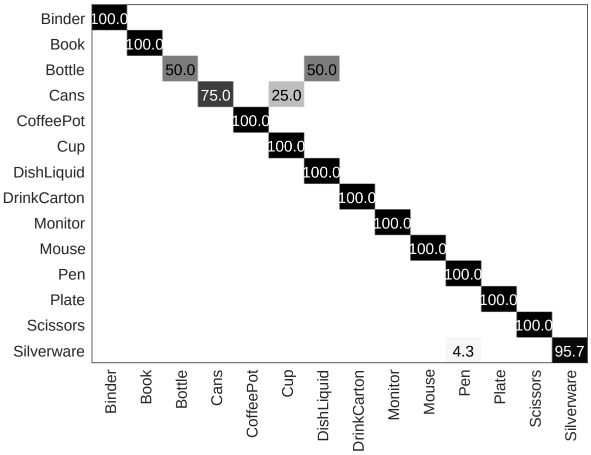

**FIGURE 8.** Confusion matrix of the multi-rotational approach on the 2D3D Object dataset.

## REFERENCES

[1] D. Marr, Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. New York, NY, USA: Henry Holt and Co., Inc., 1982. ↑1

[2] R. Nevatia and T. O. Binford, "Description and recognition of curved objects," Artificial Intelligence, vol. 8, no. 1, pp. 77 – 98, 1977. ↑1

[3] D. Marr and H. K. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," Proceedings of the Royal Society of London B: Biological Sciences, vol. 200, no. 1140, pp. 269–294, 1978. ↑1

[4] R. A. Brooks, "Symbolic reasoning among 3-d models and 2-d images," Artificial Intelligence, vol. 17, no. 1, pp. 285 – 348, 1981. ↑1

[5] M. Blum, J. T. Springenberg, J. Wülfing, and M. Riedmiller, "A learned feature descriptor for object recognition in rgb-d data," in Robotics and Automation (ICRA), 2012 IEEE International Conference on, May 2012, pp. 1298–1303. ↑1, ↑2

[6] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3d object classification," in Advances in Neural Information Processing Systems, 2012, pp. 665–673. ↑1, ↑2, ↑3, ↑7

[7] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on. IEEE, 2011, pp. 127–136. ↑1

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105. ↑1

[9] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in Proceedings of

the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1912–1920. ↑1, ↑2, ↑4

[10] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. IEEE, 2015, pp. 922–928. ↑1, ↑2, ↑4

[11] A. Caglayan and A. B. Can, "3d convolutional object recognition using volumetric representations of depth data," in Machine Vision Applications (MVA), 2017 Fifteenth IAPR International Conference on. IEEE, 2017, pp. 125–128. ↑2, ↑7

[12] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011, pp. 1817–1824. ↑2, ↑5, ↑7

[13] L. Bo, X. Ren, and D. Fox, "Depth kernel descriptors for object recognition," in 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2011, pp. 821–826. ↑2, ↑3, ↑7

[14] L. Bo, K. Lai, X. Ren, and D. Fox, "Object recognition with hierarchical kernel descriptors," in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011, pp. 1729–1736. ↑2, ↑7

[15] B. Browatzki, J. Fischer, B. Graf, H. H. Bülthoff, and C. Wallraven, "Going into depth: Evaluating 2d and 3d cues for object classification on a new, large-scale object dataset," in Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on. IEEE, 2011, pp. 1189–1195. ↑2, ↑5, ↑6, ↑8

[16] S. Tang, X. Wang, X. Lv, T. X. Han, J. Keller, Z. He, M. Skubic, and S. Lao, "Histogram of oriented normal vectors for object recognition with a depth sensor," in Asian conference on computer vision. Springer, 2012, pp. 525–538. ↑2

[17] J. Bai, Y. Wu, J. Zhang, and F. Chen, "Subset based deep learning for rgb-d object recognition," Neurocomputing, vol. 165, pp. 280–292, 2015. ↑2, ↑7, ↑8

[18] Y. Cheng, X. Zhao, K. Huang, and T. Tan, "Semi-supervised learning for rgb-d object recognition." in ICPR, 2014, pp. 2377–2382. ↑2, ↑7

[19] Y. Cheng, X. Zhao, K. Huang, and T. Tan, "Semi-supervised learning and feature evaluation for rgb-d object recognition," Computer Vision and Image Understanding, vol. 139, pp. 149–160, 2015. ↑2, ↑6, ↑7, ↑8

[20] H. F. Zaki, F. Shafait, and A. Mian, "Convolutional hypercube pyramid for accurate rgb-d object category and instance recognition," in Robotics and Automation (ICRA), 2016 IEEE International Conference on. IEEE, 2016, pp. 1685–1692. ↑2, ↑6, ↑7, ↑8

[21] A. Garcia-Garcia, F. Gomez-Donoso, J. Garcia-Rodriguez, S. Orts-Escolano, M. Cazorla, and J. Azorin-Lopez, "Pointnet: A 3d convolutional neural network for real-time object class recognition," in Neural Networks (IJCNN), 2016 International Joint Conference on. IEEE, 2016, pp. 1578–1584. ↑2

[22] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004. ↑2

[23] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," Computer vision and image understanding, vol. 110, no. 3, pp. 346–359, 2008. ↑2

[24] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1. IEEE, 2005, pp. 886–893. ↑2

[25] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," IEEE Transactions on pattern analysis and machine intelligence, vol. 21, no. 5, pp. 433–449, 1999. ↑2

[26] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in Computer vision and pattern recognition, 2006 IEEE computer society conference on, vol. 2. IEEE, 2006, pp. 2169–2178. ↑2

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 248–255. ↑2

[28] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," IEEE transactions on cybernetics, vol. 43, no. 5, pp. 1318–1334, 2013. ↑3

[29] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3d point cloud based object maps for household environments," Robotics and Autonomous Systems, vol. 56, no. 11, pp. 927–941, 2008. ↑3

[30] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." Journal of machine learning research, vol. 15, no. 1, pp. 1929–1958, 2014. ↑4

[31] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in Proc. ICML, vol. 30, no. 1, 2013. ↑4

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034. ↑4

[33] L. Bo, X. Ren, and D. Fox, "Unsupervised feature learning for rgb-d based object recognition," in Experimental Robotics. Springer, 2013, pp. 387–402. ↑7, ↑8

ALI CAGLAYAN received the B.S. degree from the Department of Computer Engineering at Hacettepe University in 2009. He is currently pursuing the Ph.D. degree at the same department. His research is in the area of computer vision and machine learning. He is particularly interested in RGB-D object recognition with deep learning techniques.

AHMET BURAK CAN received his B.Sc. and M.Sc. degrees from Hacettepe University, Department of Computer Science and Engineering in 1998 and 2001 respectively. He received his Ph.D. degree from Purdue University, Department of Computer Science in 2007. He is affiliated with Hacettepe University since 1998. His primary research interests include computer vision, information security, and distributed systems.