Exploiting Multi-Layer Features Using a CNN-RNN Approach for RGB-D Object Recognition

Ali Caglayan[1,2][0000−0002−3408−8659] and Ahmet Burak Can[2][0000−0002−0101−6878]

[1] Department of Computer Engineering, Bingol University, Bingol, Turkey
[2] Department of Computer Engineering, Hacettepe University, Ankara, Turkey
{alicaglayan,abc}@cs.hacettepe.edu.tr

Abstract. This paper proposes an approach for RGB-D object recognition by integrating a CNN model with recursive neural networks. It first employs a pre-trained CNN model as the underlying feature extractor to get visual features at different layers for RGB and depth modalities. Then, a deep recursive model is applied to map these features into high-level representations. Finally, multi-level information is fused to produce a strong global representation of the entire object image. In order to utilize the CNN model trained on large-scale RGB datasets for depth domain, depth images are converted to a representation similar to RGB images. Experimental results on the Washington RGB-D Object dataset show that the proposed approach outperforms previous approaches.

Keywords: Convolutional Neural Network · Recursive Neural Network · Transfer Learning · RGB-D Object Recognition.

1 Introduction

The prevalence of depth sensors has led to an increasing attention in developing numerous applications in computer vision and robotics. RGB-D object recognition is a challenging fundamental task among these applications. In the meantime, deep learning based methods have surpassed the conventional feature extraction based methods and dominated the field. The breakthrough of convolutional neural networks (CNNs) has enabled to replace hand-engineered feature representations with efficient transferable off-the-shelf features. Deep features have been the focus of various research efforts including object recognition (e.g. [29], [27]), detection (e.g. [16], [28]), and semantic segmentation (e.g. [16], [15]), since they offer biologically-inspired valuable information at hand. A common approach among these methods is to use the features extracted from the final fully-connected layers. The main reason behind this is that these features provide object-specific semantic information with smaller dimensions. However, as moving towards the final layers, it has been observed that these features are increasingly dependent on the chosen dataset and task [35]. On the other hand, the earlier layers capture distinctive information about the task and provide locally-activated features which are less sensitive to semantics [18,36]. One challenge of the earlier layers is the high dimensionality of features extracted from them. Consequently, features are transformed from general to specific throughout the network and the relational interest information is distributed across the network at different levels [35,18]. However, it is unclear how to exploit the information effectively.

The purpose of this paper is to develop a reliable deep feature learning approach to obtain more accurate classification of RGB-D objects by combining two key insights. The first is to employ a pre-trained CNN as a feature extractor and utilize information at different layers of the network to yield better recognition performance. The second is to apply recursive neural networks (RNNs) to reduce the dimensionality of the features and encode the CNN activations in robust hierarchical feature representations. The idea of combining a trained CNN model with the RNN structure is first presented in [8] for RGB image classification. After carrying out several experiments, the authors find that the activation weights from the 4th layer of the pre-trained network in [9] transformed by RNNs are more suitable and robust for RGB image classification. Our aim in this work is to improve on this idea by gathering feature representations at different levels in a compact and representative feature vector for both RGB and depth data. However, unlike [8], we reshape the activation maps of each layer to give the multiple RNNs in order to reduce the feature dimension. This provides a generic structure for each layer by fixing the tree structure without hurting performance and it allows us to improve recognition accuracy by combining feature vectors at different levels. The incorporation of multiple fixed RNNs together with the pre-trained CNN model allows feature transition at different layers to preserve both semantic and spatial structure of objects. Additionally, we embed depth data into the RGB domain by using surface normals in order to transfer information from a CNN model trained on the ImageNet dataset [12]. To this end, depth maps are colorized by computing three dimensional surface normals and treating each dimension as a color channel. The information from RGB images and depth maps are fused to obtain final RGB-D classification results. The proposed method is then evaluated and compared with the current state-of-the-art methods on the popular Washington RGB-D Object dataset [23] in terms of classification accuracy. The experimental results show the effectiveness of the proposed method both in terms of feature dimensions and classification accuracy. Hence, the contributions of this paper cover the following issues (The source code for our approach is available at: https://github.com/acaglayan/exploitCNN-RNN):

1. We present a novel deep feature learning pipeline which encodes information at different layers by incorporation of RNNs with a pre-trained CNN model for RGB-D object categorization.
2. We investigate features produced by a pre-trained CNN model and our pipeline. We show that RNNs represent activation maps of CNNs in a lower-dimensional space without hurting performance and allows us to encode

information at multiple levels to get further hiearchical compact representations.

3. We define a way to allow transfer learning for depth data from a CNN model trained on RGB images. To do that, we compute surface normals from depth maps and normalize them. Despite the characteristic difference between depth and RGB data, the results suggest that a pre-trained CNN on RGB images can effectively capture information from depth images in this way.
4. We provide experimental evidence showing our method improves the state-of-the-art results on the Washington RGB-D Object dataset for category recognition.

2 Method

Encouraged by the recent tremendous advances in deep learning techniques, in this work, we explore the effectiveness of using a pre-trained CNN model together with RNNs to recognize object categories for RGB-D data. Specifically, we employ the pre-trained CNN model in [9] called VGG-f, which has been widely used for object recognition (e.g. [36,8]). In order to leverage the power of CNNs pre-trained over the large-scale RGB datasets such as ImageNet [12] for depth data, we pre-process the depth inputs to encode three color channels at each pixel. We first compute the three-dimensional surface normals from depth maps in which each dimension represents a color channel. Then, the channels are scaled to map values to the 0 - 255 range.

The structure of our approach is shown in Fig. 1. The proposed approach includes a two-step hierarchical feature learning procedure. In the first step, activation maps are extracted from the pre-trained CNN model at different levels to capture useful translational invariant features. Then, these activation maps are reshaped to reduce dimensions and given to the multiple fixed-tree RNNs to learn hierarchical high-level features of the images. To learn these features, we adapt the proposed work by Bui et al. [8]. They use RNNs with a pre-trained CNN model for feature extraction in an RGB-D object benchmark using only RGB images. The key of their approach is giving the output activation maps of a single intermediate layer as is to the recursive network structure. In contrast to this setting, we however want to efficiently combine features at multiple levels to obtain complementary different feature patterns for both RGB and depth images. Therefore, we modify the baseline framework in several ways. We first reshape the activation maps of the CNN model to cope with the high dimensionality of the produced feature vector of RNNs. This allows us to capture information at different layers for further classification performance. As such, multiple layers provide a compact and representative feature vector for each object class. Secondly, we compute surface normals from depth maps and encode to the RGB color modality to make use of the large-scale RGB dataset of ImageNet for depth modality by transfer learning. Finally, we combine the final feature vector of RGB and depth streams to build highly accurate RGB-D object category recognition method.

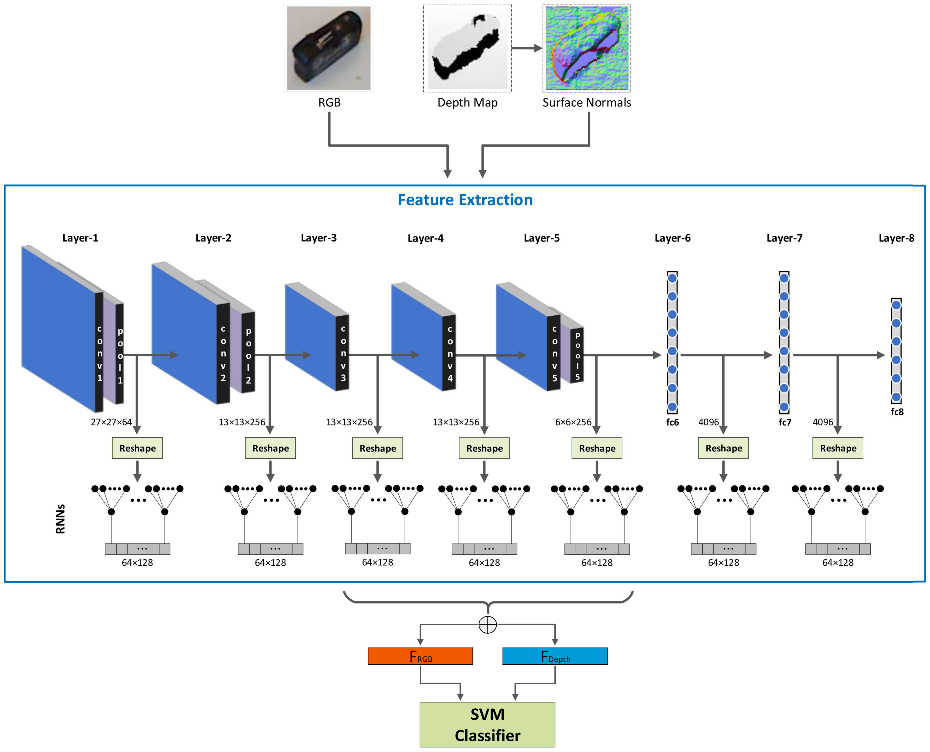

Fig. 1. Overview of the proposed method. The inputs of our approach are RGB images and colorized surface normal images. A pre-trained CNN [9] is employed to extract raw features at different layers. The multiple RNNs are used to learn higher level representations on a fixed tree structure. The learned representations at different layers are fused for final feature vectors of RGB and depth domains and given to a linear SVM classifier.

We use the pre-trained VGG-f model as a feature extractor without fine-tuning. Therefore, the procedure requires no training at feature extraction stage and works fast. The network consists of 5 successive convolutional layers (each might have sub-modules including convolution, pooling, and local contrast normalization operations) followed by 3 fully-connected layers and produces a distribution over the ImageNet dataset [12]. The dimensions of activation maps obtained from each layer are $27 \times 27 \times 64$, $13 \times 13 \times 256$, $13 \times 13 \times 256$, $13 \times 13 \times 256$, $6 \times 6 \times 256$, 4096, 4096, and 1000. The final fully-connected output is the feature representations over the 1000-classes of the ImageNet. For other layers, we reshape the activations by fixing the number of filter bank sizes to 64. Thus, for example, the outputs of fully-connected layers are formed into $8 \times 8 \times 64$ dimensions, and the convolution layers with the same size are converted into $26 \times 26 \times 64$. In this way, the new structures provide a generic ease of use and reduce the size of feature vectors generated by RNNs without sacrificing performance. We refer the readers to [9] for further details of the pre-trained network structure.

After computing activations by forward propagating input images through the pre-trained CNN model, we employ RNNs, whose inputs are the outputs of the CNN, to learn compact global feature representations. RNNs are well-studied models [26,33,32] that can learn higher level representations by applying the same operations recursively on a tree structure. Each layer merges blocks of adjacent vectors into a parent vector with tied weights where the goal is to map inputs $X \in \mathbb{R}^{K \times r \times r}$ into a lower dimensional space $p \in \mathbb{R}^{K}$ through multiple layers in the end. Then, the parent vector is passed through a nonlinear squash function. In this work, we use the *tanh* function in order to preserve the original work but any squash function that provides an adequate nonlinearity may be used (e.g. the hyperbolic tangent sigmoid or the elliot sigmoid functions. See Section 4.1). A single RNN structure produces a K-dimensional vector, where K is the length of a given input (filter bank size). We use multiple randomly initialized N RNNs in our work. Therefore, a total of $(N \times K)$-dimensional final matrix is produced in the end.

The role of RNN in the process is twofold. First, it reduces the feature space dimensionality and maximizes classification performance. Thus, it allows us to transfer information from multiple layers effectively. Second, intuitively, the semantic content of the child nodes is recursively aggregated into the parent node through the structure. In this way, the resulting information represents the contextual description of the entire image. Moreover, RNNs are random-weight based architectures without requiring back-propagation. Unlike CNNs, RNNs use non-overlapping receptive fields. Specifically, the RNNs in this study are of one-level with a single parent vector. Thus, they are computationally fast.

3 Related Work

The currently dominant object recognition solutions are based on deep feature learning techniques. The key enabling factors behind this are that these techniques rely on biologically-inspired learning models that can automatically obtain relevant information from the very low tier of the inputs and the ability to optimize them for the problem at hand. Recent works have shown that a trained CNN on a large-scale dataset can effectively be used to generate good generic representations for other visual recognition tasks [29,35,2,25]. Gupta et al. [17] encode the depth information in three channels using the camera parameters of the inputs in order to utilize a pre-trained CNN model on large-scale RGB images and focus on RGB-D object detection. Schwarz et al. [27] present an approach for RGB-D object recognition and pose estimation using the *fc7* and *fc8* activations of the pre-trained CNN of Krizhevsky et al. [22]. A different related approach is proposed by Eitel et al. [13]. They employ a two-stream CNN, one for each modality of RGB and depth channels which are finally combined with a late fusion network. They initialize both streams with weights from a pre-trained network on the ImageNet [12] and fine-tune for the final classification. The recent work of [36] uses a spatial pyramid pooling strategy at different layers of the network to encode activations of all layers before feature concatenation. Their approach of aggregating information at different levels has inspired us in this work. Asif et al. extract *fc7* features from the pre-trained VGGnet model [31] for five different feature maps to encode the appearance and structural information of objects.

Other methods based on deep feature learning have also been developed. The convolutional k-means descriptor (CKM) [5] is proposed to learn features around SURF [4] interest points. The pioneer work of Socher et al. [32] has been employed for the semi-supervised method of Cheng et al. [11] to utilize grayscale images and surface normals in addition to RGB and depth images. The same work also has been used in the subset based method of Bai et al. [3] to extract patches from several subsets for filter learning. The method of convolutional fisher kernels (CFK) [10] is proposed to integrate CNNs with Fisher Kernel encoding for RGB-D object recognition. Despite its success in terms of accuracy performance, it appears to suffer from a very high-dimensional final feature vector for classification. Zia et al. [38] propose a method that learns RGB information using the pre-trained model of VGGnet [31] and depth information using 3D CNNs to fully exploit the 3D spatial information in depth images. They also propose a hybrid 2D/3D CNN model initialized with pre-trained 2D CNNs and fine-tuned later. They finally concatenate the features from this hybrid structure with the features learnt from depth-only and RGB-only architectures to feed the resulting vector to a classifier for overall recognition performance.

Recursive neural networks (RNNs) [26,33] process structured information by graphs transformed into recursive tree structures to learn distributed representations and have been used in conjuction with other architectures for various research purposes [33,32,3,11,30,21]. In [32], Socher et al. have first introduced an RGB-D object recognition method using the collaboration of CNN with RNN to first learn RGB and depth features in a separate stage and then merge for final classification. Later, this idea has been extended to replace the single CNN layer with a pre-trained CNN model by Bui et al. in [8] for RGB images. The achievement of the AlexNet-RNN [8] shows that transforming features extracted from a pre-trained CNN model by a recursive network structure can greatly increase classification accuracy in RGB object recognition. In this paper, we adapt the pipeline of [8] for RGB-D object recognition with a new structure and follow the idea of [18,36] to utilize information extracted from multi-layers. In this respect, the proposed approach learns robust representations of objects. The empirical evaluation reveals the effectiveness of the proposed approach for RGB-D object recognition by improving the accuracy performance significantly while reducing the feature dimension on the widely used Washington RGB-D Object dataset [23].

4 Evaluation

We evaluate the proposed method on the Washington RGB-D Object dataset [23]. The dataset contains $41,877$ RGB-D images of 51 household object categories and 300 instances of these categories. The experiments are carried out using the 10 train/test splits provided in [23]. For each split, there are roughly $35,000$ training images and $7,000$ test images. From each category, one instance is used for testing and all the remaining instances are used for training. All the inputs are resized to 224×224 pixels for convenience to the VGG-f model. The dataset also provides object segmentation masks. Since the background of images is fixed and simple with no cluttered view, we do not extract the background as an extra preprocessing step. Our pipeline could easily handle the background. We first evaluate experimental results with model analysis. Then, we compare the category recognition performance of our approach with several the state-of-the-art methods. We use the open-source MatConvNet toolbox [34] and the provided pre-trained CNN model with it. The obtained feature representations are classified by using a linear SVM classifier (Liblinear [14]).

4.1 Model Analysis

We analyze our approach through several model variations. We first experimentally investigate the effect of squashing functions for the RNN on accuracy performance. To this end, we use four different nonlinearities including $ReLU$, $tanh$, $tansig$, and $elliotsig$ functions. We use the same random weights to ensure a valid comparison of non-linearities. As shown in Fig. 2, the results are close to each other in general. However, there is a slight difference between the ReLU and the other nonlinearities. While the ReLU function gives better results for depth data, the others acquire better success for RGB data. Since the difference is negligible, we use the $tanh$ nonlinearity function in this study in order to preserve the original RNN work [32].

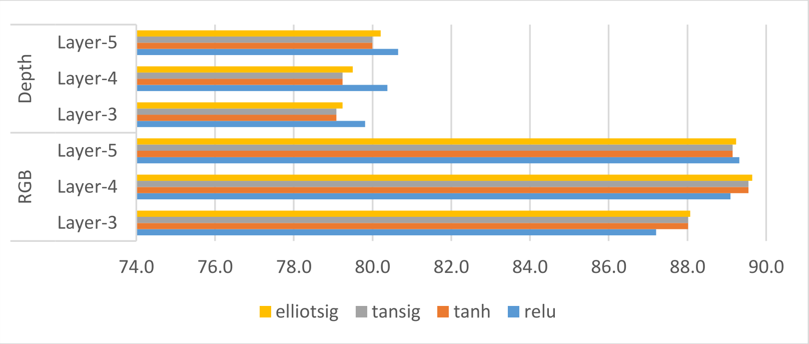

Fig. 2. Effects of different squashing functions for the RNN in terms of classification accuracy (%).

We then evaluate the effect of the RNNs on intermediate layers (*conv*3, *conv*4, and *pool*5). As can be seen from Fig. 3, the RNNs improve classification performance by significantly reducing the feature size (more than ×5 times for layers 3 and 4). The accuracy performance increases for RGB, while for depth it decreases slightly (∼1%). Nevertheless, the compact representation of the RNNs is preferable as it reduces the computational cost significantly and allows us to fuse the output of multiple layers to gain superior performance. We have chosen one of the splits as our development fold for our experiments until now. We use the all 10 splits in the rest of the experiments.

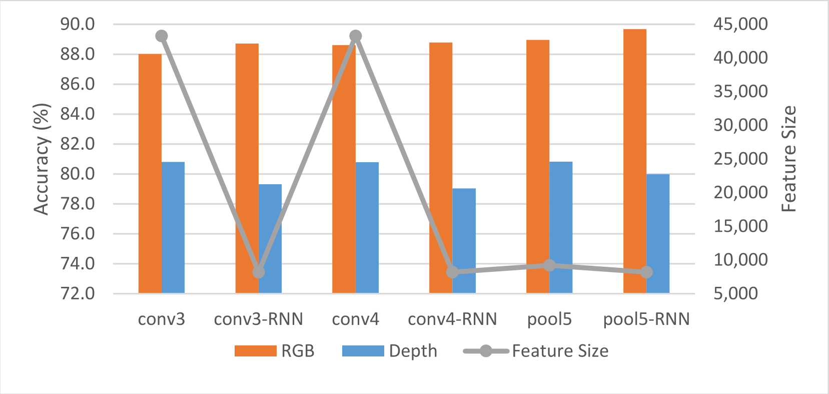

Fig. 3. Effects of RNNs in terms of accuracy performance and feature size on the mid-level raw features of the pre-trained CNN.

In our experiments, we particularly focus on the intermediate layers. The reason for this is our intuitive assumption that the outputs of the middle layers will be the optimal representations. Because it has been shown that features are eventually transformed from general to specific through deep networks [37,29]. While early layers response to low-level raw features such as corners and edges, late layers extract more object-specific features of the trained datasets. Thus, intermediate levels of the network present the optimal representations. Fig. 4 shows the average accuracy performance of each individual layer on the 10 splits together with the standard deviation. The plot verifies our assumption with a clear upward trend at the beginning and downward trend at the end.

We now move on to the empirical analysis of accuracy performance on various combinations of these mid-level representations. Table 1 presents the results demonstrating that combining feature representations at different levels significantly improves the accuracy. The combination of 4th and 5th levels for RGB gives the best accuracy, while for depth the 3th, 4th, and 5th level representations together produce the best result.

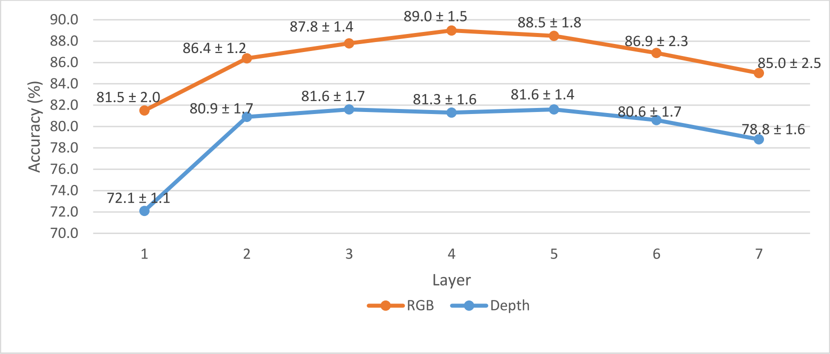

Fig. 4. Accuracy performance of our approach for individual layers.

Table 1. Accuracy performance for different combinations of the mid-level layer fusions (%).

	RGB	Depth	Feature Size
Layer3 + Layer4	89.0 ± 1.4	83.0 ± 1.7	16,384
Layer3 + Layer5	89.4 ± 1.5	83.5 ± 1.7	16,384
Layer4 + Layer5	**89.9 ± 1.6**	83.4 ± 1.7	16,384
Layer3 + Layer4 + Layer5	89.8 ± 1.5	**84.0 ± 1.8**	24,576

Finally, we fuse RGB and depth features together to evaluate combined RGB-D accuracy performance. To this end, we first consider the fusion of single layer representations that give the best results for RGB and depth separately. We then evaluate the fusion of the two layers that together provide the optimal results for RGB and depth based on the analysis in Table 1. The average accuracy results with standard deviations are reported in Table 2. Considering fusion of more layers increases the dimensionality of the feature space, which makes classification intractable with limited computational resources. Therefore, we do not consider more layers since the high accuracy advantage of our approach might fade with larger number of features.

In our experiments, we have observed that there is a slight difference between performing image normalization and non-normalization. When we apply image normalization, our best accuracy average drops by 0.2% for RGB, while it increases by 0.3% for depth. Therefore, in all experiments, we apply image normalization based on the ImageNet for depth and there is no image normalization for RGB.

Table 2. Combinations of RGB and depth data together for the final RGB-D accuracy results (RGB_4 denotes the result of 4th layer for RGB domain, $\mathrm{RGB}_{(4+5)}$ shows the fusion of 4th and 5th layers).

	Accuracy (%)	Feature Size
RGB_4 + Depth_5	92.0 ± 1.3	16,384
$\mathrm{RGB}_{(4+5)}$ + $\mathrm{Depth}_{(4+5)}$	92.5 ± 1.2	32,768

4.2 Comparative Results

In Table 3, we present accuracy results on the Washington RGB-D Object dataset, comparing our best-performing approach against several the state-of-the-art methods. The proposed method achieves the highest recognition accuracy for RGB and the combination of RGB and depth (RGB-D) data. The feature size of the AlexNet-RNN [8], which produces the closest results to our result for RGB, is twice as big as ours. As for the depth data, our approach gives quite competitive results and outperforms all the other methods except that of [36] and [10]. The Hypercube [36] utilizes color information for point cloud embedding. Thus, unlike other methods, the reported result of this method does not rely on pure depth information. The CFK [10] generates a feature set with a size of $1,568,000$ which is about $\times 64$ times larger than that of our method. On the other hand, one reason for the lower success of depth modality comparing to RGB might be that we employ a CNN model trained on the RGB dataset of ImageNet as the underlying feature extractor. This makes sense because these two data modalities have different characteristics. As a result, the proposed method learns effective discriminative deep feature representations in a fast way without requiring training and produces superior accuracy performance results.

We also present the accuracy performance of the individual object categories in Fig. 5. The results demonstrate that our approach gives high performance for most of the object categories. In general, the categories with lower results are *mushroom*, *peach*, and *pitcher* classes. The main reason for this seems to be that these categories only contain three instances, which is the minimum instance number in the dataset. Hence, this imbalance of the dataset may have biased the learning to favor of categories with more examples. In addition, intra-class variations and inter-class similarity of the object categories may make the classification difficult. In particular, the similarity of many categories in the Washington RGB-D Object dataset leads to confusion in classification. For example, the presence of many geometrically similar categories in the dataset leads to lower depth accuracy in classes such as *ball*, *lightbulb*, *lime*, *pear*, *potato*, and *tomato*. Also the depth accuracy is low in the *camera* category, whose shiny surfaces may cause corruptions in depth information. As for the RGB data, the success rate is lower in some classes, where texture information is weak in addition to the above common problems (e.g. *bowl* and *plate*).

Table 3. Accuracy comparison of our approach with the related methods on the Washington RGB-D Object dataset (%).

Method	RGB	Depth	RGB-D
Kernel SVM [23]	74.5 ± 3.1	64.7 ± 2.2	83.9 ± 3.5
HKDES [6]	76.1 ± 2.2	75.7 ± 2.6	84.1 ± 2.2
KDES [7]	77.7 ± 1.9	78.8 ± 2.7	86.2 ± 2.1
CKM [5]	-	-	86.4 ± 2.3
CNN-RNN [32]	80.8 ± 4.2	78.9 ± 3.8	86.8 ± 3.3
Subset-RNN [3]	82.8 ± 3.4	81.8 ± 2.6	88.5 ± 3.1
CNN Features [27]	83.1 ± 2.0	-	89.4 ± 1.3
MM-LRF-ELM [24]	84.3 ± 3.2	82.9 ± 2.5	89.6 ± 2.5
CNN-SPM-RNN [11]	85.2 ± 1.2	83.6 ± 2.3	90.7 ± 1.1
Hypercube [36]	87.6 ± 2.2	85.0 ± 2.1	91.1 ± 1.4
CFK [10]	86.8 ± 2.7	**85.8 ± 2.3**	91.2 ± 1.4
AlexNet-RNN [8]	89.7 ± 1.7	-	-
Fus-CNN [13]	84.1 ± 2.7	83.8 ± 2.7	91.3 ± 1.4
Fusion 2D/3D CNNs [38]	89.0 ± 2.1	78.4 ± 2.4	91.8 ± 0.9
STEM-CaRFs [1]	88.8 ± 2.0	80.8 ± 2.1	92.2 ± 1.3
This work	**89.9 ± 1.6**	84.0 ± 1.8	**92.5 ± 1.2**

5 Conclusion

We have presented a reliable deep feature learning approach using a pre-trained CNN model together with multiple-fixed RNNs to provide more accurate classification performance for RGB-D object recognition. The incorporation of RNNs with the CNN model allows us to deal with high-dimensional features and aggregate information at different layers to further leverage accuracy performance. In order to utilize the CNN models trained on large-scale RGB datasets for depth data, we colorize depth images by computing surface normals from depth maps and treat each dimension of normals as a color channel. We provide extensive experimental analysis of various parameters and comparative results on the popular Washington RGB-D Object dataset. The proposed approach produces promising performances both in terms of reduced feature dimension and high classification accuracy. There is a great potential for further improvement of the proposed approach. One potential factor that was not investigated here is fine-tuning the CNN before integrating the RNNs. Specifically, domain-specific fine-tuning might be effective for depth modality. Also, noting that the VGG-f is used as the underlying pre-trained CNN model in our approach, employing other models such as ResNet [19], DenseNet [20], etc. would be a possible future research direction to further improve accuracy performance. Training RNNs is another potential route for further improvement. Lastly, other depth coloriza-

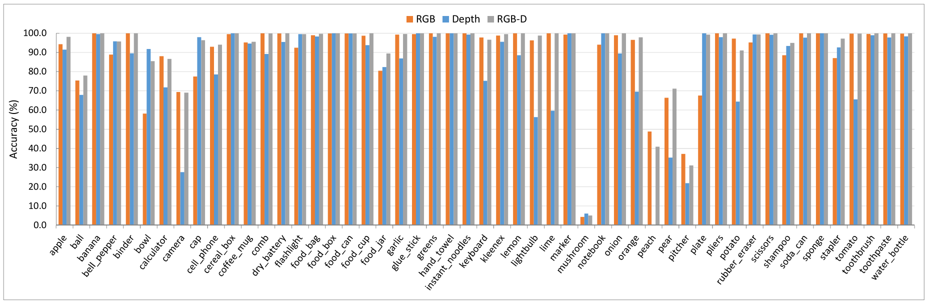

Fig. 5. Per-category success performances of our approach on the Washington RGB-D Object dataset.

tion methods and effective feature fusion techniques could also be studied in the future.

Acknowledgment

We appreciate the generous donation of graphics card by NVIDIA Corporation to support our research.

References

1. Asif, U., Bennamoun, M., Sohel, F.A.: Rgb-d object recognition and grasp detection using hierarchical cascaded forests. IEEE Transactions on Robotics **33**(3), 547–564 (2017) ↑11
2. Azizpour, H., Sharif Razavian, A., Sullivan, J., Maki, A., Carlsson, S.: From generic to specific deep representations for visual recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 36–45 (2015) ↑5
3. Bai, J., Wu, Y., Zhang, J., Chen, F.: Subset based deep learning for rgb-d object recognition. Neurocomputing **165**, 280–292 (2015) ↑6, ↑11
4. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European conference on computer vision. pp. 404–417. Springer (2006) ↑6
5. Blum, M., Springenberg, J.T., Wülfing, J., Riedmiller, M.: A learned feature descriptor for object recognition in rgb-d data. In: Robotics and Automation (ICRA), 2012 IEEE International Conference on. pp. 1298–1303. IEEE (2012) ↑6, ↑11
6. Bo, L., Lai, K., Ren, X., Fox, D.: Object recognition with hierarchical kernel descriptors. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. pp. 1729–1736. IEEE (2011) ↑11
7. Bo, L., Ren, X., Fox, D.: Depth kernel descriptors for object recognition. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 821–826. IEEE (2011) ↑11

8. Bui, H.M., Lech, M., Cheng, E., Neville, K., Burnett, I.S.: Object recognition using deep convolutional features transformed by a recursive network structure. IEEE Access **4**, 10059–10066 (2016) ↑2, ↑3, ↑6, ↑10, ↑11
9. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. arXiv preprint arXiv:1405.3531 (2014) ↑2, ↑3, ↑4, ↑5
10. Cheng, Y., Cai, R., Zhao, X., Huang, K.: Convolutional fisher kernels for rgb-d object recognition. In: 3D Vision (3DV), 2015 International Conference on. pp. 135–143. IEEE (2015) ↑6, ↑10, ↑11
11. Cheng, Y., Zhao, X., Huang, K., Tan, T.: Semi-supervised learning and feature evaluation for rgb-d object recognition. Computer Vision and Image Understanding **139**, 149–160 (2015) ↑6, ↑11
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 248–255. IEEE (2009) ↑2, ↑3, ↑4, ↑6
13. Eitel, A., Springenberg, J.T., Spinello, L., Riedmiller, M., Burgard, W.: Multimodal deep learning for robust rgb-d object recognition. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. pp. 681–687. IEEE (2015) ↑5, ↑11
14. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. Journal of machine learning research **9**(Aug), 1871–1874 (2008) ↑7
15. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. IEEE transactions on pattern analysis and machine intelligence **35**(8), 1915–1929 (2013) ↑1
16. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014) ↑1
17. Gupta, S., Girshick, R., Arbeláez, P., Malik, J.: Learning rich features from rgb-d images for object detection and segmentation. In: European Conference on Computer Vision. pp. 345–360. Springer (2014) ↑5
18. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 447–456 (2015) ↑2, ↑6
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) ↑11
20. Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. vol. 1, p. 3 (2017) ↑11
21. Kim, J., Kwon Lee, J., Mu Lee, K.: Deeply-recursive convolutional network for image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1637–1645 (2016) ↑6
22. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012) ↑5
23. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on. pp. 1817–1824. IEEE (2011) ↑2, ↑6, ↑7, ↑11
24. Liu, H., Li, F., Xu, X., Sun, F.: Multi-modal local receptive field extreme learning machine for object recognition. Neurocomputing **277**, 4–11 (2018) ↑11

25. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. pp. 1717–1724. IEEE (2014) ↑5
26. Pollack, J.B.: Recursive distributed representations. Artificial Intelligence **46**(1-2), 77–105 (1990) ↑5, ↑6
27. Schwarz, M., Schulz, H., Behnke, S.: Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on. pp. 1329–1335. IEEE (2015) ↑1, ↑5, ↑11
28. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013) ↑1
29. Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 806–813 (2014) ↑1, ↑5, ↑8
30. Sharma, A., Tuzel, O., Liu, M.Y.: Recursive context propagation network for semantic scene labeling. In: Advances in Neural Information Processing Systems. pp. 2447–2455 (2014) ↑6
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) ↑6
32. Socher, R., Huval, B., Bath, B., Manning, C.D., Ng, A.Y.: Convolutional-recursive deep learning for 3d object classification. In: Advances in Neural Information Processing Systems. pp. 656–664 (2012) ↑5, ↑6, ↑7, ↑11
33. Socher, R., Lin, C.C., Manning, C., Ng, A.Y.: Parsing natural scenes and natural language with recursive neural networks. In: Proceedings of the 28th international conference on machine learning (ICML-11). pp. 129–136 (2011) ↑5, ↑6
34. Vedaldi, A., Lenc, K.: Matconvnet: Convolutional neural networks for matlab. In: Proceedings of the 23rd ACM international conference on Multimedia. pp. 689–692. ACM (2015) ↑7
35. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in neural information processing systems. pp. 3320–3328 (2014) ↑1, ↑2, ↑5
36. Zaki, H.F., Shafait, F., Mian, A.: Convolutional hypercube pyramid for accurate rgb-d object category and instance recognition. In: Robotics and Automation (ICRA), 2016 IEEE International Conference on. pp. 1685–1692. IEEE (2016) ↑2, ↑3, ↑6, ↑10, ↑11
37. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision. pp. 818–833. Springer (2014) ↑8
38. Zia, S., Yuksel, B., Yuret, D., Yemez, Y.: Rgb-d object recognition using deep convolutional neural networks. In: 2017 IEEE International Conference on Computer Vision Workshop (ICCVW). pp. 887–894. IEEE (2017) ↑6, ↑11