Proceedings of 2017 IEEE
International Conference on Mechatronics and Automation
August 6 - 9, Takamatsu, Japan

RGB-D Object Recognition based on RGBD-PCANet Learning

Shiying Sun[1,2], Xiaoguang Zhao[1], Ning An[1,2] and Min Tan[1]
1. State Key Laboratory of Management and Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences,
Beijing 100190, China
2. University of Chinese Academy of Sciences, Beijing 100049, China
Email: {sunshiying2013, xiaoguang.zhao, ning.an, min.tan}@ia.ac.cn

***Abstract*— In this paper, a simple deep learning method namely RGBD-PCANet is proposed for object recognition effectively. The proposed method extends the original PCANet for RGB-D images. Firstly, the RGB and depth images are preprocessed to meet the requirement of the network input layer. Secondly, features of RGB-D images are extracted by the two stages RGBD-PCANet which consists of cascaded PCA, binary hashing, and block-wise histograms. Finally, the SVM method is used as classifier. We evaluate the proposed method on the popular Washington RGB-D Object dataset. Extensive experiments demonstrate that the proposed RGBD-PCANet method achieves comparable performance to state-of-the-art CNN-based methods and the runtimes are low without GPU acceleration.**

***Index Terms*— RGB-D Object recognition, PCANet, deep learning**

I. Introduction

Object recognition is of essential importance in the fields of robotics, computer vision and multimedia. And it is the basic procedure for many tasks, for example, robot grabbing, object searching and environment understanding. Because of the large variety of categories and variable viewpoints, it is still a challenging task to recognize objects reliably. Traditional object recognition methods are mainly based on available RGB images. These methods usually use features extracted from RGB images including color, texture and local features [1][2][3]. Recently, deep learning techniques provide useful tools for rich feature representation. In particular, Convolutional Neural Networks (CNN) [4] has achieved excellent performance in image recognition tasks [5][6][7]. And CNN-based methods [8][9][10] improved the accuracy in several objects recognition datasets extremely. However, color information can be easily interfered by illumination, occlusion and so on. As the popularization of the cheap RGB-D sensors such as Kinect, one can easily acquire depth information of images which brings much help in object recognition. Similar to the domain of RGB object recognition, many CNN-based methods are proposed to extract features of RGB-D images [18][19][20], and they achieve excellent performances. But GPUs are always needed for training and recognizing with these methods which is an extra cost for robots and other mobile platform. Inspired by the simple structure and outstanding performance of PCANet method [22] in feature extraction, we propose a similar deep learning network namely RGBD-PCANet for RGB-D images which can be implemented in the platform without GPUs. Compared with CNN-based methods, experimental results show that the proposed method achieves comparable accuracy result but is more efficient.

The structure of the rest of this manuscript is as follows. Section II gives an introduction to related work. The details of the proposed method are presented in section III. In section IV, we present the experimental results in public dataset. Conclusions, with recommendations for future research, are given in Section V.

II. Related Work

In recent years, significant efforts have been made for object recognition with RGB-D images. The proposed methods can be roughly categorized into two groups: hand-crafted features and machine-learned features. For hand-crafted features, Rusu et al. [11], [12] proposed to use point feature histogram to extract the 3D structure feature of objects. Lai et al. [13] introduced a RGB-D object dataset and used a combination of several hand-crafted features including spin images, SIFT, HOG and color histograms. They made a recognition baseline by comparing the performance of several classifiers such as linear support vector machine, Gaussian kernel support vector machine and random forest. In [14], Bo et al. developed a set of kernel features on depth images that combined size feature, 3D shape and depth edges in a single framework, which improved the performance on RGB-D datasets. Browatzki et al. [15] used multiple descriptors such as 3D shape context and depth buffer for depth and SURF for color. However, hand-crafted features need the prior knowledge of objects and do not have a good performance on the large-scale objects dataset. For machine-learned methods, features are learned from raw data for RGB-D object recognition. Bo et al. [16] presented a hierarchical sparse coding learning method to extract features of multichannel images. In [17], Blum et al. proposed a convolutional k-means descriptor which can

978-1-5090-6759-6/17/$31.00 ©2017 IEEE

automatically learn feature responses in the neighborhood of detected interest points and is able to combine the color and depth into one, concise representation. Due to the great success on RGB image recognition, deep learning methods such as CNN are introduced to deal with RGB-D data and also obtain excellent performance. Socher et al. [18] proposed an integrated method with the combination of convolutional filters and recursive neural network (RNN). Features from color and depth channels are learned separately and then concatenated for the final soft-max classifier. Schwarz et al. [19] proposed to use two pre-trained CNNs to extract features from color and depth images respectively. In [20], Eitel et al. proposed a similar structure as [19]. However, they trained the fusion CNNs end-to-end, which achieved higher accuracy. Bai et al. [21] proposed to divide input images into several subsets according to their shapes and colors, and each subset is learned separately to extract features by RNNs. Despite the high accuracy achieved by CNN-based methods, they always need extra GPUs to accelerate the training and feature extraction process.

Recently, Chan et al. [22] proposed a simple deep model named PCANet in which PCA was employed to learn multistage filter banks. It can learn robust invariant features for various image recognition tasks. This method is easy to design and train on the platform without GPUs. Then PCANet method was applied to more image recognition tasks such as scene recognition [23], live fish recognition [24]. Many modified versions were proposed to improve the recognition performance including SPCANet [25], 2D-PCANet [26], Weighted-PCANet [27] and so on. In our method, we employ the similar structure with PCANet but modify the first layer to make full use of color and depth information.

III. Object Recognition Based on RGBD-PCANet

Benefit from the great success of PCANet method in face recognition domain, we propose a similar method to extract the features of RGB-D images. First, RGB-D image pairs are preprocessed to fit the requirement of feature extraction process. Then RGBD-PCANet method is used to extract the object feature. The final output is fed to an SVM classifier. We train the RGBD-PCANet using Washington RGB-D Object dataset [13] and then train the SVM classifier with the output of feature extraction.

A. Image Preprocessing

The image preprocessing process consists of two steps. The first step is encoding the raw depth image to the pseudo-color depth image. The second step is scaling the color image and the depth image to a constant size.

1) Encoding the Raw Depth Images: Pixels in raw depth images represent the distances between the corresponding points to the camera. To make full use of the surface depth information of the object, the raw depth image is encoded to the pseudo-color depth image. First, we normalize all depth values to lie between 0 and 255 which map the raw depth image to a gray image. Then the normalized depth images are transformed to three-channel RGB images to obtain more distinguishing depth information. A hierarchy mapping method is applied on the normalized depth images. For each pixel in depth image, the gray value is transformed to color value which encode the depth information by values of RGB. Mapping table is shown in Table I. Examples of normalized depth images and pseudo-color depth images are shown in Fig. 1.

TABLE I
MAPPING TABLE

Input gray level	Output color
$0 \sim 31$	Blue
$32 \sim 63$	Green
$64 \sim 95$	Pale blue
$96 \sim 127$	Purple
$128 \sim 159$	Red
$160 \sim 191$	Orange
$192 \sim 223$	Yellow
$224 \sim 255$	Canary yellow

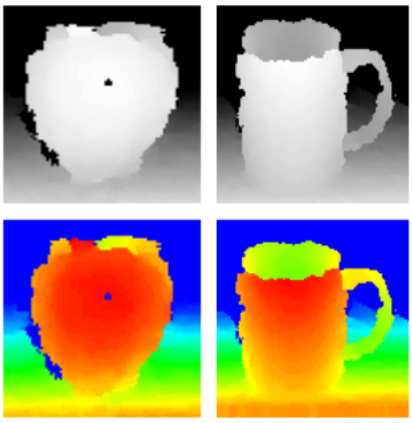

Fig. 1. Examples of normalized depth images (top) and pseudo-color depth images (bottom).

2) Scaling the Images: To fit the requirement of the feature extraction process, input color and depth images should be scaled to an appropriate size. The simplest method is to resize the cropped image by warping the image. However, the object may lost its inherent shape information by this procedure. So we employ the scaling process proposed in [20]. The original image is expanded to a square image by tiling the border of the longest side along the axis of the shorter side. Then the square image is scaled to the constant size. Fig. 2 shows the image scaling results of the two methods. The experimental results of different image sizes in Fig. 5 proved that the best size is 60×60.

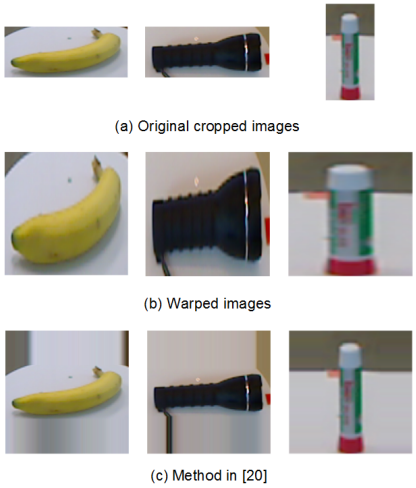

Fig. 2. Image scaling results of warping approach and method in [20].

B. Feature Extraction and Recognition

PCANet method is designed to extract features of color images. We follow the principle of PCANet and modify the data input method for RGB-D images. The proposed feature extraction method is named RGBD-PCANet and the block diagram is illustrated in Fig. 3. The details of the proposed method are described as following.

1) The First Convolution Layer: Suppose that there are N input RGB-D image pairs for training which is denoted by $\{I_i\}_{i=1}^{N}$. The depth and color image size is $m \times n$ and the patch size is $k_1 \times k_2$. For a given RGB-D image pair, we have a color image and a pseudo-color depth image. Then an RGB-D image pair is regarded as a 6-channel image. The 6 channels are r, g, b, dr, dg, db where dr, dg, db are the r, g, b channels of the pseudo-color depth image. For each channel, all patches in the i-th image are collected with the overlapping approach; i.e., $x_{i,1}, x_{i,2}, ..., x_{i,mn} \in \mathbb{R}^{k_1 k_2}$ where each $x_{i,j}$ denotes the j-th vectorized patch in I_i. Then we subtract the patch mean from each patch and get $\overline{X}_i = [\overline{x}_{i,1}, \overline{x}_{i,2}, ..., \overline{x}_{i,mn}]$, where $\overline{x}_{i,j}$ is a mean-removal patch. By constructing the same matrix for all input images and putting them together, we get:

$$X = [\overline{X}_1, \overline{X}_2, ..., \overline{X}_N] \in \mathbb{R}^{k_1 k_2 \times Nmn} \tag{1}$$

So for a given RGB-D image pair, we gather the same individual matrix for 6 channels of RGB-D image pairs, denoted by $X_r, X_g, X_b, X_{dr}, X_{dg}, X_{db} \in \mathbb{R}^{k_1 k_2 \times Nmn}$, respectively. Assuming that the number of filters in layer i is L_i, we use PCA to minimize the reconstruction error within a set of orthogonal filters:

$$\min_{V \in \mathbb{R}^{6k_1 k_2 \times L_1}} \left\| X - VV^T X \right\|_F^2, \ \ s.t. \ \ V^T V = I_{L_1} \tag{2}$$

where $X = [X_r^T, X_g^T, X_b^T, X_{dr}^T, X_{dg}^T, X_{db}^T]$ and I_{L_1} is an identity matrix of size $L_1 \times L_1$, V is a matrix consisting of a set of eigenvectors. Solve the optimization problem, and we get the L_1 principal eigenvectors of XX^T. Then the PCA filters of the first stage can be denoted as:

$$W_l^{r,g,b,dr,dg,db} = mat_{k_1,k_2,6}(q_l(XX^T)) \in \mathbb{R}^{k_1 \times k_2 \times 6}, \tag{3}$$

where $l = 1, 2, ..., L_1$, $q_l(XX^T)$ is the l-th principal eigenvector of XX^T, $mat_{k_1,k_2,6}(v)$ is a function that maps $v \in \mathbb{R}^{6k_1 k_2}$ to a matrix $W \in \mathbb{R}^{k_1 \times k_2 \times 6}$. Finally, the output of the first layer is:

$$I_i^l = I_i * W_l^1, i = 1, 2, ..., N \tag{4}$$

where I_i is the i-th input image, and W_l^1 is the l-th filter of the PCA filter bank in the first layer.

2) The Second Convolution Layer: This layer is similar to the second layer of PCANet. The output of the last layer is used as the input of this one. All the overlapping patches of ${I_i}^l$ are collected and the patch mean subtraction process is performed. The we get ${\overline{Y}_i}^l = [\overline{y}_{i,l,1}, \overline{y}_{i,l,2}, ..., \overline{y}_{i,l,mn}] \in \mathbb{R}^{k_1 k_2 \times mn}$, where $\overline{y}_{i,l,1}$ is the j-th mean-removed patch in ${I_i}^l$. Then $Y^l = [{\overline{Y}_1}^l, {\overline{Y}_2}^l, ..., {\overline{Y}_N}^l] \in \mathbb{R}^{k_1 k_2 \times Nmn}$ is defined as all the mean-removed patched of the l-th filter output. So all of the filter outputs can be expressed as:

$$Y = [Y^1, Y^2, ..., Y^{L_1}] \in \mathbb{R}^{k_1 k_2 \times L_1 Nmn} \tag{5}$$

Filters in second convolution layer are solved as:

$${W_\ell}^2 = mat_{k_1,k_2}(q_\ell(YY^T)) \in \mathbb{R}^{k_1 \times k_2}, \quad \ell = 1, 2, ..., L_2 \tag{6}$$

Finally, for each input ${I_i}^l$ of the second layer, we will have L_2 outputs, each convolves ${I_i}^l$ with ${W_\ell}^2$ for $\ell = 1, 2, ..., L_2$:

$$O_i^l = \{{I_i}^l * {W_\ell}^2\}_{\ell=1}^{L_2} \tag{7}$$

The number of outputs of the second layer is $L_1 L_2$.

3) The Output Layer: After the second layer, for each of the L_1 input ${I_i}^l$, there are L_2 real-valued output filters $\{{I_i}^l * {W_\ell}^2\}_{\ell=1}^{L_2}$. These outputs are binarized to get $\{H({I_i}^l * {W_\ell}^2)\}_{\ell=1}^{L_2}$. The value of $H(\cdot)$ is 1 if the argument is positive, and 0 otherwise. Then we view each corresponding pixel of the L_2 output as L_2 binary bits of a decimal number, denoted as:

$$T_i^l = \sum_{\ell=1}^{L_2} 2^{\ell-1} H({I_i}^l * {W_\ell}^2) \tag{8}$$

The above process convents the L_2 outputs in O_i^l back into a single integer-valued image, whose every pixel is an integer in the range $[0, 2^{L_2} - 1]$.

Now L_1 single integer-valued images are obtained, then each of them is partitioned into H blocks and the histogram of the decimal values in each block is computed. Finally, the H histograms are concatenated into one vector $Hist(T_i^l), l = 1, 2, ..., L_1$. After the encoding process above, the feature of the input image I_i is defined as:

$$f_i = [Hist(T_i^1), ..., Hist(T_i^{L_1})]^T \in \mathbb{R}^{(2^{L_2}) L_1 H} \tag{9}$$

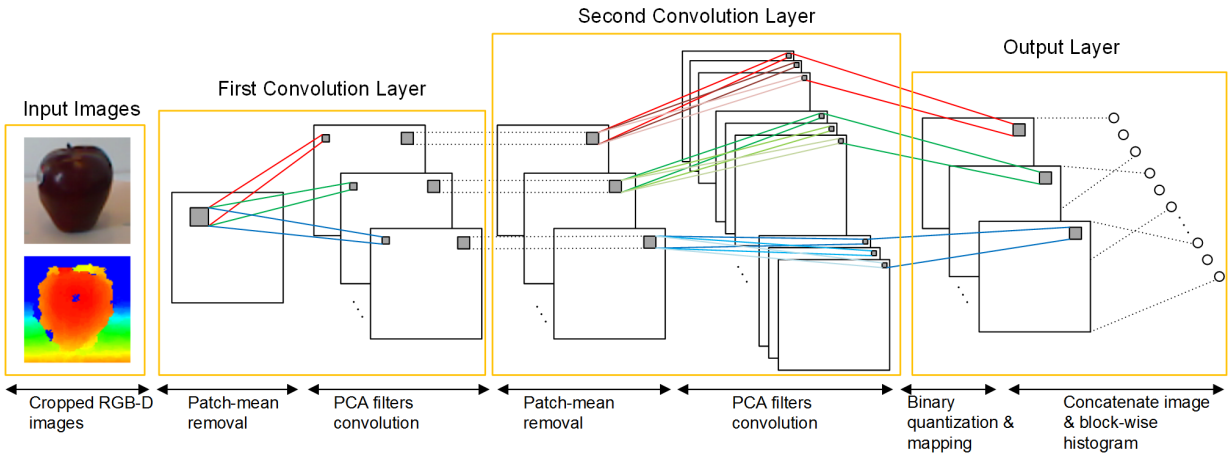

Fig. 3. The block diagram of RGBD-PCANet method.

The model parameters of RGBD-PCANet include the patch size k_1, k_2, the filters number L_1, L_2, the block size for histograms. In our experiments, the image size is set 60×60, the patch size is 5×5, $L_1 = 40$, $L_2 = 8$ and the block size is 7×7.

As the obtained object images consists of complex poses, we connect the Spatial Pyramid Pooling [28] process to the output layer and a pyramid of 4×4, 2×2, 1×1 is used. The dimension of each pooled feature is reduced to 2048 by PCA. So the feature dimension of each input RGB-D image pair is $2048 \times (4 \times 4 + 2 \times 2 + 1) = 43008$. This process can extract information invariant to large poses.

After feature extraction by RGBD-PCANet method, we use linear Support Vector Machines (SVMs) as the classifier to recognize categories of objects.

IV. Experimental Results And Analysis

RGB-D object recognition methods are usually evaluated on the challenging Washington RGB-D object dataset. There are 300 household objects with 51 categories in the dataset. Each object was captured with a Kinect style 3D camera witch was placed at three different heights and from multiple views.

Object recognition experiment in this paper was focused on category recognition. So the proposed RGBD-PCANet method was evaluated using the same ten cross-validation splits as in [13]. Each split consisted of roughly 35,000 training images and 7,000 images for testing. The task of the RGBD-PCANet was to predict the correct category of a new instance. Concerning the parameters of RGBD-PCANet, we set the image size 60×60, patch size 5×5, $L_1 = 40$, $L_2 = 8$ and the block size 7×7 to get the best result.

To show the effectiveness of the RGBD-PCANet method, we made a comparison of category recognition accuracies on the RGB-D object dataset with other methods. The result is shown in Table II. It can be seen that our method achieved very competitive results compared with state-of-the-art CNN-based method [20]. Note that our method has such a simple structure while the method in [20] need a fussy parameter fine-tune process and extra GPUs to accelerate the train and feature extraction procedures. Our method can achieve a comparable result under the condition that only common CPU are equipped, which verifies the effectiveness of the proposed method. The per-class recall is presented in Fig. 4 and almost two-thirds of the categories achieve a value greater than 95%. To find the best input image resolution, we conducted the experiment with different resolution input images. Results in Fig. 5 show that performance achieves the best when image size is 60×60. The main reason is that the resolutions of original images in the RGB-D dataset are mostly around 60×60. Besides, the higher input image size will cost more computation time in feature extraction process.

TABLE II
Comparison With Other Approaches Reported for the RGB-D Object Dataset

Method	Accuracy(%)
Nonlinear SVM [13]	83.9 ± 3.5
KDES [14]	86.2 ± 2.1
Upgraded HMP [16]	87.5 ± 2.9
CKM [17]	86.4 ± 2.3
CNN-RNN [18]	86.8 ± 3.3
CNN Features [19]	89.4 ± 1.3
Fus-CNN [20]	91.3 ± 1.4
ours	90.6 ± 2.3

Then we conducted experiments with four different P-CANet based baselines: 1) PCANet trained using RGB images only, named RGB PCANet; 2) PCANet trained using depth images only, named Depth PCANet; 3) PCANet with separate training for color and depth images, followed by concatenating the features of color and depth images, similar as structures in [19], named RGBD concatenated PCANet;

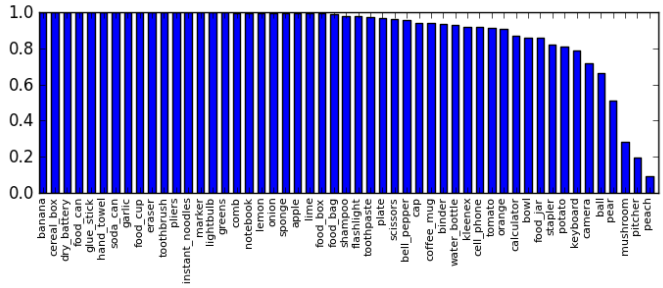

Fig. 4. Pre-class recall of our method on all test-splits.

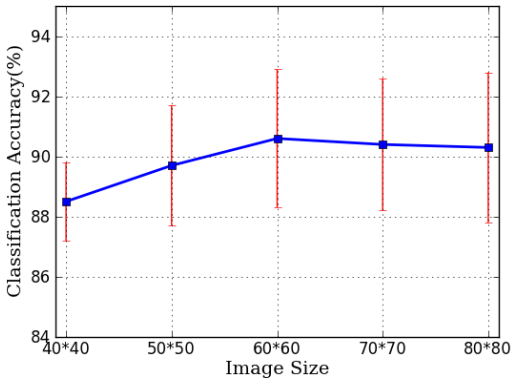

Fig. 5. Classification accuracy with different image sizes.

4) RGBD-PCANet trained using 4-channel inputs formed by RGB images and grayscale depth images, named RGBD-PCANet with 4-channel input. Linear SVM method was used as classifier when testing all the four baselines. Table III shows the comparison of recognition accuracy between our method and the four baselines. Results demonstrate that the depth information brings improvement of recognition accuracy indeed. And the performance of proposed structure was better than that of RGBD concatenated PCANet method. The comparison of our method with RGBD-PCANet with 4-channel input demonstrate that the pseudo-color depth image can provide more distinguishing information than grayscale depth image.

TABLE III
COMPARISON WITH DIFFERENT BASELINES ON RGB-D OBJECT DATASET

Method	Accuracy(%)
RGB PCANet	82.3 ± 3.4
Depth PCANet	75.6 ± 2.0
RGBD concatenated PCANet	88.4 ± 3.4
RGBD-PCANet with 4-channel input	88.9 ± 2.5
ours	90.6 ± 2.3

Computing power is usually very constrained in robotic applications. We tested the average runtime of feature extraction procedure on Washington RGB-D object dataset and on the notebook computer with a Core i5 CPU @ 2.5GHZ. Our method achieved 0.285s per input object which is low enough to allow frame rates of up to 3HZ. While the CNN-based method usually can achieve a high execution efficiency on the condition that GPUs are equipped. Schwarz et al. [19] presented the runtimes of the feature extraction procedure for a single object on a computer with an Intel Core i7 CPU 2.7GHz chipset and an NVidia GeForce GT 730M for acceleration. In their experiment platform, method in [16] cost 1.153s to process one frame, while CNN-based method in [19] cost 0.186s. As expected, the runtime of CNN-based methods could be low with the help of GPUs. However, without acceleration by GPUs, our method can still achieve a high efficiency which indicates that the proposed method is more practicable and suitable for mobile robots.

V. CONCLUSION

In this paper, we proposed an effective RGBD-PCANet method for object recognition with RGB-D images. The proposed method is composed of RGB-D images preprocessing, feature extraction and SVM classification. The proposed RGBD-PCANet is a simple deep learning method which extend the original PCANet method to jointly leverage the RGB and depth information. The proposed method was evaluated using the popular Washington RGB-D Object dataset. Compared with the latest CNN-based methods, the proposed method achieved comparable performance without the fussy parameter fine-tune process and extra GPUs for acceleration, which demonstrated the effectiveness. What's more, the runtime is low which means the proposed method is more practicable for application on robots and other mobile platforms. In the future, work will focus on a more efficient, robust approach to achieve higher precision for RGB-D objects recognition and more experiments on real scenes.

ACKNOWLEDGMENT

This work was Supported by the National Natural Science Foundation of China (61673378, 61421004).

REFERENCES

[1] M.J. Swain and D.H. Ballard, *Color indexing*, International journal of computer vision, 7(1), 1991, 11-32.
[2] D.G. Lowe, *Object recognition from local scale-invariant features*, Proc. 7th IEEE Conf. on Computer vision, 1999, 2, 1150-1157.
[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, *Speeded-up robust features (SURF)*, Computer vision and image understanding, 110(3), 2008, 346-359.
[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86(11), 1998, 227-2324.
[5] A. Krizhevsky, I. Sutskever, and G.E. Hinton, *Imagenet classification with deep convolutional neural networks*, Advances in neural information processing systems, 2012, 1097-1105.
[6] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, *Learning hierarchical features for scene labeling*, IEEE transactions on pattern analysis and machine intelligence, 35(8), 2013,1915-1929.

[7] M. Oquab1, L. Bottou, I. Laptev, and J. Sivic, *Learning and transferring mid-level image representations using convolutional neural networks*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, 1717-1724.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, 580-587.

[9] K. Simonyan, and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, arXiv preprint arXiv: 1409.1556, 2014.

[10] C. Szegedy, W. Liu, Y. Jia, and P. Sermanet et al., *Going deeper with convolutions*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, 1-9.

[11] R.B. Rusu, N. Blodow, M. Beetz, *Fast point feature histograms (FPFH) for 3D registration*, IEEE Conference on Robotics and Automation, 2009, 3212-3217.

[12] R.B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, *Fast 3D recognition and pose using the viewpoint feature histogram*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010, 2155-2162.

[13] K. Lai, L. Bo, X. Ren, and D. Fox, *A large-scale hierarchical multi-view rgb-d object dataset*, IEEE International Conference on Robotics and Automation (ICRA), 2011, 1817-1824.

[14] L. Bo, X. Ren, and D. Fox, *Depth kernel descriptors for object recognition*, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011, 821-826.

[15] B. Browatzki, J. Fischer, B. Graf, and H.H. Blthoff et al., *Going into depth: Evaluating 2D and 3D cues for object classification on a new, large-scale object dataset*, IEEE International Conference on Computer Vision Workshops, 2011, 1189-1195.

[16] L. Bo, X. Ren, and D. Fox, *Unsupervised feature learning for RGB-D based object recognition*, Experimental Robotics, 2013, 387-402.

[17] M. Blum, J.T. Springenberg, J. Wulfing, and M. Riedmiller, *A learned feature descriptor for object recognition in RGB-D data*, IEEE International Conference on Robotics and Automation, 2012, 1298-1303.

[18] R. Socher, B. Huval, B. Bhat, and C.D. Manning et al., *Convolutional-recursive deep learning for 3D object classification*, Advances in Neural Information Processing Systems, 2012, 665-673.

[19] M. Schwarz, H. Schulz, and S. Behnke, *RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features*, International Conference on Robotics and Automation, 2015, 1329-1335.

[20] A. Eitel, J.T. Springenberg, L. Spinello, and M. Riedmiller et al., *Multimodal deep learning for robust RGB-D object recognition*, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2015, 681-687.

[21] J. Bai, Y. Wu, J. Zhang, and F. Chen, *Subset based deep learning for RGB-D object recognition*, Neurocomputing, 165, 2015, 280-292.

[22] T.H. Chan, K. Jia, S. Gao, and J. Lu, et al., *PCANet: A simple deep learning baseline for image classification?*, IEEE Transactions on Image Processing, 24(12), 2015, 5017-5032.

[23] C. Chen, D.H. Wang, and H. Wang, *Scene character recognition using PCANet*, Proc. 7th IEEE Conf. on Internet Multimedia Computing and Service, 2015.

[24] H. Qin, X. Li, J. Liang, and Y. Peng, et al., *DeepFish: Accurate underwater live fish recognition with a deep architecture*, Neurocomputing, 187, 2016, 49-58.

[25] L. Tian, C. Fan, Y. Ming, and Y. Jin, *Stacked PCA Network (SPCANet): An effective deep learning for face recognition*, IEEE International Conference on Digital Signal Processing, 2015, 1039-1043.

[26] Z. Jia, B. Han, and X. Gao, *2DPCANet: Dayside Aurora Classification Based on Deep Learning*, CCF Chinese Conference on Computer Vision, 2015, 323-334.

[27] C. Yuan, and J. Huang, *Weighted-PCANet for Face Recognition*, International Conference on Neural Information Processing, 2015, 246-254.

[28] K. He, X. Zhang, S. Ren, and J. Sun, *Spatial pyramid pooling in deep convolutional networks for visual recognition*, European Conference on Computer Vision, 2014, 346-361.