SCENE SEMANTIC SEGMENTATION FROM INDOOR RGB-D IMAGES USING ENCODE-DECODER FULLY CONVOLUTIONAL NETWORKS

Zhen Wang *, Te Li, Lijun Pan, Zhizhong Kang

China University of Geosciences, Beijing - (comige@gmail.com, telizy@126.com, panjijun0819@163.com, zzkang@cugb.edu.cn)

Commission IV, WG IV/5

KEY WORDS: Indoor Scene, Semantic Segmentation, RGB-D Images, Encode-Decoder process, Fully Convolutional Networks, Multiple Kernel Maximum Mean Discrepancy (MK-MMD), Full Connect CRFs

ABSTRACT:

With increasing attention for the indoor environment and the development of low-cost RGB-D sensors, indoor RGB-D images are easily acquired. However, scene semantic segmentation is still an open area, which restricts indoor applications. The depth information can help to distinguish the regions which are difficult to be segmented out from the RGB images with similar color or texture in the indoor scenes. How to utilize the depth information is the key problem of semantic segmentation for RGB-D images. In this paper, we propose an Encode-Decoder Fully Convolutional Networks for RGB-D image classification. We use Multiple Kernel Maximum Mean Discrepancy (MK-MMD) as a distance measure to find common and special features of RGB and D images in the network to enhance performance of classification automatically. To explore better methods of applying MMD, we designed two strategies; the first calculates MMD for each feature map, and the other calculates MMD for whole batch features. Based on the result of classification, we use the full connect CRFs for the semantic segmentation. The experimental results show that our method can achieve a good performance on indoor RGB-D image semantic segmentation.

1. INTRODUCTION

Due to the increasing attention for indoor environments and the development of the low-cost RGB-D sensors such as the Kinect, the RGB-D images can be used as data input for more and more indoor applications such as indoor mapping, modelling and mobility. The automatic semantic segmentation for indoor RGB-D images is the basis on the scenes understanding to further serve these applications. Especially for the indoor scenes, the depth information is very important. Many objects have similar color or texture, which are difficult to be distinguished by only RGB images (Tao, 2013).

The semantic segmentation has been studied for a long time in the fields of remote sensing (Qin, 2010, Kampffmeyer, 2016, Lin, 2016, Marmanis, 2016) or compute vision (Arbel áez, 2012, Couprie, 2012, Long, 2015, Noh, 2015). As semantic segmentation divides images into some non-overlapped meaningful regions, one or more of the three main methods – conditional random fields (CRFs) methods (Hu, 2016), segmentation combining with merging methods (Forestier, 2012), and the deep learning methods (Chen, 2016), are used. The CRFs methods can effectively use the pairwise information, which helps the edges of the objects to be clear segmented. The segmentation combining with the merging methods always uses knowledge to merge an over segmented image into the meaningful regions. With the great development of the deep learning, the deep learning methods can classify the images with high precision, which can serve as pre-processing for the two methods above. Moreover, parts of the two methods above can be presented by the deep learning network, for instance the work which shows the CRFs can be approximate as the recurrent neural networks (Zheng, 2015).

However, because of the specific characteristics of the indoor RGB-D images, the semantic segmentation methods of RGB or remote sensing images cannot be directly used. The D images show the depth information (but not spectral), so the pixel values do not indicate the variances in the different classes. Directly using the RGB-D images as four channel images cannot make good use of feature information between RGB images and D images. Therefore, the key to semantic segmentation for RGB-D images is how to effectively utilize the D information to conduct the RGB information to process semantic segmentation.

The semantic segmentation methods for RGB-D images can also be sorted into methods with or without deep learning. The methods without deep learning use the depth information explicitly. Koppula (2011) proposed a graphical model that captures various features and contextual relations, including local visual appearance and shape cues, object co-occurrence relationships and geometric relationships. Tang (2012) designed a histogram of oriented normal vectors (HONV) to capture local geometric characteristics for RGB-D images. Silberman (2012) segmented the indoor scenes by the support inference from RGB-D images. Gupta (2013) proposed an algorithm for object boundary detection and hierarchical segmentation. Gupta (2014) proposed a new geocentric embedding for D images and demonstrated that this geocentric embedding worked better than using the raw D images for learning feature representations with convolutional neural networks. Huang (2014) converted the RGB-D images to a 3D point clouds with color to segment the RGB-D images.

Compared to the methods without deep learning, the methods with deep learning use the depth information more implicitly by a variety of network architectures. Ling Shao (2017) analyzed four prevalent basic deep learning models (i.e., deep belief networks (DBNs), stacked de-noising auto-encoders (SDAE), convolutional neural networks (CNNs) and long short- term memory (LSTM) neural networks) for the RGB-D dataset and showed that CNNs obtained the best results. Richard Socher (2012) introduced a model based on a combination of CNN and RNN for 3D object classification. Zaki (2017) proposed a deeply supervised multi-modal bilinear CNN for semantic

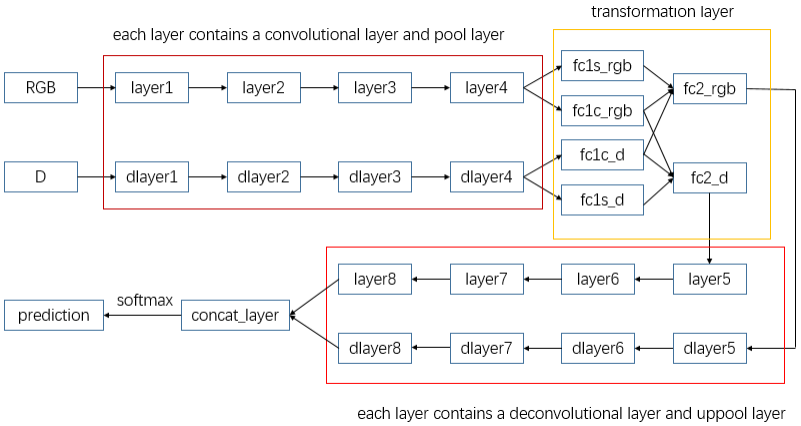

Figure 1. Architecture of the network

segmentation. Couprie (2013) first used a multiscale network for the RGB images while cutting the D images into super-pixels, and then aggregated the classifier predictions in the super-pixels to obtain the labels for the super-pixels. Wang (2016) proposed a feature transformation network to bridge the convolutional networks and de-convolutional networks and found the common and special features between RGB and D images automatically. Our motivation comes from this study hence the similar use of its architectures. But for the feature transformation network, we took a different approach to find the common and special features.

This paper proposes a deep network and the use of full connect CRFs for semantic segmentation. The main contribution of this paper is the proposition of a loss function which can find the common and special features of RGB and D images to enhance performance of classification

This paper proposes a deep network and the use of full connect CRFs for semantic segmentation. The main contribution of this paper is the proposition of a loss function which can find the common and special features of RGB and D images to enhance performance of classification

2. MAIN BODY

2.1 Deep Learning Architectures

The deep learning architectures are based on SegNet (Badrinarayanan, 2015) combining with the Multiple Kernel Maximum Mean Discrepancy (MK-MMD). The architectures are shown in Figure 1. Before feeding data into the network, each channel of RGB-D images is normalized by the means and variances of the channel. Then, the RGB images as a three channel input and D images as a single channel are fed into the network separately. This way, highlighting pseudo depth edges due to RGB edges or vice-versa can be reduced. In the network, a symmetric encoder-decoder process is used, which contains four convolutional and pooling layers for RGB, four convolutional and pooling layers for D, a transformation layer, four corresponding de- convolutional and un-pooling layers for RGB, four corresponding de-convolutional and un-pooling layers for D, and the softmax layer. The encoder-decoder process can effectively catch the global and the local features of the images as shown in SegNet. The transformation layer is used to find the similarities between the RGB and D images to help improve the performance of semantic segmentation. The details are in the next section. The softmax layer is used to output the prediction probability of the network. The size of the convolutional kernel in the convolutional and deconvolutional layers is 7×7×64. The non-overlapping max pooling with a 2×2 window is used. The activation function is ReLu for the convolutional and deconvolutional layers. The Batch Normalization (Ioffe and Szegedy, 2015) is used before the activation.

2.2 The Transformation Layer

Although the SegNet can also classify the RGB-D images as the architectures in Figure 1 without the transformation layer, the network cannot effectively utilize the information derived from RGB and D images respectively because of the over-fitting, therefore the loss function is needed for regularization.

As can be seen in RGB-D images, RGB and D images have the same labels, but the obvious differences are the color and texture. Therefore, we try to find the similarities which may be the same edges or other things to help the network for semantic segmentation. This procedure is followed by the last pooling layer, because after the convolution and pooling, the influence of the color and texture is reduced. Besides, the last pooling layer has the biggest receptive field in the network and it can maintain more global information.

By using the same architectures (Wang, 2016); the fc1c_rgb and fc1s_rgb are generated by layer4, and the fc1c_d and fc1s_d are generated by dlayer4. The differences between the fc1c_rgb and fc1c_d are then minimized and the difference between fc1s_rgb and fc1s_d maximized. This way, both the common and special parts of the RGB and the corresponding D images are automatically extracted in the network. The loss function of the whole network is shown as Eq.1:

$$L = l_s(label) + l_d(fc1c\_rgb, fc1c\_d) - l_d(fc1s\_rgb, fc1s\_d) \quad (1)$$

where l_s is the softmax cross entropy, l_d is a measure of distance, which will be introduced in the next section.

To further enhance the common information, the fc2_rgb and fc2_d which are used for de-convolutional and un-pooling take double the common information. The fc2_rgb are obtained by the sum of the two commons and the fc1s_rgb and fc2_d are obtained by the sum of the two commons and the fc1s_d.

2.3 MK-MMD

The difference between the fc1c_rgb and fc1c_d or fc1s_rgb and fc1s_d should be measured. We do not strictly keep the fc1c_rgb and fc1c_d the same, as it may reduce the capacity of the network. Therefore, the l_2 distance and the cross entropy distance are not used. The MK-MMD which describes the differences between two distributions is used here, which can find similarity but not exactly the same things.

MMD is a kernel-based modern approach that addresses the problem of comparing the data samples from two probability distributions (Karsten, 2006). If x has distribution P and y has distribution Q, respectively, the MMD can be written as Eq.2:

$$MMD^2(\mathbf{F},\mathbf{P},\mathbf{Q}) := \sup_{f\in F}(E_P[f(\mathbf{x})] - E_Q[f(\mathbf{y})]) \quad (2)$$

where E is the expectation function. F is a function set.

If the F is a unit ball in reproducing kernel Hilbert space (RKHS), the MMD (**F**,**P**,**Q**)=0, if and only if **P**=**Q** (Gretton, 2012). Based on the condition, an unbiased estimator of MMD by shown in Eq. 3:

$$MMD^2(\mathbf{F},\mathbf{X},\mathbf{Y}) = \frac{1}{n(n-1)}\sum_{i\neq j}^{n} k(\mathbf{x}_i,\mathbf{x}_j) - \frac{2}{nm}\sum_{i,j=1}^{n,m} k(\mathbf{x}_i,\mathbf{y}_j) - \frac{1}{m(m-1)}\sum_{i\neq j}^{m} k(\mathbf{y}_i,\mathbf{y}_j) \quad (3)$$

where k(·, ·) is a Gaussian kernel

However, only one kernel is not flexible enough and cannot adequately describe a variety of distributions. Therefore, the single kernel in Eq.3 is replaced by the multiple kernels as shown in Eq.4 forming the MK-MMD and now the kernel can be seen as the positive linear combination of kernels:

$$\mathbf{K} := \{k = \sum_{u=1}^{m}\beta_u k_u \mid \sum_{u=1}^{d}\beta_u = 1, \beta_u \geq 0, \forall u\} \quad (4)$$

where, k_u is a Gaussian kernel

Specifically, in measuring the distances, we tried two different ways. One is to find the distances between all the feature maps of the RGB and D image in RGB-D images of a batch and the other is to find the distances between feature maps of the RGB and D image in one RGB-D image. They are shown as follows:

(1) As the data is all obtained in the classroom, all the images may obey the same distribution. The l_d(X,Y) is shown as Eq. 5:

$$l_d(X,Y) = MMD^2(\mathbf{F},\mathbf{X},\mathbf{Y}) \quad (5)$$

For finding the common parts, the **X** represents all the feature maps of RGB images in the fc1c_rgb in a batch and **Y** represents all the feature maps of D images in the fc1c_d in the batch. For finding the special parts, the **X** is all feature maps of RGB images in the fc1s_rgb in a batch and **Y** is all feature maps of D images in the fc1s_d in the batch. For specially, as an example, $\mathbf{X}_i$ is a matrix. The size of row equals to the batch size and the size of column is the number of feature maps multiply the pixel number of feature maps.

(2) Calculate the MMD between the feature maps. The l_d(X,Y) is shown as Eq. 6:

$$l_d(X,Y) = \sum_{i=0}^{m} MMD^2(\mathbf{F},\mathbf{X}_i,\mathbf{Y}_i) \quad (6)$$

where *m* is the number of feature maps.

For finding the common parts between the feature maps, when one RGB-D image input, the $\mathbf{X}_i$ is the *i*th feature map in the fc1c_rgb and $\mathbf{Y}_i$ is also the *i*th feature map in the fc1c_d. For finding the special parts, the $\mathbf{X}_i$ is the *i*th feature map in the fc1s_rgb and $\mathbf{Y}_i$ is the *i*th feature map in the fc1s_d. For specially, as an example, $\mathbf{X}_i$ is a matrix and the sizes of row and column are the same as those of a feature map.

2.4 Fully Connected CRFs

Because the results of the network always look chaotic and the boundaries of different classes are blended, the CRFs are used to deal with this problem. However, the traditional CRFs which only use the information in the short range are not suitable for the score maps produced by the deep convolutional neural networks (Chen, 2015). The Fully Connected CRFs (Krähenbühl, 2012) which can use the information in the long range are used here. The model employs the energy function as shown in Eq.7-10:

$$E(x) = \sum_i \psi_i(x_i) + \sum_{i,j}\psi_{ij}(x_i, x_j) \quad (7)$$

$$\psi_i(x_i) = -\log P(x_i) \quad (8)$$

$$\psi_{ij}(x_i, x_j) = \mu(x_i, x_j)\sum_{m=1}^{K} w^{(m)}k^{(m)}(f_i, f_j) \quad (9)$$

$$k(f_i, f_j) = w^{(1)}\exp(-|p_i - p_j|^2/2\theta_\alpha^2 - |I_i - I_j|^2/2\theta_\beta^2) + w^{(2)}\exp(-|p_i - p_j|^2/2\theta_\gamma^2) \quad (10)$$

where x_i and x_j are the labels of pixel *i* and pixel *j*. The $\psi_i(x_i)$ is the unary potential calculated by Eq.4, which describes the probability of a label assignment to a pixel. The $p(x_i)$ is the probability for pixel i labeled x_i, which can be outputted by the network. The $\psi_p(x_i, x_j)$ is the pairwise potential calculated by Eq.5, which describes the relationship between the two pixels. The $\mu(x_i, x_j)$ is the potts model, i.e. $\mu(x_i, x_j)$=1 , when $x_i \neq x_j$, otherwise $\mu(x_i, x_j)$=0 . $k^{(m)}(f_i, f_j)$ is the *m*th Gaussian kernel. $w^{(m)}$ is the *m*th linear combination weights for *m*th Gaussian kernel. As shown in Eq.6, the $k^{(m)}(f_i, f_j)$ contains two parts. The former is the appearance kernel which controls the nearby pixels with similar color likely to be in the same class. The latter is the smoothness kernel which removes small isolated regions. The θ_α, θ_β, θ_γ are the parameters of the Gaussian kernel.

The fully connected CRFs can be an efficient approximate probabilistic inference (Krähenbühl, 2012), which can deal with an image in a short time.

When all the probabilities for pixels are obtained by the network, these probabilities are fed into the fully connected CRFs. After the inference of the fully connected CRFs is finished, the probabilities for each pixel with all labels are obtained and the label with the max probability is set as the label of the pixel.

3. EXPERIMENTS

In this section, to evaluate the performance of our method, it is applied to the real data acquired by the Microsoft Kinect depth camera in the laboratory room scenes which contain a total of four classrooms. The size of a RGB-D image is 960x540. In the RGB images, the fan, the table and the walls are white, and the display and the stool are black. The only color information of the RGB image is difficult to distinguish. Therefore the depth information is used to help us for semantic segmentation. Examples of the obtained RGB and D images are shown in Figures 2 (i)-(j) and Figures 3 (i)-(j). However, because the range of Kinect depth camera is only 1 to 3 meters, there is a large number of missing data which is the objects out of the range in D images, as shown in the sides of the Figures 2-3 (j). Also these are no depth information on the black surface because the infrared is absorbed by black objects. As shown in the red boxes in Figure 3 (j), the things in the boxes are parts of seats, tables and displayers, which are black in the red boxes in Figure 3 (j). Moreover, much grid-like missing data is in D image everywhere. All of the missing data will have a certain impact on semantic segmentation results.

Based on objects' essential attributes, we classify the RGB-D images from the scenes into 11 classes by handcraft as the ground truth. There are walls, floors, ceilings, displayers, seats, tables, curtains (and windows), fans, hangings, lights and doors. Table 1 shows the proportion of each object in overall samples for training and testing.

	Train		Test	
	number	proportion	number	proportion
Wall	5160840	26.13%	1481909	24.83%
Floor	2593811	13.13%	658858	11.04%
Ceiling	3893564	19.72%	1342520	22.50%
Displayer	1483842	7.51%	459501	7.70%
Seat	1423088	7.21%	278329	4.66%
Table	3212701	16.27%	1091727	18.29%
Curtain	1405532	7.12%	480981	8.06%
Fan	139283	0.71%	64927	1.09%
Hanging	88517	0.45%	27091	0.45%
Light	189239	0.96%	66432	1.11%
Door	156922	0.79%	15309	0.26%
Total	19747339	100%	5967584	100%

Table 1 numbers and proportion of each object in overall samples for training and testing

For classify the RGB-D images, as we adopt two different methods to calculate MMD in the network, one is to measure similarity of the whole batch and another one is to measure similarity of each feature map. For the purpose of simplicity, the first one is named RGBD+MMD1 and the second one is named RGBD+MMD2. We also compare our methods to some baselines. One uses only RGB images as input and the SegNet directly named RGB. The other named RGBD uses the architectures shown in Figure 1, but do not contain the transformation layer. That is, in the architectures, the layer4 is connected to the layer5 and the dlayer4 is connected to the dlayer5. The CRFs are implemented for all four methods. Table 2 outlines the performance of semantic segmentation by all eight methods based on precision/recall and mean IOU. Figure 2 and Figure 3 show two semantic segmentation results for all eight methods. Table 3 is the semantic segmentation charts' legend. The black areas are all the things which are not in all 11 categories, so these parts are not included in the training process and semantic segmentation results calculation. By the way, the IOU is calculated by the Eq.11 and the mean IOU is the mean of the IOU of the 11 classes.

$$IOU = \frac{\mathbf{A} \cap \mathbf{B}}{\mathbf{A} \cup \mathbf{B}} \tag{11}$$

where **A** is the predict label, **B** is ground truth.

3.1 The Performance of RGBD + MMD

According to Table 2 for two proposed MMD methods(RGBD + MMD1 and RGBD + MMD2), among the 11 categories, the classification performance of walls, ceilings, curtains (windows) and lights are the best, with precision and recall rates all over 85%, followed by floor, displayers, tables and their appendages. It can be easily found in Figures 2 and 3 that the results for vision fit to the performance of Table 2, which shows our methods can achieve high classification performance. The classification performance of fans, hangers and doors are relatively poor. In detail, the fans and hangings' recall rates are low, which means fans and hangings were partially misinterpreted into other categories. This is basically because of their limited training and testing samples and data missing in depth images, especially when objects are out of Kinect camera's sensing range. As we can see in the Figure 2, in the blue boxes the fans are partly or almost missing in the D image, which causes the two fans are not recognized well. On the contrast, doors have a high recall rate with low semantic segmentation precision. As is shown in Figure 2, the door was identified successfully, but its low semantic segmentation precision suggests that there are some other types of targets that are misinterpreted into doors. This is mainly because some shadow areas the color of which is dark and similar to the color of door are classified to doors.

Comparing the results of two different MMD methods, we find RGBD+MMD2 method is better, as its mean IOU value is higher than MMD1 by 0.9%. It is maybe because the constraint in RGBD+MMD2 method is more specific compared to the RGBD+MMD1 method in which the feature maps is not a one-to-one correspondence. As is shown at the left top side of the images in Figures 2 (a)-(h), because of the missing data of the D image, in the results of the RGBD+MMD1 and the RGBD, these regions are classified wrong. The same condition can be found at the right top side and the blue boxes in the Figure 3. However, these regions in the results of RGBD+MMD2 are classified well, which means the RGBD+MMD2 is robust for the missing data.

Compared to the results obtained by RGBD, methods that adopt the MK-MMD are better. The results show that the mean IOU value of RGBD + MMD1 and RGBD + MMD2 increased 6.7% and 7.1% relative to RGBD. Also we can see the right top side of the images in Figure 2 (a)-(h), the results of RGBD are the most affected by the missing data of D image. All of these demonstrate that the MMD constraints can improve the neural

network's capability to strengthen objects' boundary and enhance the semantic segmentation performance.

Based on the above table, we can also infer that using RGB-D images for classification is better than using only RGB images. This is because the D images contain rich distance information which could help networks to enhance objects' edges, and to some extent, D images also provide some spatial dependency which may be helpful for our models to identify targets in question. Although the RGB images do not suffer from the missing data of D images, as show in Figures 2 and 3, at the no missing data areas, the classification performance of the methods based RGB-D images are all better than that of the method only used the RGB images.

	RGB	RGB+CRF	RGBD	RGBD+CRF	RGBD+ MMD1	RGBD+ MMD1+CRF	RGB+ MMD 2	RGBD+ MMD2+CRF
Mean IOU	0.601	0.618	0.656	0.664	0.689	0.731	0.698	0.735
Wall	0.857/ 0.820	0.874/ 0.844	0.870/ 0.874	0.883/ 0.883	0.892/ 0.932	0.904/ 0.945	0.916/ 0.928	0.928/ 0.941
Floor	0.693/ 0.794	0.715/ 0.835	0.781/ 0.839	0.785/ 0.850	0.793/ 0.822	0.807/ 0.846	0.774/ 0.829	0.800/ 0.874
Ceiling	0.857/ 0.960	0.849/ 0.975	0.886/ 0.945	0.872/ 0.96	0.938/ 0.971	0.936/ 0.981	0.935/ 0.970	0.934/ 0.987
Displayer	0.718/ 0.794	0.746/ 0.831	0.743/ 0.817	0.763/ 0.854	0.734/ 0.828	0.752/ 0.876	0.725/ 0.835	0.753/ 0.890
Seat	0.559/ 0.603	0.665/ 0.636	0.635/ 0.660	0.734/ 0.659	0.636/ 0.632	0.727/ 0.662	0.692/ 0.646	0.791/ 0.694
Table	0.800/ 0.639	0.811/ 0.663	0.865/ 0.732	0.874/ 0.770	0.849/ 0.731	0.87/ 0.756	0.867/ 0.772	0.887/ 0.768
Curtain	0.946/ 0.888	0.977/ 0.912	0.940/ 0.906	0.955/ 0.912	0.960/ 0.898	0.981/ 0.920	0.951/ 0.928	0.969/ 0.928
Fan	0.799/ 0.507	0.908/ 0.234	0.776/ 0.580	0.906/ 0.255	0.799/ 0.637	0.939/ 0.481	0.798/ 0.674	0.935/ 0.484
Hanging	0.617/ 0.518	0.827/ 0.558	0.710/ 0.493	0.932/ 0.520	0.802/ 0.685	0.913/ 0.680	0.836/ 0.653	0.956/ 0.725
Light	0.906/ 0.835	0.937/ 0.731	0.766/ 0.879	0.826/ 0.770	0.887/ 0.889	0.957/ 0.878	0.878/ 0.896	0.943/ 0.839
Door	0.544/ 0.724	0.661/ 0.806	0.866/ 0.772	0.945/ 0.836	0.637/ 0.917	0.827/ 0.983	0.695/ 0.946	0.753/ 0.928

Table 2. Performance of semantic segmentation by eight methods (precision/recall and mean IOU)

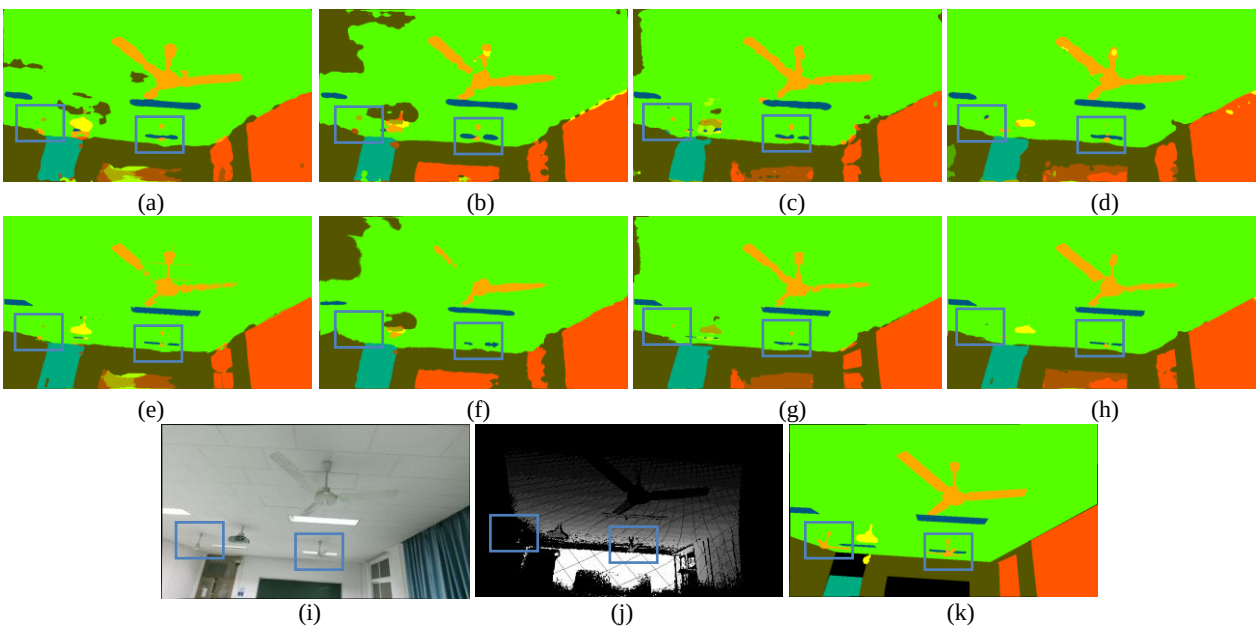

Figure 2. One example of Semantic segmentation results of eight methods. (a) RGB, (b) RGBD, (c) RGBD+MMD1, (d) RGBD+MMD2, (e) RGB+CRF, (f) RGBD+CRF, (g) RGBD+MMD1+CRF, (h) RGBD+MMD2+CRF, (i) RGB image, (j) Depth image, (k) Ground truth

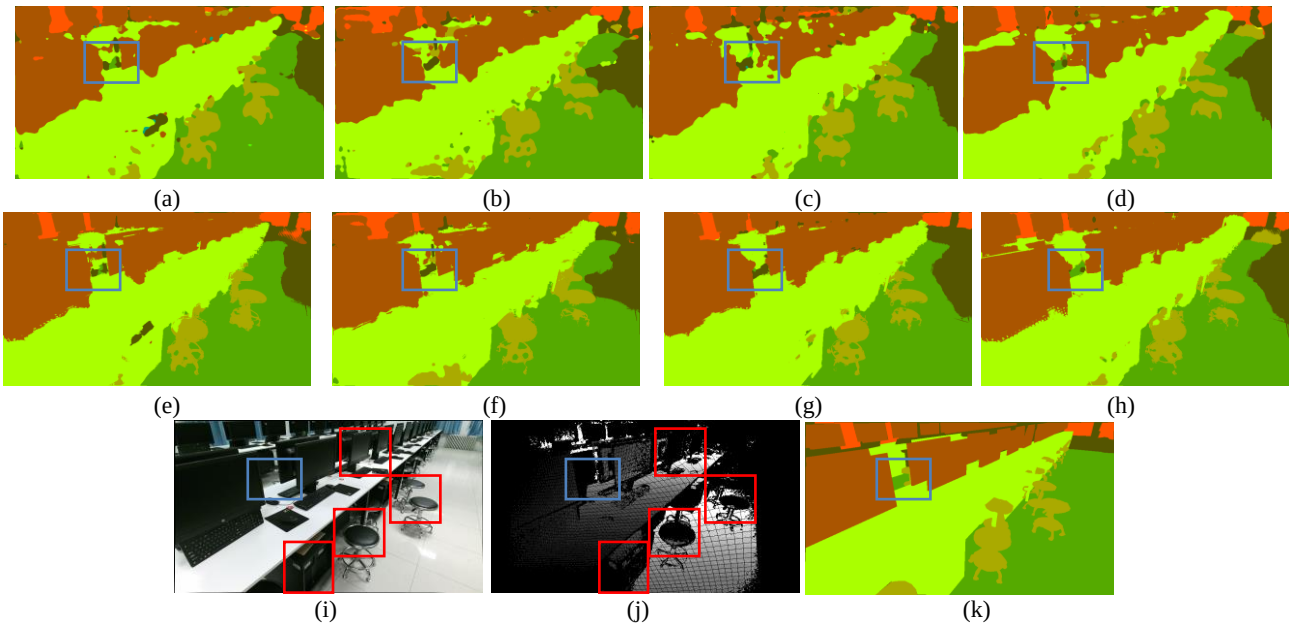

Figure 3. Another example of Semantic segmentation results of eight methods. (a) RGB, (b) RGBD, (c) RGBD+MMD1, (d) RGBD+MMD2, (e) RGB+CRF, (f) RGBD+CRF, (g) RGBD+MMD1+CRF, (h) RGBD+MMD2+CRF, (i) RGB image, (j) Depth image, (k) Ground truth

#	Color	Class
0		Ignored
1		Wall
2		Floor
3		Ceiling
4		Displayer
5		Seat
6		Table
7		Curtain
8		Fan
9		Hanging
10		Light
11		Door

Table 3. Semantic segmentation Charts' legend

3.2 The Performance of Full Connected CRFs

In Table 2, it is clear that the Mean IOU value of the four methods are improved by 1.7%, 0.8%, 4.2%, 3.7% respectively after the full connected CRFs processing. It can be seen that the CRFs play a very effect role in the semantic segmentation of the images. The CRFs could re-correct the false semantic segmentation result in the network according to the spatial relationship, and improve semantic segmentation precision. As shown in Figure 2 and Figure 3, after CRFs processing, 'pepper noises' are basically removed and we get a sharp boundary which fit the real object boundary well. In general, all the classes are semanticly separated.

However, the CRFs also lead to the semantic segmentation precision of some small objects (fans, suspended objects, etc.) reduction. This phenomenon implies that the CRFs which are based on spatial relationships and distribution probability may be relatively weak to discriminate some small objects in large scenes. And it's not hard to find out that the recall rate of fan in the scene is generally reduced after the CRFs. The reason is the CRFs in our paper only use the RGB images as reference data.

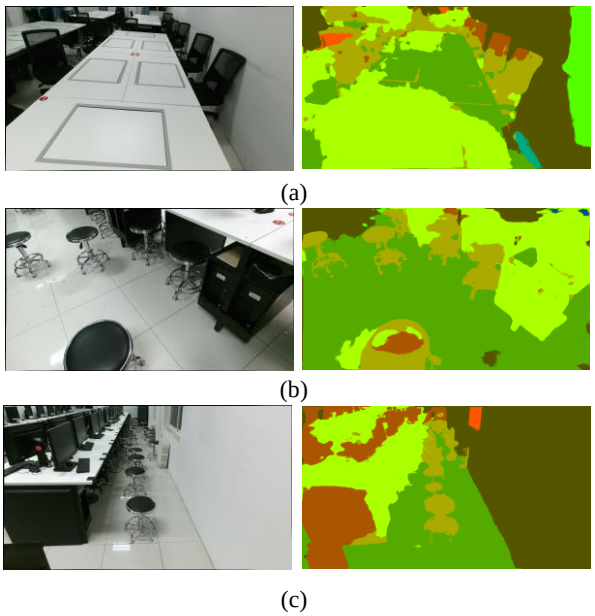

Figure 4. Unsatisfactory semantic segmentation results for tables, seats by RGBD+MMD2+CRF. (a) Unsatisfactory semantic segmentation results for tables, (b) Unsatisfactory semantic segmentation results for seats, (c) Unsatisfactory semantic segmentation results for displayers. RBG images are shown at left colomn and are the semantic segmentation results by RGBD+MMD2+CRF are are shown at right colomn.

In RGB images, the color of the fans is similar to that of the ceiling, which causes the edges of fans not to be clear enough, as is shown in Figure 2 and 3, after the CRFs, some parts of fans are recognized as ceiling by the models. CRFs which do not refer to the depth information become powerless when targets' edges are obscure in RGB images. However, there are a large number of missing data in D image, which keep the D images out of the CRFs.

3.3 Future Works

In the experiment, we find that semantic segmentation performance for tables, displayers and seats was not entirely satisfactory. Figure 4 shows these unsatisfactory semantic segmentation results by RGBD+MMD2+CRF. From the Figure 4 (a) it can be discovered that parts of the table are recognized as floor because they are all white. In Figure 4 (b), parts of the seats are recognized as displayers. Also it can be discovered from Figure 4 (c) that parts of tables are recognized as displayers. For the Figures 4 (b) and (c), it is because the confused parts are all black and these are no depth information. For these objects, it's hard to discriminate them if the surrounded objects are not considered. Obviously, our model's ability for space-dependent learning has yet to be improved. Therefore, we must strengthen the network's capability in learning spatial dependencies to improve semantic segmentation performance in identifying these three kinds of targets in the future.

4. CONCLUSION

In this paper, we proposed a network for RGB-D images classification and also semantic segmentation by full connect CRFs. Although the D images are noisy and have missing data, with the help of the designed network and the loss function, the performance of semantic segmentation maintains a high precision. In future work, the spatial dependencies will be considerate in our network.

REFERENCES

Arbeláez P., Hariharan B., Gu C., et al. 2012. Semantic segmentation using regions and parts. In: *The IEEE Conference on computer Vision and Pattern Recognition*, pp. 3378-3385.

Badrinarayanan V., Handa A., Cipolla R. 2015. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*.

Borgwardt K., Gretton A., Rasch M., et al. 2006. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14), pp.49-57.

Carreira J., Caseiro R., Batista J., et al. 2012. Semantic segmentation with second-order pooling. In: *The IEEE European Conference on Computer Vision*, pp. 430-443.

Chen L., Papandreou G., Kokkinos I., et al. 2016. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*.

Couprie C., Farabet C., Najman L., et al. 2013. Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3573*.

Forestier G., Puissant A., Wemmert C., et al. 2012. Knowledge-based region labeling for remote sensing image interpretation. *Computers Environment and Urban Systems*, 36(5), pp. 470-480.

Gretton A., Borgwardt M., Rasch B., et al 2012. A kernel two-sample test. *Journal of Machine Learning Research*, 13, pp. 723–773.

Gupta, S., Arbelaez, P., Malik, J. 2013. Perceptual organization and recognition of indoor scenes from rgb-d images. In: *The IEEE Conference on omputer Vision and Pattern Recognition*, pp. 564-571.

Gupta S., Girshick R., Arbeláez P., et al. 2014. Learning rich features from RGB-D images for object detection and segmentation. *arXiv preprint arXiv:1407.5736*.

Hu Y., Monteiro S., Saber E.. 2016. Super pixel based classification using conditional random fields for hyperspectral images. In: *The IEEE International Conference on Image Processing*, pp. 2202-2205.

Huang H., Jiang H., Brenner C., et al. 2014. Object-level segmentation of rgbd data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(3), pp. 73.

Koppula, H., Anand, A., Joachims, T., Saxena, A. 2011. Semantic labeling of 3d point clouds for indoor scenes. In: *Advances in Neural Information Processing Systems*, pp.244-252

Kampffmeyer M., Salberg A., Jenssen R. 2016. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In: *The IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1-9.

Krähenbühl P., Koltun V.. 2011. Efficient inference in fully connected crfs with gaussian edge potentials. In: *Advances in neural information processing systems*, pp. 109-117.

Lin G., Shen C., van den Hengel A., et al. 2016. Efficient piecewise training of deep structured models for semantic segmentation. In: *The IEEE Conference on computer Vision and Pattern Recognition*, pp. 3194-3203.

Ioffe, S., Szegedy, C.. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Long J., Shelhamer E., Darrell T.. 2015. Fully convolutional networks for semantic segmentation. In: *The IEEE Conference on computer Vision and Pattern Recognition*, pp. 3431-3440.

Marmanis D., Wegner J., Galliani S, et al. 2016. Semantic segmentation of aerial images with an ensemble of CNSS. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3, pp. 473-480.

Noh H., Hong S., Han B.. 2015. Learning deconvolution network for semantic segmentation. In: *The IEEE International Conference on Computer Vision*. pp. 1520-1528.

Qin A., Clausi D.. 2010. Multivariate image segmentation using semantic region growing with adaptive edge penalty. *IEEE Transactions on Image Processing*, 19(8), pp. 2157-2170.

Shao L., Cai Z., Liu L., et al. 2017. Performance evaluation of deep feature learning for RGB-D image/video classification. *Information Sciences*, 385, pp. 266-283.

Silberman, N., Hoiem, D., Kohli, P., et al 2012. Indoor segmentation and support inference from rgbd images. In: *The IEEE European Conference on Computer Vision*, pp.746-760.

Socher R., Huval B., Bath B., et al. 2012. Convolutional-recursive deep learning for 3d object classification. In: *Advances in Neural Information Processing Systems*. pp. 656-664.

Tao D., Jin L., Yang Z., et al. 2013. Rank preserving sparse learning for kinect based scene classification. *IEEE Transactions on Cybernetics*, 43(5), pp. 1406–1417.

Wang J., Wang Z., Tao D., et al. 2016. Learning common and specific features for rgb-d semantic segmentation with deconvolutional networks. In: *The IEEE European Conference on Computer Vision*, pp. 664-679.

Zaki H., Shafait F., Mian A.. 2017. Learning a deeply supervised multi-modal RGB-D embedding for semantic scene and object category recognition. *Robotics and Autonomous Systems*, 92, pp. 41-52.

Zheng S., Jayasumana S., Romera-Paredes B., et al. 2015. Conditional random fields as recurrent neural networks. In: *The IEEE International Conference on Computer Vision*, pp. 1529-1537.