Depth Estimation and Object Detection for Monocular Semantic SLAM Using Deep Convolutional Network

Changbo Hou, Xuejiao Zhao, Yun Lin*
College of Information and Communication Engineering
Harbin Engineering University
Harbin, China
linyun@hrbeu.edu.cn

***Abstract*—It is still challenging to efficiently construct semantic map with a monocular camera. In this paper, deep learning is introduced to combined with SLAM to realize semantic map production. We replace depth estimation module of SLAM with FCN which effectively solves the contradiction of triangulation. The Fc layers of FCN are modified to convolutional layers. Redundant calculation of Fc layers is avoided after optimization, and images can be input in any size. Besides, Faster RCNN, namely, a two-stage object detection network is utilized to obtain semantic information. We fine-tune RPN and Fc layers by transfer learning. The two algorithms are evaluated on official dataset. Results show that the average relative error of depth estimation is reduced by 12.6%, the accuracy of object detection is improved by 10.9%. The feasibility of the combination of deep learning and SLAM is verified.**

***Index Terms*—monocular depth estimation, indoor object detection, deep learning, ORB-SLAM2**

I. Introduction

Semantic information is essential for intelligent machines to understand the world, e.g. intelligent scene analysis and reasonable route planning for sweeping robots. Although geometric information is acquired to solve the problem of ”Where am I?” and ”What is around me?” through SLAM technology, the function of ” Pick something up around here” cannot be realized. Recently, deep learning has performed well at acquisition of semantic information [1-6]. Therefore, we introduce convolutional neural network to combine with ORB-SLAM2 making full use of the complementarity of geometric information and semantic information.

ORB-SLAM2 provides three sensor interfaces: monocular, binocular and RGB-D. The monocular interface uses a single camera to capture images for local mapping. It is not limited to the weight and volume of the sensor unit. It has the advantages of low cost and extensive application scenarios. The binocular interface obtains the parallax from two cameras to estimate the depth of each pixel, which needs huge calculation [7-10]. RGB-D cameras can directly get depth information, but the price is extremely expensive and the measured distance is limited. The navigation and three-dimensional reconstruction in indoor scenes should reduce costs as much as possible under the premise of ensuring accuracy so that customers’ consumption needs can be pretty met. Thus, we choose monocular camera.

Monocular ORB-SLAM2 via triangulation to complete depth estimation which must ensure translational camera motion, otherwise the epipolar constraint cannot be satisfied. Triangulation is not available when the camera is only rotating. Nevertheless, triangulation contradiction is caused despite the existence of translation, i.e. failed matching will be caused by increasing translation. If the translation is too small, the accuracy becomes worse. For the above problems, FCN is adopted in this paper to automatically learn features layer by layer. Depth information can be given directly from a single image.

Generally, there are two ways to obtain semantic information: semantic segmentation and object detection. Much of papers [4,5,6,11] propose semantic segmentation methods based on deep learning to collect semantic information. However, this method is considerably complicated because the robot needs to focus on the semantic information of each pixel during the motion. Moreover, the robot equipment cannot meet the calculation speed when performing pixel-level semantic classification [12]. Therefore, we select object detection which satisfy the requirements without extra calculation.

In this work, we make monocular camera move in an unknown indoor environment acquiring continuous images sequences. Depth estimation and object detection is respectively implemented through FCN and Faster RCNN. Finally, the three-dimensional semantic map is completed with the help of point cloud tools. In this paper, we lay emphasis on the research of monocular depth estimation and indoor object detection in ORB-SLAM2 rather than paying attention to the pose estimation, loop closing and local mapping. In a word, our main contributions are:

- FCN is adopted to replace ORB-SLAM2 depth estimation module to solve the contradiction of triangulation and overcome the difficulty that depth estimation cannot be realized during pure rotational motion.
- We adjust the parameters of Faster RCNN during training. And a better learning rate of indoor object detection is given. The detection accuracy is improved by 10.9%.

978-1-7281-8915-4/20/$31.00 ©2020 IEEE
DOI 10.1109/QRS-C51114.2020.00051

- NYU depth V2 dataset is 16 times bigger than original dataset with data enhancement, effectively improving the model generalization capability. According to what I know, the previous work [10-11] directly used the official images for training.

II. RELATED WORK

In this section, we first narrate triangulation and monocular depth estimation, and then introduce semantic information acquisition based on deep learning.

A. Monocular Depth Estimation

The monocular interface of ORB-SLAM2 estimates depth value of map points through triangulation [13]. The principle of triangulation is shown in Fig. 1.

O_1 and O_2 are the optical centers of the camera, point P corresponds to the position of a 3D point in the scene, p_1 and p_2 are the feature points of P point on the image I_1 and I_2. Straight line O_1p_1 and straight line O_2p_2 intersect at a point P in the scene. Due to the influence of noise, these two straight lines usually cannot intersect. Least square method is generally used to solve the above problem in ORB-SLAM2. According to the epipolar geometry, let x_1 and x_2 be the normalized coordinate of the two feature points, satisfying:

$$s_1 x_1 = s_2 R x_2 + t \tag{1}$$

s_1 and s_2 are the depth value of the two images to be solved. The rotation matrix R and translation matrix t are obtained from the camera pose estimation. Simultaneously pre-multiply $x_1{}^\wedge$ on both sides in (1), it can be obtained:

$$s_1 x_1{}^\wedge x_1 = s_2 x_1{}^\wedge R x_2 + x_1{}^\wedge t \tag{2}$$

The left side in (2) is zero and the right side is an equation of s_2 which can be solved by the least square method. Similarly, s_1 can be obtained.

It can be seen that the triangulation is obtained by the camera translation. When the camera rotates merely, the epipolar constraint will always be satisfied. Thus, triangulation is invalid. Even if the translation motion exists, it is liable to cause the contradiction of triangulation. The uncertainty of triangulation is shown in Fig. 2. When the feature point is shifted by t, the visual angle changes by $\delta\theta$, and the measured depth value changes by δd. When t is small, the change of δd is not obvious, uncertainty of depth is large, and accuracy of measurement will be bad. When t is large, the change of δd is more obvious, uncertainty of depth is small, match will be easy to failure.

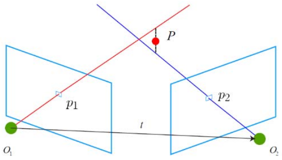

Fig. 1. Triangulation in ORB-SLAM2.

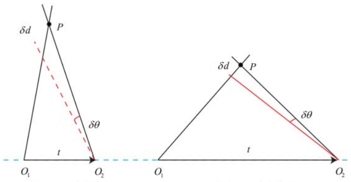

Fig. 2. Uncertainty of triangulation2.

Compared with binocular and RGB-D, monocular has the lowest cost. However, it is most difficult to obtain depth values due to the lack of visual point information. Early researches for depth estimation are based on traditional methods. Saxena et al. [14] proposed the Make3D algorithm. The idea of the algorithm is as follows. Firstly, the image is processed by super pixel segmentation, afterwards the multi-scale local features and the global features of images are extracted from the super pixel block for training the Markov random field. Finally, simulate and establish the depth value of each point in the scene and give the relative depth relationship between points. The algorithm relies on horizontal alignment of images lacking flexibility. Javier [15] proposed a pixel classifier to jointly predict semantic labels and depth information, proving that the information from depth estimation and semantic segmentation can be shared and promoted each other. Liu et al. [16] used super pixels to model the image area and proposed the discrete continuous optimization for depth estimation. [16] is improved by adjusting middle-level area and global scene layout [17]. Nevertheless, this type of method adopts artificial features. The characteristics of these features directly affect the accuracy of the estimation.

Later researches put more and more attention on the convolutional neural network to directly regress 3D depth values from 2D pixel values through a single image. Compared with traditional algorithms, it can achieve depth estimation more excellent. Eigen et al. [18] proposed CNN for monocular depth estimation for the first time. Two scale networks are utilized to directly predict depth value by adjusting image global information and local detail information separately. The disadvantage is that the network parameters are large and the rate of convergence is slow during training. Li et al. Reference [19] improved the transmission of feature information between various scale networks in [18]. The convergence of the network is accelerated by introducing jump layers fused with intermediate layer of each scale network. It also introduces the relative depth limitation of pixels to improve the accuracy of the depth map. Still, FC layers require a fixed input image size, and the image resolution is low. Liu et al. Reference [20] proposed a joint frame called DCNF, which combined CNN and CRF into a unitive frame, and used FCSP to greatly optimize the processing speed of the frame. Considering the continuity of depth values, Xu et al. Reference [29] proposed a deep convolutional neural field model combined with continuous CRF which was used to improve depth estimation. The optimal solution of the log-likelihood can be obtained by analyzing and calculating the partition function in the probability density function. Accuracy is effectively improved at the edges and outlines. But the way to gain depth is no longer an end-to-end neural network.

In this paper, an end-to-end network FCN is selected to achieve depth estimation. Unlike [18], FCN no longer contains FC layers. images of any size can be input. It is not necessary that each training image and test image has the same size. Furthermore, the network avoids the problem of repeated memory and redundant calculation caused by dividing the image into many small image blocks for training the network. Data processing is more efficient.

B. *Semantic Information Acquisition*

Image segmentation can be used to gain pixel-level semantic information. Some researches add a decoding network based on the convolutional neural network to implement semantic segmentation. Efficiency models appear successively, such as SegNet, ENet, PSPNet and SPNet [1,2,3]. Afterwards, some papers [4,5,6] transformed the 2D semantic information to 3D semantic information that make he classification judgment of 3D points of the target in the map is realized. The obtained semantic information through semantic segmentation is more fit for complex outdoor environments, such as autonomous driving. In this paper, we hopes that robots can pick up the certain object, which does not need to infer the label of each pixel. There is no need to infer the label of each pixel. Therefore, we choose object detection to obtain indoor semantic information.

Girshick et al. [21] proposed RCNN for the first time introducing convolutional neural network into the field of object detection, and then improved and proposed Fast RCNN [22] which feature extraction of CNN is performed only once for the entire image. The repeated calculation of features in RCNN is reduced. Also, classification and regression are unified to improve the speed and accuracy of detection. In addition, [22] proposed a multi-task model that shared parameters between the classification layers and position regression layers to promote each other. However, it is time-consuming for Fast RCNN to find all candidate boxes with selective search. In order to solve the above problem, Ren et al. Reference [23] proposed Faster RCNN abandoning the traditional method of candidate box extraction. RPN is adopted to extract candidate boxes greatly improved the speed during training. The above algorithm is the two-stage object detection algorithm which performs almost perfect at accuracy. The emergence of the one-stage object detection algorithm make speed perform well. Redmon et al. Reference [24] proposed the YOLO algorithm no longer extracting candidate boxes. Image classification and object positioning was integrated into one network through Fc layers. The speed has been extremely improved while the accuracy is lower than Faster RCNN. Liu et al. proposed the SSD algorithm via appending multiple convolution layers of different scales [25]. It not only faster than YOLO but also guarantee the accuracy of Faster RCNN.

We hope to obtain more precise semantic information so that robots can better complete tasks like obstacle avoidance. Obviously, Faster RCNN is more suitable. According to the character of indoor objects, we modify the training parameters and fine-tune the RPN and Fc layers by transfer learning, a better learning rate for our datasets is given.

III. MONOCULAR SEMANTIC SLAM

A. *System Overview*

The overview of our indoor monocular semantic SLAM system is shown in Fig. 3. The input is a series of monocular images, and the output is a 3D semantic map with data association. The ORB-SLAM2 framework tracks the thread to locate the camera and chooses whether to add key frames. Next, extraction and matching for ORB feature, pose estimation, nonlinear optimization, and local mapping is performed. Meanwhile, FCN for monocular depth estimation is employed to obtain depth values. According to the results from FCN, the RGB-D interface of ORB-SLAM2 completes pose estimation and subsequent functions through 3D-2D, which no longer requires a cumbersome initialization process. Faster RCNN for object detection is applied to obtain semantic information. We select common indoor objects and complete detection of five kinds of object. Referring to the results from Faster RCNN, data association correspond 2D semantic points to 3D geometric points. Each key frame connects to one or more 2D objects and each 3D map point connects to one object. 3D map points are indirectly connected to one or more objects through key frames, thereby realizing the conversion from 2D semantic points to 3D semantic points. In the end, the point cloud tool is used to generate visualize monocular semantic map. In this paper, the main work is to complete monocular depth estimation based on FCN and indoor object detection based on Faster RCNN.

B. *Depth Estimation*

FCN is a semantic segmentation network proposed by Long et al. [26] which can output the classification of each pixel. Monocular depth estimation also needs to process each pixel, while the output is the depth regression value of each pixel. Enlightened by image segmentation, FCN-8 shown in Fig. 4(a) is adopted to achieve monocular depth estimation.

The backbone network of FCN is VGG16, including 13 Conv layers, 5 Pool layers and 3 Fc layers. The last three Fc layers respectively correspond to one-dimensional vectors with dimensions 4096, 4096, 1000, where 1000 is the classification number. In fact, the Fc layers and the Conv layers can be converted to each other because the function form of neurons is the dot product. Undoubtedly, any Fc layer can be converted into a Conv layer, e.g. the dimension of a Fc layer is K, let the input be $m \times n \times n$, where m is the number of channels, and n is the size of the feature map, this Fc layer can be equivalent to a convolutional layer of a $n \times n$ filter with dimension of K, where $Pad = 0$ and $Stride = 1$, the output is $K \times 1 \times 1$. That is to say, the size of the filter is consistent with the size of the input feature map. The conversion of the Fc layers is shown

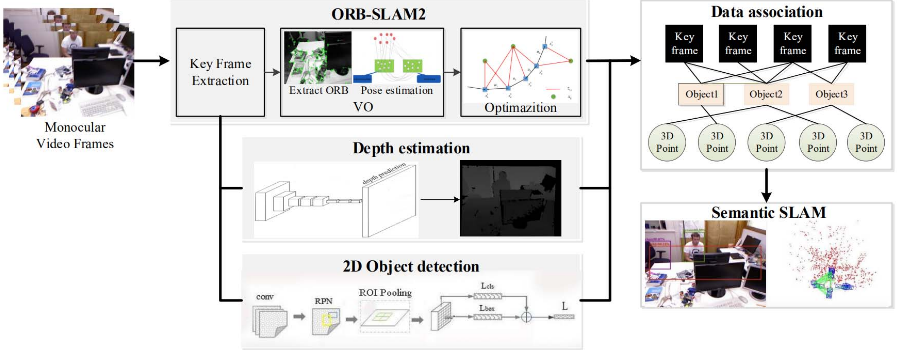

Fig. 3. Overview of our system.

in Fig. 4(b). The three Fc layers of $4096 \times 1 \times 1$, $4096 \times 1 \times 1$, $1000 \times 1 \times 1$ are transformed to 1×1 Conv layers of 4096, 4096, 1000 which no longer require the input size. Although this process does not reduce the network parameters, it can receive input images of any size. All the 13 Conv layers use 3×3 kernel of $stride = 1$, $pad = 1$. The pooling method is 2×2 maximum pooling of $stride = 2$. ReLU as an activation function is applied after each convolution layer. According to the calculation formula of the output size of the feature map:

$$o = (i + 2p - k)/s + 1 \tag{3}$$

According to (3) the output resolution is 1/32 of the input during the forward propagation of the network, i.e. the output is 7×9 when the input is 228×304. If using interpolation to directly restore the original size, inevitable image distortion will be caused. Obviously it is not allowed to ORB-SLAM2. The upsample layer can avoid image distortion as much as possible through keeping on learning in network training. The essence of upsample is transposed convolution, as shown in Fig. 4(c). Relationship between input and output is satisfied:

$$o = s \times (i - 1) + k - 2p \tag{4}$$

The essence of upsample is still a type of convolution calculation comparing (2) and (3), but the output size is larger than the input size. FCN-32 directly expands the output by 32 times through bilinear interpolation, which is particularly rough. FCN-16 increases pool5 by 2 times through upsample layers and adds it to pool4 result. FCN-8 is shown in Fig. 4(a) including feature messages of three layers of pool3, pool4, and pool5. The layer fusions make the output more detailed. All the upsample layers are trained in the network like conv layers, while bilinear interpolation in the last step does not participate in the training process.

The Huber function is chosen as the loss function, expressed as:

$$L(x) = \begin{cases} |x|, |x| \le c \\ \frac{x^2+c^2}{2c}, |x| > c \end{cases} \tag{5}$$

Where $L(x)$ is the target value of the loss function, $x = \tilde{y} - y$, $\tilde{y}$ is the predicted value in the network, y is the true value. The parameter c is calculated by $c = \frac{1}{5}\max_i(|\tilde{y}_i - y_i|)$. When x is between $-c$ and c, $L(x)$ is $L1$ norm. When x is outside this range, $L(x)$ is the $L2$ norm. $L(x)$ is continuous at point c. For each gradient descent, the value of c is calculated first, then the value of $L(x)$ is solved.

C. Object Detection

The network structure of Faster RCNN is shown in Fig. 5, which is composed of five modules: backbone network, RPN, ROI Pooling layer, softmax classifier, NMS. VGG16 is adopted as the backbone network for preliminary feature extraction, RPN extracts candidate regions, ROI Pooling layer further extracts features from the candidate regions, the classification result is given by softmax classifier, position result is given by NMS. RPN is employed to complete object detection in two-stage. Although the extra calculation is introduced, the accuracy is better compared with the one-stage network. This is exactly what we expect.

The loss function of RPN in Faster RCNN is given as:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{\text{reg}}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{6}$$

Equation (6) is composed of classification loss function L_{cls} and regression loss function L_{reg}, where L_{reg} is expressed as:

$$\begin{aligned} L_{reg}(t_i, t_i^*) &= \sum_{i \in \{x,y,w,h\}} smooth_{L_1}(t_i - t_i^*) \\ smooth_{L_1}(x) &= \begin{cases} 0.5x^2, |x| \le 1 \\ |x| - 0.5, otherwise \end{cases} \end{aligned} \tag{7}$$

Where i represents the number of anchors. When anchor is the target to be detected, $p_i = 1$ otherwise $p_i = 0$. t_i^* represents the coordinate of bounding box related with the positive sample anchor. The selection of positive sample bounding box is based on the IOU threshold. In this paper, IOU is set to 0.7. If the IOU value of the bounding box is higher than 0.7, it is considered as a positive sample. If the IOU value of the bounding box is less than 0.3, it is considered a negative sample, namely the background. t_i is the coordinate of the predicted bounding box from RPN relative to the bounding box from anchor:

$$\begin{aligned} t_x &= (x - x^a)/w^a, t_y^* = (y - y^a)/h^a \\ t_w &= \log(w/w^a), t_h = \log(h/h^a) \end{aligned} \tag{8}$$

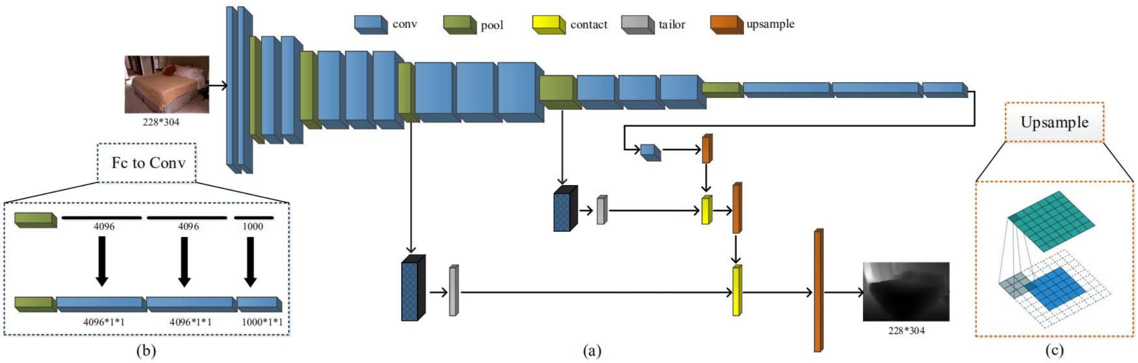

Fig. 4. FCN-8. (a) detail architecture. (b) transformation from Fc to Conv. (c) upsample process.

t_i^* is the coordinate of the real object box in the original image relative to the bounding box from anchor:

$$\begin{aligned} t_x^* &= (x^* - x^a)/w^a, t_y^* = (y^* - y^a)/h^a \\ t_w^* &= \log(w^*/w^a), t_h^* = \log(h^*/h^a) \end{aligned} \tag{9}$$

The purpose of training is to make t_i and t_i^* infinitely close.

We utilize transfer learning in Faster RCNN network training process, using the parameters trained by VOC to initialize the weights. At the beginning of the training, the parameters of the Conv layers shared by the RPN and the Faster RCNN are fixed, and only fine-tune the parameters of the specific layers of RPN. Keeping the parameters of the Conv layers shared by the RPN and Faster RCNN network as well as the parameters of the specific layers of RPN constant, only fine-tune the parameters of specific Fc layers in Faster RCNN. However, the initial learning rate ($lr = 2e - 4$) of transfer learning on outdoor VOC data is not apply to our indoor objects. Therefore, we modified the hyperparameters and set different initial learning rates for training to find out a better learning rate for our network so that the accuracy of indoor detection can be improved. 5 kinds of object is achieved: Person, Monitor, Table, Chair and sofa (corresponding label: person, tvmonitor, diningtable, chair, sofa).

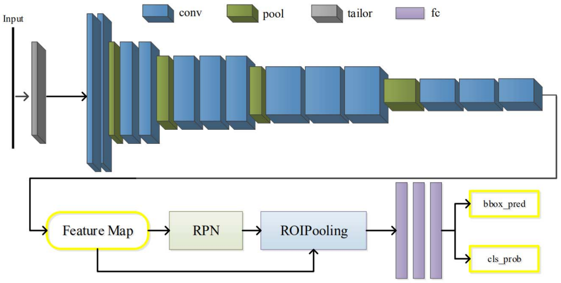

Fig. 5. Structure of Faster-RCNN.

IV. EXPERIMENTS RESULTS AND ANALYSIS

In this section, we elaborate the experimental setup and dataset of monocular depth estimation and indoor object detection, using the official evaluation standard for verification. The performance of algorithms in this paper is analyzed.

A. Experimental Setup

Ubuntu16.04 system, 16G RAM, CPU of i7. TensorFlow, a kind of mainstream frameworks for deep learning is used to build FCN and Faster RCNN in this paper. We train and test on NVIDIA GTX 2080Ti with 12G of GPU memory.

B. Dataset

In this paper, we built our own dataset of monocular depth estimation and indoor object detection based on the NYU Depth V2 dataset. Original RGB images are given in Fig. 6. The second column in Fig. 6 shows the depth labels of NYU Depth V2 dataset. NYU DepthV2 dataset is a series of image sequences captured by Microsoft Kinect camera, including 2898 images of 640×480 size in 464 indoor scenes. It consists of 1449 pairs of images: 1449 RGB images (.jpg format) and 1449 depth labels (.png format). We expand the NYU depth V2 dataset with data enhancement:

- Scale: 3 kinds of sizes of 200×150, 400×300, 800×600.
- Rotation: 3 kinds of rotation of 180 degrees, clockwise 90 degrees, counter-clockwise 90 degrees.
- Crop: 2 kinds of sizes with random crop.
- Color: 4 kinds of RGB values with random multiplication.
- Brightness: 4 kinds of brightness with random adjustment.

The training data is expanded 16 times from 1449 to 23184 (called dataset1) through the above five ways. The obtained 2D image will contain the various forms of rotation, translation, scale, and color because the camera will produce a variety of gestures during the movement process in 3D space. The model should also perform an outstanding capability in this case. In addition, object labels are given base on 1449 RGB images of NYU Depth V2 dataset, as shown in Fig. 6. We finally use a total of 18574 images including the 17125 images of PASCAL VOC2012 as the indoor object detection dataset (called dataset2).

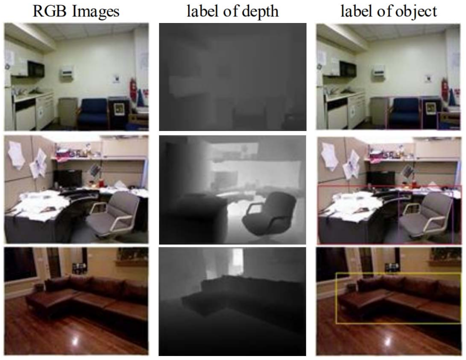

Fig. 6. Our dataset based on NYU. Displayed are RGB images (the first column), labels for depth estimation (the second column) and labels for object detection(the third column).

C. Depth Estimation

The size of RGB images from dataset1 is decreased from 640×480 to 320×240 in order to reduce the calculation cost before training in the depth estimation network. Afterwards, RGB images with 304×228 size is obtained by random center cropping, which is used as network input for training. The images of depth labels are processed in the same way so that the resolution is the same as the resolution finally predicted by the network and performance can be evaluated. 1000 images are randomly selected as the test data, the remaining images are used as the training data. The FCN for depth estimation uses SGD for training. The loss function in (5) is adopted and the entire network is end-to-end with batch training. The batch size is set to 4, the momentum is set to 0.9. A total of 100K iterations is carried out, which takes about 2.5 days. The initial learning rate is 0.0001 and it is reduced to 1/10 of the original value every 10K iterations. The loss value is recorded every 100 iterations. Compared with CNN in [18], the loss curve is shown in Fig. 7. It can be seen from Fig. 7(a) that both network training loss values can quick convergence and continuous decline with the number of iterations increases, but the FCN is better than CNN. Fig. 7(b) gives more detailed results of the 2K-14K iterations, showing that FCN converges faster. Fig. 7(c) gives more detailed results of 96.4K-97.2K iterations. It can be seen that the minimum loss of [18] reaches about 11, while the minimum loss of FCN can reach about 5. The loss values of FCN performed more excellently in the final stage of training.

Table I gives the test results in this paper and compares with algorithms of other papers. It can be seen from Table I that the errors of Liu [16], Li [32] and Liu [20] using traditional algorithms of feature extraction are higher than Eigen[18] and FCN. Reference [20] used the AlexNet, δ_1 is increased to 0.614 which is slightly higher than 0.611 in [18]. But other performances in [18] are better than [20] because AlexNet used in [20] is shallow. Comparing [18] and FCN, the rel is decreased from 0.215 to 0.188, a difference of 0.027, an increase of 12.6%. δ_3 is extremely close, δ_2 is equal to 0.891 and δ_1 is equal to 0.652 in this paper, which are higher than [18]. The overall performance of FCN is better.

TABLE I
COMPARISON RESULTS OF DIFFERENT ALGORITHMS

algorithm	rel	rms	$\log_{10}$	δ_1	δ_2	δ_3
Liu et al. [16]	0.335	1.060	0.127	–	–	–
Li et al. [32]	0.232	0.821	0.094	0.621	0.886	0.968
Liu et al. [20]	0.230	0.824	0.095	0.614	0.883	0.971
Eigen et al. [18]	0.215	0.907	–	0.611	0.887	0.971
Our	0.188	0.754	0.079	0.652	0.891	0.972

D. Object Detection

In dataset2, 3000 images are randomly chosen as the test data, the remaining images are used for training. The entire network is end-to-end. The batch size is set to 1, the momentum is set to 0.9, and the weight updating method uses the Adam gradient descent algorithm with adaptive learning rate [31]. Manually set the initial learning rate, and use dynamic attenuation ($lr = b * \gamma^{([\frac{n}{s}])}$,where b represents the initial learning rate, γ represents the attenuation, s represents the stride of decay, and n represents the number of batches trained currently.) to decay the learning rate to gradually make learning effect gradually converge. During RPN training, set the IOU threshold to 0.7 and 0.3 to select whether the bounding box is a positive sample or a negative sample. In the NMS, the threshold of the intersection region with the current highest box is set to 0.7. A total of 100k iterations is set. The training of the whole network takes about 1 day. Three levels of initial learning rate($e{-}3$, $e{-}4$, $e{-}5$) are set for training, the loss curve is shown in Fig. 8. Fig. 8(a) shows the overall loss trends of the three initial learning rates. The node whose loss value jumps sharply is the starting point of network training. The different training parameters of the three kinds of learning rate leads to different sudden nodes. There is no curve where the loss value can be always kept at the lowest during training, but the loss value of $lr = 2e{-}3$ is lower at the most moments. More detail results of loss comparison are given in Fig. 8(b) and Fig.8(c). It can be seen from Fig. 8(b)(c) that the loss value of $2e{-}3$ is always lower than 0.3 at 60K-65K iterations, the loss value is always lower than 0.15 at 77K-80K iterations, which works better in training process. The results of $2e{-}4$ and $2e{-}5$ are similar. $2e{-}5$ is better at 60K-65K iterations, and $2e{-}4$ is better at 77K-80K iterations. It can be concluded that the initial learning rate in this paper performs better on the level of $e{-}3$. Finally, the initial learning rate is set to $e{-}3$ in this paper.

TABLE II
MAP RESULTS FOR EACH CLASS

learning rate	person	monitor	table	chair	sofa
$lr = 2e{-}3$	0.879	0.650	0.519	0.523	0.623
$lr = 2e{-}4$	0.762	0.565	0.491	0.490	0.573
$lr = 2e{-}5$	0.691	0.576	0.383	0.518	0.550

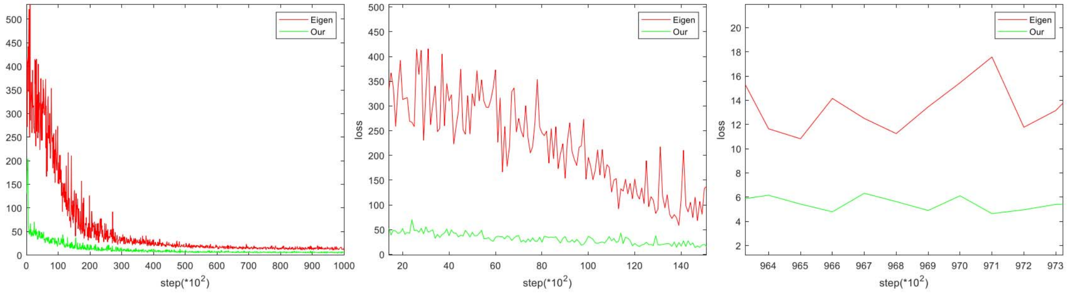

Fig. 7. Comparison of loss function during training between CNN and FCN. Called (a), (b) and (c) from left to right. (a) global loss value. (b) detailed value of convergence. (c) detailed value at the end of training.

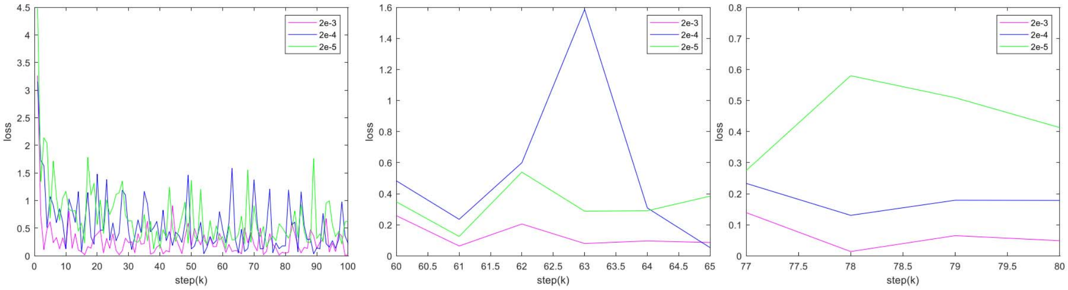

Fig. 8. Loss curves of three levels of initial learning rates. Called (a), (b) and (c) from left to right. (a) global loss value. (b) detailed value at 60K-65K iterations. (c) detailed value at 77K-80K iterations.

TABLE III
THE OVERALL PERFORMANCE OF DIFFERENT ALGORITHMS

network	mAP	FPS
Faster RCNN	0.639	9
SSD	0.612	23

3000 images were tested in this paper and the comparison results of the detection accuracy of each type under different lr are given in Table II. It can be seen from the second rows and third rows of Table II, $2e{-}4$ is slightly higher than $2e{-}5$ for the three kinds of object (person, table, and sofa), $2e{-}5$ is a little higher than $2e{-}4$ for two kinds of object (monitor and chair). Compared the fourth rows with the second and third rows in Table II, the mAP of each class in $2e{-}3$ is better than $2e{-}4$ and $2e{-}5$, which is improved by 10.9%. Because the initial learning rate of this level is more suitable for the study of indoor goals in our dataset. The missed detection rate and the false detection rate are reduced, thereby the accuracy is improved. Therefore, the accuracy of object detection can be effectively improved through adjusting the appropriate parameters according to the target characteristics. In addition, Table III shows the overall performance comparison with the one-stage network. It can be seen from Table III that the mAP of SSD is 0.612, which is lower than Faster RCNN. However, the detection speed can reach 23 frames per second, while Faster RCNN only perform 9 frames per second. The speed of SSD is much faster than Faster RCNN because SSD belongs to one-stage object detection network. It can generate candidate boxes directly via one step without extra calculation for candidate boxes in RPN. The visualization results of object detection are shown in Fig. 9.

V. CONCLUSIONS AND FUTURE WORK

In this work, a novel semantic map can be built benefiting by the complementarity between deep learning and ORB-SLAM2. Compared with the traditional method of triangulation in ORB-SLAM2, the method in this paper can directly estimate the depth value from a single image. It avoids the clumsy initialization process in monocular SLAM and does not requires fixed input compared to CNN. In addition, we utilize transfer learning to fine-tune the RPN in Faster RCNN. It shows that reasonable design of hyperparameters, especially the initial learning rate, can effectively improve the accuracy of object detection.

No matter what algorithm is adopted, the value of depth prediction is difficult to reach the same as the real value. In future work, the network needs to be improved to promote the accuracy of depth estimation. Besides the 2D semantic points obtained in this paper needs to be associated with 3D geometric points. Still, an extra step is generated. Reference [4,5,28] proposed an end-to-end deep learning network for the 3D point cloud, which researched on the 3D voxel level. The semantic classification and 3D geometric box were given. 3D semantic information can be obtained directly. In the future, our research will focus on the 3D end-to-end network to establish a more efficient 3D semantic map.

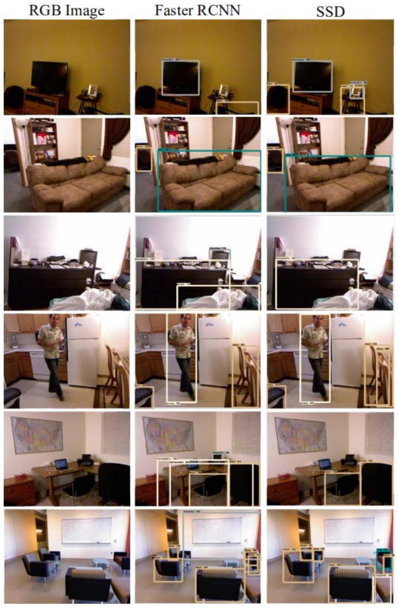

Fig. 9. Visual display of object detection. Displayed are RGB images (the first column), results of Faster RCNN (the second column) and results of SSD (the third column).

Acknowledgment

This work is supported by the National Key Research and Development Program of China under Grant 2018AAA0102702, Natural Science Foundation of Heilongjiang Province (JJ2019LH2398), Key Laboratory of Advanced Marine Communication and Information Technology, Ministry of Industry and Information Technology, Harbin Engineering University, Harbin, China.

References

[1] V. Badrinarayanan, A. Kendall, R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 99, 2017.

[2] W. Zhou, A. Zyner, S. Worrall, E. Nebot, "Adapting Semantic Segmentation Models for Changes in Illumination and Camera Perspective," IEEE Robotics and Automation Letters,vol. 4, no. 2, pp. 461-468, 2019.

[3] Q. Hou, L. Zhang, M. M Cheng, J. Feng, "Strip Pooling: Rethinking Spatial Pooling for Scene Parsing," 2020.

[4] Q. Hu, B. Yang, L. Xie, et al., "RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds," 2019.

[5] Z. Huang, Y. Yu, J. Xu, F. Ni, X. Le, "PF-Net: Point Fractal Network for 3D Point Cloud Completion," 2020.

[6] Q. Liu, R. Li, H. Hu, D. Gu, "Indoor Topological Localization Based on a Novel Deep Learning Technique," Cognitive Computation, pp. 1-14, 2020.

[7] G. Clement, M. A. Oisin, J. B. Gabriel, "Unsupervised monocular depth estimation with left-right consistency," The 29th IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, pp. 19-21, June 2016.

[8] Y. D. Zhang, G. Ravi, S. W. Chamara, et al., "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," The 31th IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, pp. 19-21, June 2018.

[9] Z. Su, L. Lu, F. Yang, X. He, D. Zhang, "Geometry constrained correlation adjustment for stereo reconstruction in 3D optical deformation measurements," Optics Express, 2020.

[10] F. Liu, SB. Zhou, YL. Wang, X. He, D. Zhang, "Binocular Light-Field: Imaging Theory and Occlusion-Robust Depth Perception Application," IEEE TRANSACTIONS ON IMAGE PROCESSING, 2020.

[11] J. Mccormac, A. Handa, S. Leutenegger, A. Davison, "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks," IEEE Int Conf Robot Automat (ICRA), pp. 4628–463, May/June 2016.

[12] L. Zhang, L. Wei, P. Shen, et al., "Semantic SLAM based on Object Detection and Improved Octomap," IEEE Access, 2018, pp. 1-1.

[13] J. Davis, D. Nehab, R. Ramamoorthi, S. Rusinkiewicz, "Spacetime stereo: a unifying framework for depth from triangulation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 2, pp. 296-302, 2005.

[14] A. Saxena, M. Sun, A. Y. Ng, "Make3D: learning 3D scene structure from a single still image," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, pp. 824-840, 2008.

[15] A. Javier, Montoya-Zegarra, D. Jan, Wegner, Ľubor Ladický, "Mind the Gap: Modeling Local and Global Context in (Road) Networks," German Conference on Pattern Recognition. Springer International Publishing, 2014.

[16] M. Liu, M. Salzmann, X. He, "Discrete-continuous depth estimation from a single image," IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2014, pp. 716-723.

[17] W. Zhuo, M. Salzmann, X. He, F. Liu, "Indoor scene structure analysis for single image depth estimation," Computer Vision and Pattern Recognition. IEEE, 2015, pp. 614-622.

[18] D. Eigen, C. Puhrsch, R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," Computer Science, 2014, pp. 2366-2374.

[19] J. Li, R. Klein, A. Yao, "Learning fine-scaled depth maps from single RGB images," 2016.

[20] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image" In Proc. Conf. Computer Vision and Pattern ecognition (CVPR), pp. 5162–5170, 2015.

[21] R. Girshick, J. Donahue, T. Darrell, J Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[22] R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision(ICCV), 2015, pp. 1440–1448.

[23] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Advances in neural information processing systems, 2015, pp. 91–99.

[24] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.

[25] W. Liu, D. Anguelov, D. Erhan, et al., "SSD: Single Shot MultiBox Detector," European conference on computer vision, 2016, pp. 21–37.

[26] J. Long, E. Shelhamer, T. Darrell, "Fully convolutional networks for semantic segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39 no. 4, pp. 640-651, 2014.

[27] K. He, X. Zhang, S. Ren, J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," European Conference on Computer Vision. Springer International Publishing, 2014, pp. 346-361.

[28] Q. Fan, W. Zhuo, C. K. Tang, et al., "Few-Shot Object Detection with Attention-RPN and Multi-Relation Detector," 2019.

[29] D. Xu, R. Elisa, W. L. Ouyang, X. Wang, N. Sebe, "Multi-scale continuous CRFs as sequential deep networks for monocular depth estimation," The 30th IEEE Conference on Computer Vision and Pattern Recognition,Hawaii,USA, pp. 22-25, July 2017.

[30] S. Zhang, C. Chi, Y. Yao, Z. Lei, S. Z. Li, "Bridging the Gap Between Anchor-based and Anchor-free Detection via Adaptive Training Sample Selection," 2019.

[31] F. Y. Wang, J. J. Zhang, X. Zheng, et al., "Where does AlphaGo go: from church-tuning thesis to AlphaGo thesis and beyond," IEEE/CAA Journal of Automatica Sinica, vol. 3, no. 2, pp. 113-120, 2016.

[32] Y. Bai, L. Fan, Z. Pan, L. Chen, "Monocular Outdoor Semantic Mapping with a Multi-task Network," 2019.

[33] H. Long, Y. Yuan, W. Qingjun, G. Fei, G. Yong, "Indoor scene segmentation based on fully convolutional neural networks," Image and Graphics, 2019.