Fusing Object Context to Detect Functional Area for Cognitive Robots

Hui Cheng[1], Junhao Cai[1], Quande Liu[2], Zhanpeng Zhang[3]*, Kai Yang[1], Chen Change Loy[2], Liang Lin[1,3]
[1]Sun Yat-Sen University
[2]The Chinese University of Hong Kong
[3]SenseTime Group Limited

***Abstract*— A cognitive robot usually needs to perform multiple tasks in practice and needs to locate the desired area for each task. Since deep learning has achieved substantial progress in image recognition, to solve this area detection problem, it is straightforward to label a functional area (affordance) image dataset and apply a well-trained deep-model-based classifier on all the potential image regions. However, annotating the functional area is time consuming and the requirement of large amount of training data limits the application scope. We observe that the functional area are usually related to the surrounding object context. In this work, we propose to use the existing object detection dataset and employ the object context as effective prior to improve the performance without additional annotated data. In particular, we formulate a two-stream network that fuses the object-related and functionality-related feature for functional area detection. The whole system is formulated in an end-to-end manner and easy to implement with current object detection framework. Experiments demonstrate that the proposed network outperforms current method by almost 20% in terms of precision and recall.**

I. INTRODUCTION

Cognitive robot needs to locate the desired operation area before it performs the actual action. For example, when a robot aims to open a drawer, it needs to decide whether a spherical or wrap grasp to perform and where to perform such an action, according to the shape and location of the drawer handle. Given an input scene image, the localization and recognition of such operation area can be termed as functional area (affordance) detection as proposed in [24]. With the functional knowledge, the robot can interact with human and objects through different actions and tasks. This problem is challenging due to the large appearance variation in the real world. For example, for a function of "spherical grasp", the target area can be a handle of door, drawer or even other spherically shaped artifacts.

Deep learning has achieved significant success for object classification and detection [9], [10], [19]. A simple solution to solve the functional area detection problem is to extract some potential regions from the image, and then classify the functionality of these regions by a deep convolutional neural network (CNN) as in [24]. There are two drawbacks taking this approach. Firstly, this method only uses the feature from the image region and ignores the context of the area, which is critical to cope with appearance variations caused by occlusion and viewpoint changes (see Fig. 1). For example, a water tap/valve is usually located near the bottom of the faucet. Detecting the larger faucet can help locating the valve, which is otherwise hard to detect. Secondly, deep learning usually requires a large amount of training data and annotating a large functional area detection dataset is laborious and expensive.

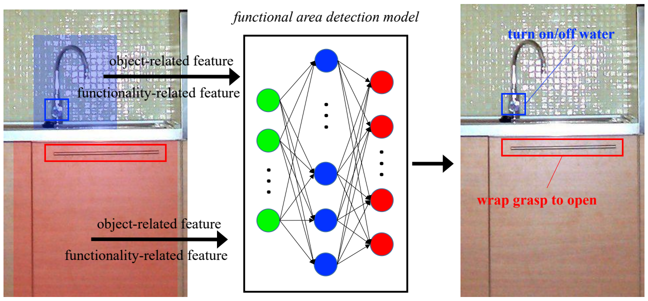

functional area detection model
object-related feature
functionality-related feature
object-related feature
functionality-related feature
turn on/off water
wrap grasp to open

Fig. 1. Object context features (extracted from the shaded areas) are employed to facilitate functional area detection indicated by the bounding boxes.

We believe that functional area detection can be benefited from better usage of contextual knowledge or prior. In fact, functional area detection is not a standalone problem. It is different but highly related to the object detection problem. For example, if a robot can detect a door, it is likely that there should be an area where it can perform a pull or a push action. In particular, the object information of the image region or surrounding area can be used as useful prior for functionality inference. In addition, object classification and functionality decision share some similar image features, such as the image edge and shape. Given the large amount of existing object dataset, we can transfer the knowledge learned from these datasets to the new system for functionality description. This can reduce the requirement of training data for the new system.

To this end, we propose a two-stream network architecture for functional area detection. One of the network streams is trained using an object detection dataset, such as COCO [12], to learn object-related representation. The other stream learns the functionality-related feature. Each stream is built based on the current state-of-the-art object detection model [19]. The features extracted by these two streams are then fused for functionality inference. The contributions of this work include: 1) This is the first attempt to incorporate object knowledge for functional area detection; 2) A novel two-stream network is proposed to fuse object-related and

*Corresponding author

functionality-related features to solve this problem. Experiments demonstrate the effectiveness of employing object related features in the task of functional area detection and the superior performance of our method compared to current approaches.

II. RELATED WORK

A. Object Attribute Classification

To determine the functionality of an area is similar to the problem of object attribute classification, which has been extensively studied in the computer vision and robotics community. For example, in face and human analysis [14], [11], algorithms discover the attributes such as 'gender', 'race' and 'hair style'. Other attribute analysis tasks, such as cloth color and style for fashion search [25], are also popular research topics. In the robotics community, there are extensive studies on the object color, shape and material attributes recognition in RGB-D data [20].

Object affordance, another kind of attributes, is more related to our study. In particular, the affordance of an object can be viewed as a function of interactions with human or other agents [8]. For example, Pieropan et al. [17] investigate object categorization according to function and learns the affordances of objects from human demonstration, such as 'readable' and 'drinkable'. Recently, Myers et al. [15] detect the tool parts' affordance in RGB-D images using hand-crafted geometrical features. The problem studied in our work is related to but different from these problems in that the definition here is not object-centric but area-centric. The algorithm needs to detect the functional area that can be either an object or a small part of an object. This problem is more challenging since the features of the object part may not be as discriminative as that of the whole object.

B. Object Detection

Another related topic is object detection since we also need to locate the functional area besides classification. Object detection also attracts extensive research interests in image semantic understanding. In early attempts, an algorithm typically applies a sliding window on the whole image and performs classification on each location. To handle scale variations of objects, image pyramid is often introduced to the original input image. Deformable part based model (DPM) [6] is also a classic method for object detection. In particular, each object is denoted as a collection of parts arranged in a pictorial structure. Each part is described by hand-crafted feature, such as HOG [4]. Recently, deep learning technique has shown substantial progress in image recognition by learning high-level feature abstraction. There are multiple deep learning based detection methods that show promising results, such as faster R-CNN [19], YOLO [18] and SSD [13]. The main idea of these approaches is to use a deep convolutional neural network (CNN) for learning object representation from scratch given raw images and the corresponding annotations rather than designing features by hand.

C. Functional Area Detection

The problem of functional area detection is proposed in [24] recently, where a specific definition and dataset is also presented. To solve this problem, the authors in [24] propose a two-stage approach. The input of the system is a static image of indoor scene. In the first stage, the system uses a visual attention method called selective search [21] to propose a set of bounding boxes that are potentially to be a functional area. Selective search is mainly built on a diverse amount of visual features, such as color, intensity and edge information. In the second stage, a deep network takes each of the bounding boxes from the first stage as input. The output is the probability that an area belongs to a certain class of functionality. This deep network is first trained on a large external general image dataset and the parameters are fine-tuned on the collected functional area dataset. Our method differs to this algorithm since we explicitly employ the object-related features learned from other object dataset without additional annotation and we formulate a new end-to-end two-stream network structure that is easier to train and implement. In the experiments, we also observe substantial improvement with our method.

III. APPROACH

This section presents the proposed approach. Firstly, we formulate our problem and show the specific definition of the functional area in Sec. III-A. Then we illustrate how we can employ a faster R-CNN [19] framework to perform multi-scale functional area detection. Based on the faster R-CNN approach, in Sec. III-C, we present our proposed two-stream network that integrate both the object-related and functionality-related feature for area detection. The training approach for this network is then described in Sec. III-D. It is worth pointing out that faster R-CNN [19] is adopted here given its good performance on generic object detection. Other detection frameworks can be similarly applied. The important contribution of this work is the notion of exploiting object contextual information for functional area detection.

A. Problem Definition

To formulate the functional area detection problem, we follow the setting of [24] and assume that the robot takes a static indoor image as input, and outputs a collection of rectangles that contain a target area, each with the corresponding functionality label. There are some previous works that define the functionality ontologies. In particular, Worgotter et al. [22] categorize manipulation actions into some basic types according to hand-object relations. The authors of [24] further study the common set of a robot can perform in an indoor environment and propose a robot functionality ontology, as shown in Fig. 2.

In general, there are three main categories: 'Small part of furniture/appliance/wall', 'Objects', and 'Furniture'. For each category, there are further defined main functions. For example, for 'Small part of furniture/appliance/wall', we have 'Open' for the function of grasp, and 'Turn on/off' for the intended media. After the main function, it is the end

Functional Area	Main Function	End Category	Symbol	Example
Small part on a furniture/ appliance wall	Open	Spherical grasp to open		Doorknob
		Wrap grasp to open		Door handle
	Turn on/off	Turn on/off electricity		Electrionical switch
		Turn on/off water		Tap
		Turn on/off fire		Gas stove switch
Object (vessels and tools)	Move	Two hands raise and move		Bowl, Basin
		Cylindrical grasp to move		Bottle
		Hook grasp to move		Suitcase handle
		Pinch grasp to move		Paper, towel
	Manipulate	Manipulate elongated tools		Screwdriver
Furniture	Use of furniture	To sit, to place and etc.		Chair, sofa

Fig. 2. Functionality ontologies studied in this work. The definitions are originally made in [24].

category, such as ‘Spherical grasp to open’ and ‘Wrap grasp to open’. In total, we have 11 end categories and thus the objective is to detect such 11 kinds of areas given the input image. To facilitate the visualization, each end category is accompanied with a specific symbol as shown in Fig. 2.

B. Multi-scale Detection in an End-to-End Network

According to the definition in Sec. III-A, a functional area can be an object or only a small part of the object. That means the area can be either small or large in the whole image. The large scale variation imposes extra challenges for the algorithm. This problem is also critical for object detection. Early attempts to solve this problem is to apply a sliding window over the image pyramid. However, this may not be efficient for some applications. Faster R-CNN [19] object detection framework is a promising method to solve this problem.

Figure 3 illustrates the process of faster R-CNN in detection. In general, faster R-CNN is composed of two modules. The first module takes an image as input, and generates the regions of interest. The second module extracts deeper features from the regions of interest to infer the exact target class and location. Since some small areas may be missed in the first stage, it is important to detect plausible regions in different scales.

In particular, for the first module, called Region Proposal Network (RPN), it can be implemented with a fully convolutional network with the input image. Then we can have the feature map generated by the network. A small network is then applied on each grid of this feature map and produces the rectangular proposals, each with a score. As shown in Fig. 3, the target of RPN contains two parts: the coordinates of each rectangle, and the proposal score of the rectangle (higher score means more likely to be a region of interest). In particular, the coordinates are represented with reference to the anchor boxes of different scales and aspect ratios. That means the feature of a grid (fixed size) inferences areas of different sizes. In other words, the output proposal can be generated from the features of itself or additionally with surrounding context. By setting the anchor boxes’ different scales and aspect ratios, we can exploit different areas’ contextual information to locate areas with different scales. To this end, we obtain the regions of interests and the corresponding feature from the RPN.

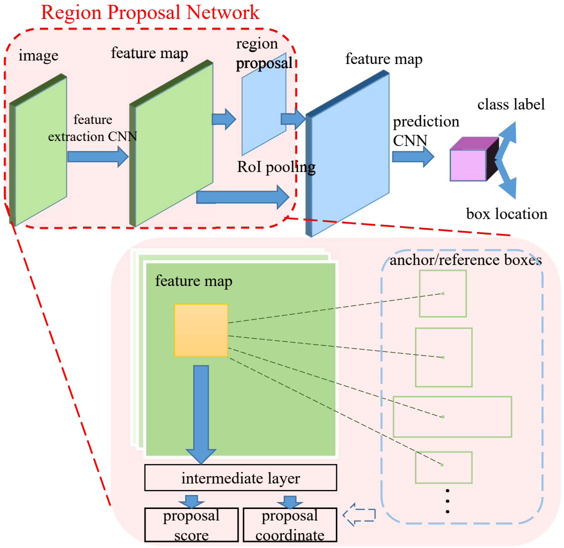


Fig. 3. Illustration of faster R-CNN [19]. With different scales and aspect ratios of the anchor/refrence box, we can detect functional area with different scales in a forward pass of the network.

For the second module, it contains another deep network with several convolutional layers for deeper feature extraction. For each region, it takes the features from the RPN as input, uses the convolutional layers to extract new features and performs rectangle classification and location regression refinement. These two modules can be trained in an end-to-end manner. After these two modules, we can obtain one or more rectangles for each class (i.e., functional category). The rectangles for each class may highly overlap with each other. To reduce redundancy, a non-maximum suppression (NMS) [16] post-process is performed based on the class score and the output is the desired result.

To this end, we have the RPN for proposal generation of different scales, and we can use the faster R-CNN as the multi-scale method for functionality area detection. However, without object annotation, this method cannot exploit the surrounding object knowledge, especially for the second module. To take advantages of the object knowledge, one can use existing object detection datasets to train the network first and then fine-tune on the functional area dataset. In our experiments, we also find that this is a strong baseline. But with limited samples for the functional area data, this method will have the risk of overfitting and the object knowledge in the network may be vanished in the fine-tuning process. To solve this problem, we proposed to use both of the object-related feature and functionality-related feature in a two-stream network as will be described in Sec. III-C.

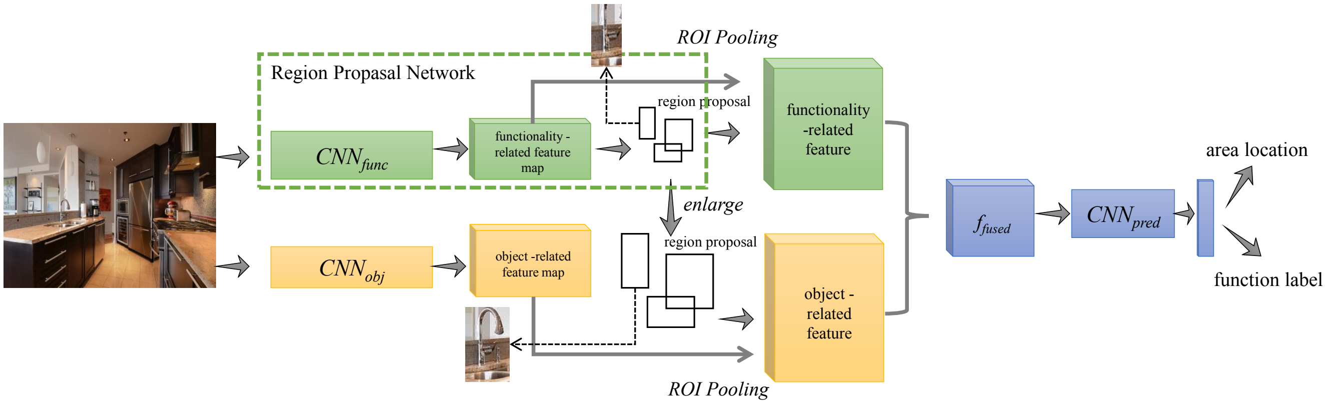


Fig. 4. An illustration of the proposed two-stream network. The upper stream (the green part) extracts the functionality-related feature and the lower stream (the yellow part) extracts the object-related feature. Note that the object-related feature is extracted from the enlarged region of the area proposal. In this case, we extract and incorporate the surrounding object information for final area inference.

C. *Two-stream Network for Object Context Fusion*

Figure 4 illustrates the proposed two-stream network. In general, the input is an image I that is fed into two networks, namely, functionality CNN (CNN_{func}) and object CNN (CNN_{obj}), to extract the functionality-related and object-related feature maps, respectively. With the functionality-related feature maps, we generate the area proposals of a collection of bounding boxes, as the RPN described in Sec. III-B. Then we can use these proposals to extract the corresponding features from the feature maps and get a fixed-sized feature by ROI pooling as in [19]. On the other hand, we enlarge proposals by a fixed scale (i.e., 0.5), and use these larger proposals to extract the surrounding object context features, which can also be obtained by ROI pooling. Here we design CNN_{obj} and CNN_{func} with same network structure (but different parameter values that are learned as will be described in Sec. III-D). To this end, we can fuse the functionality-related features f_{func} and object-related features f_{obj} by

$$f_{fused} = \lambda f_{func} + (1 - \lambda) f_{obj}, \tag{1}$$

where λ weights the importance of the two features. Here we set $\lambda = 0.5$ in our implementation. The fused feature f_{fused} is then fed into another network CNN_{pred} to generate the feature $CNN_{pred}(f_{fused})$. With $CNN_{pred}(f_{fused})$, we predict the final area location and functionality label by regression. Note that in this work we just fuse the features by a fixed liner combination to verify usefulness of object-realted feature. A learnable combination way can be further explored to improve the performance (for example, we learn λ in the training process).

D. *Training with Object Knowledge Transfer*

To leverage the existing object detection dataset and transfer the object knowledge, the training process for the two-stream network contains two stages. For the first stage, we need to initialize the network parameters. Here we first follow the training scheme of faster R-CNN [19] to train an object detection network as in Fig. 3, using an existing object detection dataset. As described in [19], the training process can be conducted end-to-end by back propagation and stochastic gradient descent [2]. Then the RPN feature extraction CNN and its parameter is used as CNN_{func} and CNN_{obj} of the two-stream network. And the prediction network in Fig. 3 and its parameter is used as the CNN_{pred} of the two-stream network.

For the second stage, to reserve the learned object knowledge, we fix the parameters of CNN_{obj}, and fine-tune the rest part of the network using the functional area dataset. Similar to the training of faster R-CNN, there are two loss functions for the two-stream network. The first one is the loss of RPN, which is a combination of binary classification loss (whether the proposal is a functional area) and regression loss (the Euclidean distance between the ground truth location and predicted location). The second one is the loss of the final results, which is the same as the RPN loss except that the classification loss is not binary but multiple classes for the functionality type instead. We can see that the two loss functions are differentiable, so as for the feature combination Equation (i.e., Eqn.(1)). Also, the ROI pooling operation can be differentiable w.r.t. the proposal coordinates as described in [3]. To this end, we can employ stochastic gradient descent to train this model directly.

IV. RESULTS

A. *Implementation Details*

We implement our network with the Tensorflow [1] machine learning toolbox. For the network structure, we employ the ResNet-101 as in [7] for its state-of-the-art performance on image recognition. In particular, both CNN_{func} and CNN_{obj} contain the first 3 ResBlocks in [7] (i.e., conv1, conv2_x, conv3_x, and conv4_x, totally 91 convolutional layers of ResNet-101). CNN_{pred} contains the final block (i.e., conv5_x layers) of ResNet-101. The final result is produce by two sibling fully connected layers. When we fine-tune the two-stream network, the learning rate is set 0.0003 and the batch size is 1. The scales of the anchor box are set as 0.125, 0.2, 0.5, 1, 2 to capture areas in different scales, and

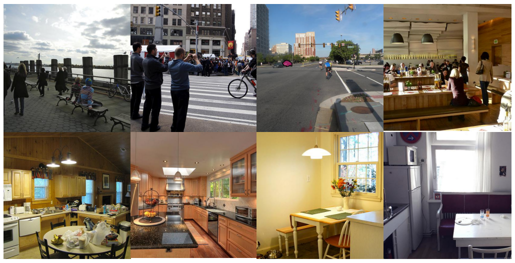


Fig. 5. Example images of the COCO dataset [12].

the aspect ratios are 0.5, 1, 2. The Intersection over Union (IoU) value for NMS process is set 0.7.

B. Dataset and Evaluation Metrics

We perform experiments on the functional area dataset proposed in [24]. In particular, this dataset contains around 600 kitchen images from the SUN dataset [23]. Some examples are depicted in Fig. 7. For each image, different types of functional area rectangles are annotated by human. In total, there are around 10,000 annotated area samples. The statistical distribution of these samples are shown in Fig. 6. To evaluate the proposed method, we follow the same experiment protocol published in [19]. The training set contains 90% images of the whole set and the rest are used for testing.

For the object network, we use the COCO dataset [12] to learn the object-related feature as described in Sec. III-D. In particular, COCO contains 91 common object categories, such as person, car, desk, bottle and bowl. The images are collected from the Internet, with various scenarios, such as kitchen, street and park. Each image is labeled with the rectangles of the included objects. Some examples are shown in Fig. 5. The training set of this dataset contains around 80k images.

For evaluation metrics, we use the precision, recall and F1 score as in [24]. A correct prediction means: 1) the predicted functional type is correct; 2) the Intersection over Union (IOU) of the predicted rectangle and ground-truth rectangle is bigger than 0.5. The precision, recall and F1-score are calculated as follows: $precision = \frac{\#truepositive}{\#truepositive + \#falsepositive}$, $recall = \frac{\#truepositive}{\#truepositive + \#falsenegative}$, and $F1 = 2 \times \frac{precision \times recall}{precision + recall}$, where $\#truepositive$, $\#falsenegative$ denote the number of true positive and false negative samples, respectively.

C. Baseline Methods

To demonstrate the effectiveness of the proposed algorithm, we employ the following baseline methods:

- Selective search + CNN classification method in [24]. This method first uses selective search [21] that generates the candidate regions. After that, a CNN takes each region as input and output the predicted functional category. The CNN is first pre-trained on the ImageNet classification dataset [5] and then fine tuned on the training data. We report the result of this method in the original paper [24]. In particular, since [24] employs a 'hard sample mining' technique to refine the CNN model, the precision and recall is different in different rounds of network refinement, we report result of all the three rounds performed in [24].
- Faster R-CNN method [19]. As described in Sec. III-B, faster R-CNN is a building block of our method. We treat it as a baseline for the experiments. In particular, we first pre-train the faster R-CNN with the COCO dataset, and then fine tune on the training data as described in Sec. III-D. For a fair comparison, we also use the ResNet-101 [7] network for feature extraction and the prediction network for the final result is also the same as the proposed approach. Other training parameters are the same as [7] to achieve reasonable performance. The main difference between faster R-CNN and our method is that 1) we reserve and employ the object-related feature for the functional area inference, and 2) we use more anchor boxes' scale and aspect ratio variations to capture areas in different scales.

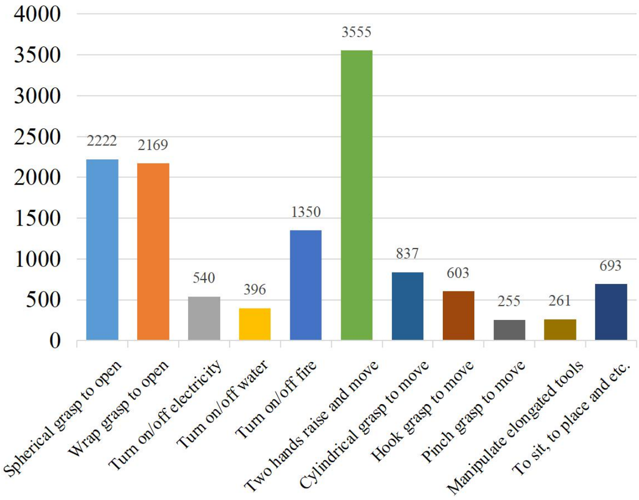


Fig. 6. Histogram for the numbers of different functional area annotated in the dataset of [24].

D. Quantitative Comparison to Baseline Methods

Table I shows the precision, recall and F1 score of the proposed method and other baselines. We can observe that faster R-CNN can achieve better performance then the two-stage method of [24]. This is because faster R-CNN is formulated in an end-to-end manner by joint feature learning and area detection. The multi-scale anchor boxes can handle areas in different sizes. It also shows that the proposed method produce superior performance compared to existing methods. Since the proposed method is built on faster R-CNN, the result demonstrates that the additionally proposed object context fusion, and more anchor boxes' variation is effective.

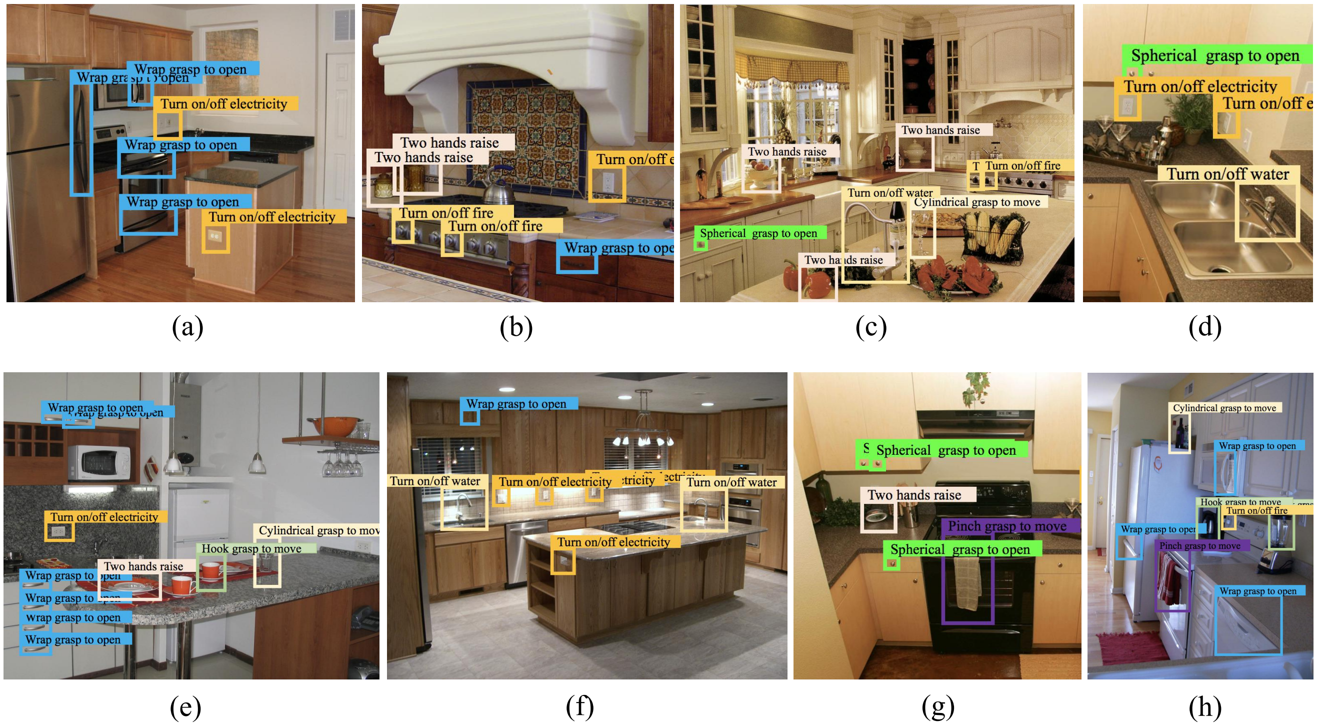


Fig. 7. Example results of the proposed method in the dataset of [24].

TABLE I
PRECISION, RECALL AND F1 SCORE OF DIFFERENT METHODS.

Method	Precision	Recall	F1 score
One round of [24]	5.26%	23.92%	0.0862
Two rounds of [24]	16.18%	15.05%	0.1559
Three rounds of [24]	31.52%	11.58%	0.1694
Faster R-CNN in [19]	38.04%	24.57%	0.2985
Our proposed method	50.12%	29.16%	0.3729

E. Ablative Analysis

To further evaluate the contribution of different components of the proposed method, we perform experiments with different options and settings in our model. The results are presented in Table II. For the Model A in Table II, we remove the two-stream structure and only use one stream (i.e., the faster R-CNN). And we also set the anchor boxes' scale and ratio as the original ones in [19]. This result in the original faster R-CNN. In Model B, we keep the same anchor box parameters and employ the two-stream structure to fuse the surrounding object-related feature. We can see that this improve the precision substantially, from 38.04% to 52.29%. This demonstrates the effectiveness of the object context. Then we evaluate the contribution of the anchor box's variations by using larger range of scale and aspect ratio (i.e., Model C and Model D). This also boosts the performance. Again, the superior performance of using object context is demonstrated as Model D is better than Model C.

F. Qualitative Analysis

To visualize the performance of the proposed method, some example results are presented in Fig. 7. It shows that the result covers areas in different sizes. Even some small drawer handle can be detected as 'Wrap grasp to open'. However, we also observe many missed areas, such as the gas stove switches in Fig. 7 (b). This is caused by 1) the details of the image may be missed due to the subsample layers in the feature extraction CNN; 2) the NMS post-process that merge neighboring areas with overlap. We can investigate other multi-scale detection framework such as [13] to reserve these details in future research.

To further study how to improve the performance of the current method, we visualize the confusion matrix of the predicted area and ground truth label of these areas, as shown in Fig. 8. Since this is not a classification problem, to obtain the ground truth label of the predicted area, we find the annotated area with the largest overlap with the predicted area. If the IoU is bigger than 0.5, then the ground truth is set the label of that annotated area. Otherwise, the ground truth label is 'Background'. From Fig. 8, we observe that most of the error cases are that the 'Background' is classified as certain functional area. To further study this problem, we investigate the result and find some false positive cases. But we also find some areas that are actually correct but the annotations are missing. Some of these cases are shown in Fig. 9. We can see that some of the areas are in different viewpoints or in a crowded scene. This indeed imposes some challenges for the annotator.

The generalization ability of the proposed method also

TABLE II

PRECISION, RECALL AND F1 SCORE OF OUR METHOD WITH DIFFERENT PARAMETER SETTINGS.

Model ID	Anchor Box Scales	Anchor Box Aspect Ratios	with Object Context	Precision	Recall	F1 score
A	[0.5 1.0 2.0]	[0.5 1.0 2.0]	No	38.04%	24.57%	0.2985
B	[0.5 1.0 2.0]	[0.5 1.0 2.0]	Yes	52.29%	24.78%	0.3363
C	[0.125 0.25 0.5 1.0 2.0]	[0.25 0.5 1.0 1.5 2.0]	No	50.18%	28.17%	0.3608
D	[0.125 0.25 0.5 1.0 2.0]	[0.25 0.5 1.0 1.5 2.0]	Yes	50.12%	29.16%	0.3729

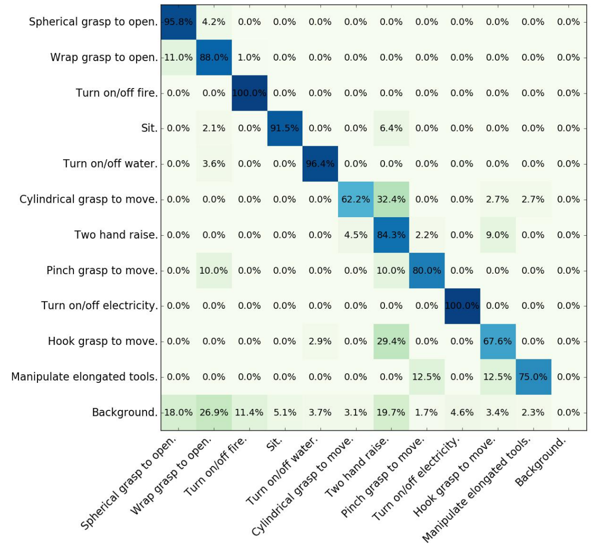


Fig. 8. Confusion matrix of the result produced by the proposed method. The vertical axis is the ground truth label and the horizontal axis is the predicted label.

worths investigation. Since the training and testing data is in the kitchen environment, we download some other indoor images (but not in a kitchen) from the web and test our method on these images. The results are shown in Fig. 10. Although there are some errors, we can see that some meaningful novel areas are detected. For example, the pillow is not presented in the kitchen training dataset, but the algorithm can still assign the 'pinch grasp to move' label. Similarly, it learns that a bed can be assigned to 'sit'. This demonstrates that the model has the ability to be extended to some new areas.

V. CONCLUSION

In this work, we investigate how to incorporate the surrounding object context to boost the functional area detection. We formulate a two-stream deep network structure that extract and fuse the functionality-related and object-related feature for functional area inference. The problem of handling multi-scale areas are also discussed. We compare the proposed method with existing functional area detection approach [24] and deep-learning-based object detection approach [19] and our superior performance is demonstrated. We also evaluate the effectiveness of the proposed object-related feature fusion. Interesting, we also find that our method can handle some areas with different objects that are not seen in the training data. This motivates us that we can study how to extend the algorithm in the future. It is also worth pointing out we employ faster R-CNN [18] in our framework because of its state-of-the-art performance on generic object detection. Other detection framework can also be similarly applied in the future. The important contribution of this work is the usage of object contextual information for functional area detection.

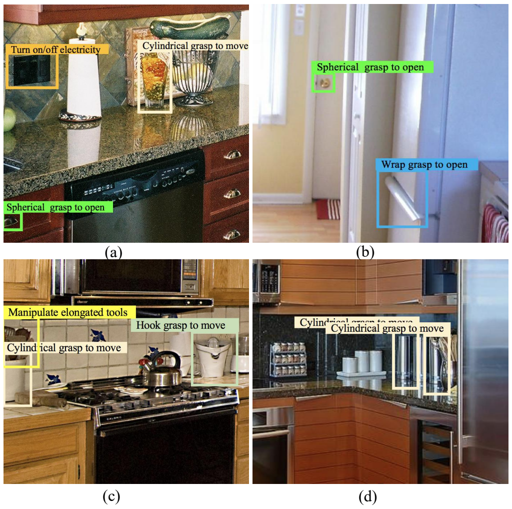

Turn on/off electricity
Cylindrical grasp to move
Spherical grasp to open
(a)
Spherical grasp to open
Wrap grasp to open
(b)
Manipulate elongated tools
Hook grasp to move
Cylind ical grasp to move
(c)
Cylindrical grasp to move
Cylindrical grasp to move
(d)

Fig. 9. Some target areas with missing annotation but correctly detected by our method.

ACKNOWLEDGMENT

This work is supported by the NSFC-Shenzhen Robotics Projects (U1613211), major program of Science and Technology Planning Project of Guangdong Province (2017B010116003), Guangdong Natural Science Foundation (1614050001452), and the Fundamental Research Funds for the Central Universities.

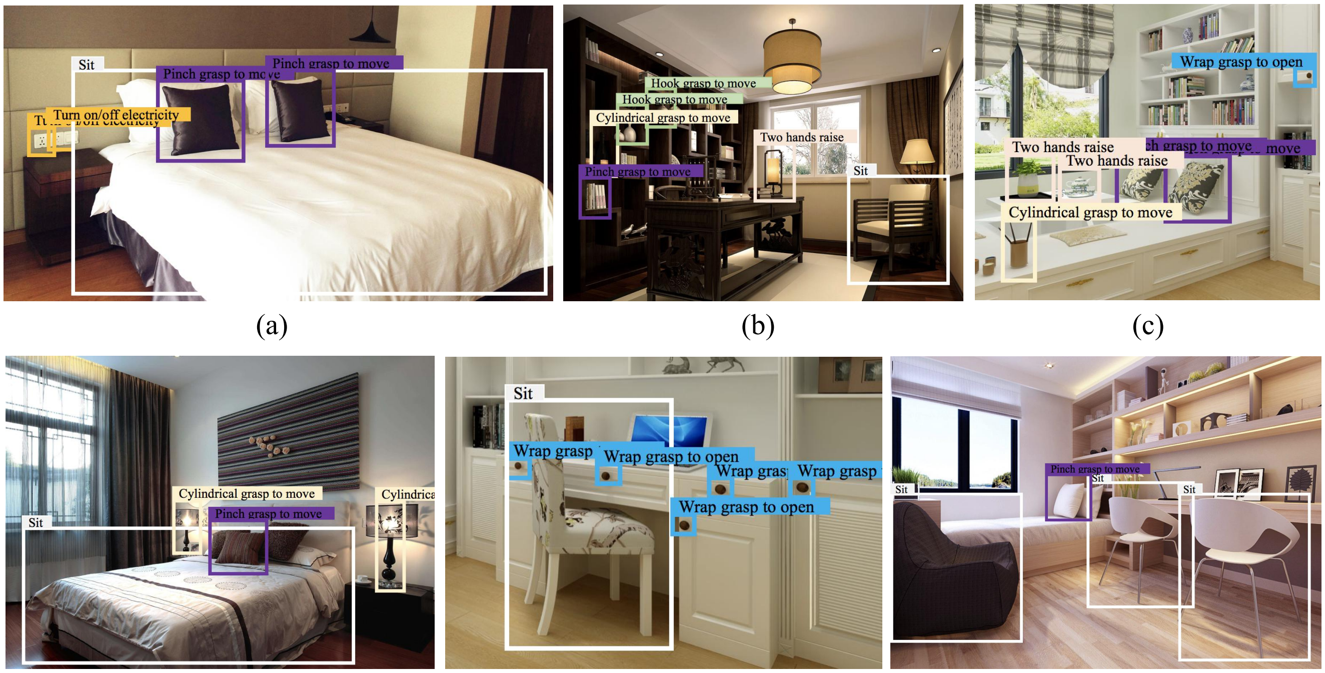
REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT*, pages 177–186. 2010.

[3] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158, 2016.

[4] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.



(a) (b) (c)

(d) (e) (f)

Fig. 10. Example results of the proposed method in images from the Internet. We can find that the algorithm can detect some areas that are not seen in the training kitchen dataset [24].

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, 2009.

[6] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[8] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[10] Sulabh Kumra and Christopher Kanan. Robotic grasp detection using deep convolutional neural networks. *arXiv preprint arXiv:1611.08036*, 2016.

[11] Yining Li, Chen Huang, Chen Change Loy, and Xiaoou Tang. Human attribute recognition by deep hierarchical contexts. In *Proceedings of European Conference on Computer Vision*, pages 684–700, 2016.

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of European Conference on Computer Vision*, pages 740–755, 2014.

[13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proceedings of European Conference on Computer Vision*, pages 21–37, 2016.

[14] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

[15] Austin Myers, Ching L Teo, Cornelia Fermüller, and Yiannis Aloimonos. Affordance detection of tool parts from geometric features. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1374–1381, 2015.

[16] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *Proceedings of International Conference on Pattern Recognition*, volume 3, pages 850–855, 2006.

[17] Alessandro Pieropan, Carl Henrik Ek, and Hedvig Kjellström. Functional object descriptors for human activity modeling. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1282–1289, 2013.

[18] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[20] Yuyin Sun, Liefeng Bo, and Dieter Fox. Attribute based object identification. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2096–2103, 2013.

[21] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[22] Florentin Wörgötter, Eren Erdal Aksoy, Norbert Krüger, Justus Piater, Ales Ude, and Minija Tamosiunaite. A simple ontology of manipulation actions based on hand-object relations. *IEEE Transactions on Autonomous Mental Development*, 5(2):117–134, 2013.

[23] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010.

[24] Chengxi Ye, Yezhou Yang, Ren Mao, Cornelia Fermüller, and Yiannis Aloimonos. What can i do around here? deep functional scene understanding for cognitive robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 4604–4611, 2017.

[25] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. Memory-augmented attribute manipulation networks for interactive fashion search. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.