Object Recognition for Service Robots Based on Faster R-CNN (Postprint)

Authors: Shi Jie, Zhou Yali, Zhang Qizhi

Date: 2018-08-13T00:00:00+00:00

Abstract

As robots are increasingly deployed in the service industry, particularly playing important roles in household services, the demand for information acquisition and object recognition capabilities of service robots has grown significantly. Traditional daily commodity recognition pipelines typically employ classical image recognition and machine learning algorithms, such as Support Vector Machine (SVM), Random Forest, or AdaBoost, utilizing basic features like gradient, texture, or color of target images for recognition. While these methods can be applied in relatively simple backgrounds, they struggle to achieve outstanding performance in complex background environments and are difficult to attain high accuracy. Currently, Convolutional Neural Networks (CNN) demonstrate superior performance in object recognition and have become the preferred choice in many object recognition scenarios. Considering the hardware configuration cost of service robots, the fast algorithm of Region-based Convolutional Neural Network (R-CNN), namely Faster R-CNN, is introduced into the system for item recognition using CPU-based computation. The CNN network is utilized to extract image features, followed by a region proposal layer. Experimental results demonstrate that applying deep learning-based recognition methods to service robot platforms is feasible, yielding accurate recognition performance and achieving favorable detection results in experiments.

Full Text

Item Recognition Based on Faster R-CNN in Service Robots

Shi Jie, Zhou Yali, Zhang Qizhi
(School of Automation, Beijing Information Science & Technology University, Beijing 100192, China)

Abstract

As robots become increasingly prevalent in the service industry, particularly for domestic applications, the demand for robust information collection and target recognition capabilities in service robots has grown substantially. Traditional commodity recognition pipelines typically employ classical image recognition and machine learning algorithms such as Support Vector Machines (SVM), Random Forest, or Adaboost, leveraging basic features like gradients, textures, or colors. While these methods perform adequately in simple backgrounds, they struggle to achieve high accuracy in complex environments. Convolutional Neural Networks (CNN) currently represent the state-of-the-art in target recognition and have become the preferred choice for many applications. Considering the hardware cost constraints of service robots, this paper introduces Faster R-CNN—a fast algorithm based on Region-based Convolutional Neural Networks (R-CNN)—into our system for item recognition using CPU computation. The CNN extracts image features, followed by a region proposal layer. Experimental results demonstrate that applying deep learning recognition methods to service robot platforms is feasible, yielding accurate recognition and effective detection performance.

Keywords: service robot; deep learning; Faster R-CNN; commodity recognition

---

0 Introduction

Platform-based item recognition represents a crucial trend for future industry development. On one hand, service robots possess sophisticated hardware foundations capable of completing fundamental tasks; on the other hand, deep learning not only enables multi-target recognition but also achieves high accuracy. Object detection and recognition, which depend on computer vision advancements, have long been a key research focus in image engineering. However, due to technological limitations, low public awareness of object recognition, and restricted algorithmic application scenarios, rapid development in object detection only began in the 1990s. While human vision can easily locate and identify target objects through texture, color, and depth information, computers process images as numerical RGB matrices, making it difficult to abstract concepts like shelves or items directly. Additional challenges—including complex backgrounds, diverse object poses, and varying illumination—further complicate detection. Compared to traditional item recognition, images captured by service robots present even greater difficulties: more complex backgrounds, inconsistent lighting, varying target distances, and shape variations of identical items.

Traditional methods primarily employ basic image processing techniques such as background modeling [**?**], HOG (Histogram of Oriented Gradients) [**?**], K-means clustering [**?**], feature point matching [**?**], and SURF (Speeded Up Robust Features) [**?**]. Among these, the Viola-Jones framework [**?**], proposed in 2001,

gained widespread attention for its speed, simplicity, and low computational cost, enabling real-time face detection in cameras using Haar features [**?**] in cascaded multi-scale sliding windows that quickly discard false classifications.

Deep learning, as an extension of machine learning, has become particularly prominent in computer vision. Similar to its superiority in image classification, deep learning now represents the best approach for target detection. In recent years, significant progress has been made: NYU researchers proposed Overfeat [**?**] in 2013, introducing a multi-scale sliding window algorithm using convolutional networks. Shortly after, Girshick et al. from UC Berkeley introduced the Region-based Convolutional Neural Network (R-CNN) algorithm [**?**], a major breakthrough that improved detection performance by 50% over traditional methods. R-CNN employs a three-stage approach: (a) region proposal extraction using methods like selective search [**?**]; (b) CNN-based feature extraction from regions; and (c) SVM classification [**?**]. Despite its success, R-CNN suffered from training inefficiencies.

In 2015, Girshick published Fast R-CNN [**?**], which evolved into a pure deep learning method. Like R-CNN, it uses selective search for region proposals, but extracts features from the entire image using CNN, then applies Region of Interest (ROI) pooling, followed by backpropagation for classification and bounding box regression. This approach is faster and enables end-to-end differentiation, though it still relies on selective search as a bottleneck. Subsequently, Ren et al. introduced Faster R-CNN [**?**], adding a Region Proposal Network (RPN) [**?**] to eliminate selective search and enable fully end-to-end training. The RPN outputs potential targets based on "objectness" scores, which are then processed by ROI pooling and fully connected layers for classification.

With deep learning' s explosive growth and widespread robot adoption in services, this paper applies deep learning to an intelligent service robot platform. Considering both accuracy and speed, we utilize Faster R-CNN as our target detection algorithm on a self-developed service robot with path planning, pedestrian tracking, object grasping, and autonomous navigation capabilities. This work establishes the foundation for future autonomous grasping and navigation. Faster R-CNN offers two training modes: the 2015 NIPS "alternating optimization" (alt-opt) method [**?**], which iteratively trains RPN and Fast R-CNN in alternating fashion; and "End-to-End" training, which fuses RPN and Fast R-CNN into a single network. The latter is recommended for its lower memory usage, faster training, and slightly higher accuracy. This paper applies deep learning-based Faster R-CNN to a domestic service robot platform, demonstrating feasibility and effectiveness through competitions and experiments, showing superior performance over traditional methods.

1 Item Recognition Algorithm

1.1 Fast R-CNN Algorithm

While R-CNN uses selective search to extract potential bounding boxes, it suffers from severe speed bottlenecks due to redundant feature computation across all regions. Fast R-CNN addresses three key problems:

a) **Slow testing**: Fast R-CNN normalizes and feeds the entire image into CNN, sharing computations before the final convolutional layer by incorporating proposal box information into the feature map.

b) **Slow training**: During training, Fast R-CNN processes one image at a time, extracting CNN features and region proposals simultaneously. Training data flows directly to the loss layer in GPU memory, eliminating duplicate computation for early layers and removing the need for large disk storage.

c) **Large training space requirements**: Fast R-CNN unifies category classification and location regression within a single deep network, eliminating extra storage needs.

The Fast R-CNN model adopts a CNN architecture. [Figure 1: see original paper] shows the traditional CNN structure, where pooling layers are alternately inserted between convolutional layers. Features extracted through convolutional layers are filtered and combined to form new feature maps—more abstract representations of the original image—finally normalized into one-dimensional arrays for the fully connected layers that perform classification and detection.

1.2 Faster R-CNN Algorithm

Faster R-CNN' s key innovation is the RPN, which predicts proposals directly, generating fewer and faster predictions than selective search, with most operations performed on GPU. The convolutional networks are shared between RPN and Fast R-CNN, critically improving detection speed. Two key distinctions from other detection networks are: (a) using RPN instead of selective search for proposal generation, and (b) sharing convolutional networks between the proposal and detection networks.

The overall Faster R-CNN framework: - Processes the entire image through CNN for feature extraction - Generates region proposals (300 per image) via RPN - Maps proposals to the final convolutional layer to generate ROI features - Jointly trains classification probabilities and bounding box regression using Softmax Loss and Smooth L1 Loss

[Figure 2: see original paper] illustrates the Faster R-CNN network architecture.

1.2.1 Region Proposal Network (RPN) To detect objects at multiple scales, two main approaches exist: (a) cropping inputs or (b) applying sliding

windows of different sizes to feature maps. RPN adopts a different strategy. Taking any-sized image as input, it outputs a set of scored candidate boxes.

RPN' s core idea uses a convolutional network to directly generate proposals through sliding windows, requiring only a single pass over the final convolutional layer. At each sliding window position, it simultaneously predicts k region proposals, yielding $4k$ outputs for box coordinates and $2k$ scores. The training data generation process checks whether an anchor covers the ground-truth area by more than 75%; if so, the anchor is labeled as "object present." Otherwise, the anchor with maximum coverage is selected. This estimates the probability of each proposal being object or non-object.

The anchor mechanism and bounding box regression enable multi-scale, multi-aspect ratio detection. RPN is a fully-convolutional network (FCN) [**?**] that can be trained end-to-end for proposal generation, predicting object boundaries and scores by adding two convolutional layers (cls and reg) to the base CNN. To handle size variations, Faster R-CNN employs three aspect ratios (1:1, 2:1, 1:2) and three scales (128, 256, 512), creating nine types of anchor boxes. These nine windows form a 256-dimensional vector on the convolutional feature map (512-dimensional for VGG models), from which the top 300 scoring windows are selected as final proposals. [Figure 3: see original paper] shows the RPN network structure.

The RPN objective function combines classification and regression losses. Following [**?**], classification uses cross-entropy while regression uses robust Smooth L1:

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

where i is the anchor index in a mini-batch, p_i is the predicted probability of anchor i being an object, and p_i^* is the ground-truth label (1 for positive, 0 for negative). t_i represents the 4-parameter coordinate vector of the predicted bounding box, and t_i^* is the ground-truth vector associated with the anchor.

The Smooth L1 loss is defined as:

$$\text{Smooth}_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

The classification loss in RPN is:

$$L_{cls}(p_i, p_i^*) = -\log[p_i^* p_i + (1 - p_i^*)(1 - p_i)]$$

For bounding box regression, we use parameterizations for the four coordinates:

$$t_x = (x - x_a)/w_a, \quad t_y = (y - y_a)/h_a$$
$$t_w = \log(w/w_a), \quad t_h = \log(h/h_a)$$
$$t_x^* = (x^* - x_a)/w_a, \quad t_y^* = (y^* - y_a)/h_a$$
$$t_w^* = \log(w^*/w_a), \quad t_h^* = \log(h^*/h_a)$$

where x, y, w, h denote the box's center coordinates, width, and height. Variables x, x_a, x^* respectively represent the predicted box, anchor box, and ground-truth box (similarly for y, w, h). This can be viewed as bounding box regression from an anchor to a nearby ground-truth box.

1.2.2 Training RPN Network RPN networks are typically trained end-to-end via backpropagation and stochastic gradient descent (SGD). We follow an "image-centric" sampling strategy where each mini-batch contains a single image with many positive and negative anchors. While optimizing over all anchors is possible, results would be biased toward negative samples due to their dominance. Instead, we randomly sample 256 anchors per image to compute the mini-batch loss, maintaining a positive-to-negative ratio up to 1:1. If fewer than 128 positive samples exist, negative samples fill the remainder.

---

2 Experimental Platform and Results

2.1 Experimental Platform

[Figure 4: see original paper] shows the Sun@Home robot from Beijing Information Science & Technology University' s domestic service robot team. The platform consists of a Kinect II sensor [**?**], adjustable lifting mechanism, 3-DOF manipulator, omnidirectional wheel chassis, and a 270° range laser radar. The Kinect II offers higher resolution and color fidelity than its predecessor, with a color camera resolution of 1920×1080, depth sensor resolution of 512×424, 30 fps frame rate, and 0.5–4.5 m detection range. These high-definition images enhance algorithmic precision for object localization and recognition.

2.2 Dataset

Since the official Faster R-CNN uses the 1000-class ImageNet dataset for high accuracy, we fine-tune the pre-trained model on a custom VOC dataset, achieving stable parameters without collecting millions of images.

1) Data collection and preparation: Using the robot' s Kinect sensor, we recorded 10 target object categories, decomposing them into 2,010 JPG images with corresponding 2,010 XML label files. The training and validation sets contain 90% (1,809 images), while the test set contains 10% (201 images). XML

files and images are organized into Annotation, ImageSets/Main, and JPEGImages directories following the VOC2007 format. [Figure 5: see original paper] shows the actual data collection scenario.

Object categories: Cola, milk, yogurt, toothpaste, coffee, green tea, body wash, shampoo, water, and soap.

2) Training platform: Considering the service robot lacks a high-performance GPU, training was performed on a server with Intel Core i7-6700K CPU (4.00 GHz×8), GeForce GTX 1080, Ubuntu 14.04 64-bit, and 256 GB SSD.

2.3 Feature Map Extraction

Faster R-CNN' s core feature extraction relies on CNNs that learn object color, shape, and texture features while also capturing background context. Caffe' s visualization method reveals key target information. [Figure 6: see original paper] shows feature maps from various convolutional layers: conv1 and conv2 extract low-level features like color and edges; conv3 captures texture features; conv4 and conv5 extract more critical high-level features. [Figure 7: see original paper] shows how pooling layers refine feature extraction, making key object information more prominent.

2.4 Experimental Results and Analysis

Definitions: False detection occurs when an object is detected but incorrectly recognized; missed detection occurs when an object is not detected at all.

1) Comparison Between Faster R-CNN and Fast R-CNN We implemented Faster R-CNN and compared it with Fast R-CNN. Training the additional RPN network significantly impacts accuracy. For RPN training, we fine-tune a pre-trained model, using RPN-generated proposals to train Fast R-CNN, then fix Fast R-CNN' s convolutional layers to fine-tune RPN, iterating until convergence.

Multi-target test: [Figure 8: see original paper] shows the test scene containing all dataset objects. Expected result: correctly localize and recognize all objects. [Figure 9: see original paper] shows Fast R-CNN results, while [Figure 10: see original paper] shows Faster R-CNN results. Fast R-CNN correctly identifies coffee, green tea, body wash, milk, and cola, but misidentifies yogurt as milk and misses soap—likely due to yogurt' s similarity to milk and soap' s small size with background-matching color. Faster R-CNN correctly localizes and recognizes all objects, demonstrating superior performance.

False/missed detection test: We test with milk tea, paper cups, soap, and paper rolls—only soap belongs to the dataset. Expected result: recognize only soap. [Figure 11: see original paper] shows Fast R-CNN results, while [Figure 12: see original paper] shows Faster R-CNN results. Fast R-CNN correctly identifies soap but also misidentifies the paper roll as soap with only 74.2% confidence, as

both share similar color and texture features. Faster R-CNN correctly identifies soap without false detections. Neither algorithm falsely detects milk tea or paper cups.

Environmental adaptability test: We test with shampoo, facial cleanser, and bottled coffee—similar in outline but only shampoo and coffee are in the dataset. Expected result: recognize shampoo and coffee only. [Figure 13: see original paper] shows Fast R-CNN results, while [Figure 14: see original paper] shows Faster R-CNN results. Both correctly identify some dataset objects but misidentify canned coffee as cola, likely because the dataset lacks coffee samples at that specific angle while containing multi-angle cola samples with similar color features under the same lighting. This indicates that robot camera angles should closely match training data angles, and data augmentation through scale transformation is crucial.

Comprehensive results from multiple tests are summarized in .

TABLE:1 Comparison of Fast R-CNN and Faster R-CNN Parameters

Method	mAP	Training Time	Test Time	False Detection Rate	Missed Detection Rate
Fast R-CNN	75%	~10h	~2.1s	20%	5%
Faster R-CNN	90%	~14h	~1.5s	6%	4%

Analysis shows Fast R-CNN achieves ~75% accuracy with ~10h training time and ~2.1s test time, while Faster R-CNN achieves ~90% accuracy with ~14h training and ~1.5s test time. Though neither achieves real-time performance, Faster R-CNN's test time is significantly lower because its convolutional network generates proposals (reduced from ~2,000 to 300) and shares computations with the detection network, improving proposal quality.

2) Comparison Between End-to-End and alt-opt Training Methods Faster R-CNN provides two training algorithms, both outperforming Fast R-CNN.

Multi-target test: Under normal shooting angles with similar-shaped/colored objects including non-dataset items, alt-opt and End-to-End achieve comparable accuracy on standard-angle images, correctly identifying dataset objects without detecting non-dataset items.

Abnormal angle test: Using shampoo, facial cleanser, and bottled coffee, [Figure 15: see original paper] shows alt-opt results while [Figure 16: see original paper] shows End-to-End results. alt-opt correctly identifies shampoo and

bottled coffee but falsely detects facial cleanser as coffee, likely due to their similar dark colors and contours. End-to-End correctly detects shampoo and bottled coffee without false positives and is slightly faster, as alt-opt' s four-stage training generates more weight values. End-to-End is more suitable for our platform.

Comprehensive results are shown in .

TABLE:2 Comparison of alt-opt and End-to-End Parameters

Method	mAP	Training Time	Test Time	False Detection Rate	Missed Detection Rate
alt-opt	89%	~14h	~1.6s	6%	5%
End-to-End	90%	~11h	~1.5s	5%	4%

Analysis: In specific scenarios, Faster R-CNN outperforms Fast R-CNN. For our platform, End-to-End training is slightly superior to alt-opt. In practice, Faster R-CNN recognizes more objects, and increasing dataset diversity improves results. Multiple convolutional layers extract richer features, ensuring recognition accuracy.

3 Conclusion

Traditional item recognition methods suffer from low efficiency, slow detection, and high false/missed detection rates. Deep learning automates feature extraction through neural networks, using forward/backpropagation to adjust parameters and avoid single-algorithm limitations. This paper employs the effective Faster R-CNN algorithm on a service robot platform, achieving 90% recognition accuracy in domestic service robot competitions. Considering cost and space constraints, our experiments run on CPU without GPU, achieving ~1.5s detection speed with End-to-End training and ~1.6s with alt-opt, both at ~90% accuracy—significantly outperforming traditional methods. However, the method has lighting requirements; overly dark conditions impair recognition, and small objects remain challenging. The Sun@Home recognition team will continue development, eventually incorporating a suitable GPU based on hardware constraints, as real-time performance is essential for industrial robotics.

References

[1] Song Huanhuan. Research and implementation of background modeling

method under complex scene [D]. Nanchang: Nanchang University, 2015.

[2] Taigman Y, Yang Ming, Ranzato M A, et al. Deepface: closing the gap to human-level performance in face verification [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. [S. l.]: IEEE Press, 2014: 1701-1708.

[3] Pan Wei, Zuo Xin, Shen Gouqiang, et al. The design and implementation of item identification system [J]. Science & Technology Vision, 2015 (5): 167.

[4] Lin Ting. Research on feature points matching algorithm sequence image [J]. Modern Computer, 2016 (10): 28-32.

[5] Hu Xiuxiang. Object recongnition and localization based on RGB-D data [D]. Tianjin: Civil Aviation University of China, 2016.

[6] Viola P, Jones M J. Robust real-time face detection [J]. International Journal of Computer Vision, 2004, 57 (2): 137-154.

[7] Shi Dongcheng, Ni Kang. Motion gesture tracking based on compressed sensing W-HOG features [J]. CAAI Trans on Intelligent Systems, 2016, 11 (1): 124-128.

[8] Sermanet P, Eigen D, Zhang X, et al. OverFeat: integrated recognition, localization and detection using convolutional networks. [C]// Advances in Neural Information Processing Systems. [S. l.]: ICLR Press, 2014.

[9] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]// Proc of ImageNet Large-Scale Visual Recognition Challenge Workshop. [S. l.]: ICCV Press, 2013: 10-15.

[10] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]// Proc of ImageNet Large-Scale Visual Recognition Challenge Workshop. [S. l.]: ICCV Press, 2013: 10-15.

[11] Kazemi F M, Samadi S, Poorreza H R, et al. Vehicle recognition using curvelet transform and SVM [C]// Proc of the 4th International Conference on Information Technology. [S. l.]: IEEE Press, 2007: 516-521.

[12] Girshick R. Fast R-CNN [C]// Proc of IEEE International Conference on Computer Vision. [S. l.]: ICCV Press, 2015: 10-15.

[13] Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks [C]// Proc of Conference on Neural Information Processing Systems. [S. l.]: NIPS Press, 2015: 1-15.

[14] Felzenszwalb P F, Girshick R B, McAllester D, et al. Object detection with discriminatively trained part-based models [C]// Proc of IEEE Transactions on Pattern Analysis and Machine Intelligence. [S. l.]: TPAMI Press, 2010.

[15] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[16] Zeiler M D, Fergus R. Visualizing and understanding convolutional neural networks [C]// Proc of European Conference on Computer Vision. 2014.

[17] Wan Shining. Research and implementation of face recongnition based on convolution netural network [D]. Chengdu: University of Electronic Science and Technology of China, 2016.

[18] Song Huansheng, Zhang Xiangqing, Zheng Baofeng, et al. Vehicle detection based on deep learning in complex scene [J]. Application Research of Computers, 2018, 35 (4): 1270-1273.

[19] Shen Lili. Intelligent home robot control system based on the visual identity by kinect [J]. Modular Machine Tool & Automatic Manufacturing Technique, 2017 (12): 75-80, 84.

[20] (Reference for Kinect sensor—implied from context)

Note: Figure translations are in progress. See original paper for figures.