Object Detection on Images in Docking Tasks Using Deep Neural Networks

Ivan Fomin(✉), Dmitrii Gromoshinskii, and Aleksandr Bakhshiev

The Russian State Scientific Center for Robotics and Technical Cybernetics (RTC),
Tikhoretsky Prospect, 21, 194064 Saint-Petersburg, Russia
{i.fomin,d.gromoshinskii,alexab}@rtc.ru

Abstract. In process of docking of automated apparatus there is a problem of determining of them relative position. This problem may be effectively solved with algorithms for relative position calculation, based on television picture formed by camera, installed on one apparatus and observing another one, or docking position. Apparatus position and orientation calculates using visual landmarks positions and information about 3D configuration of observing object and visual landmarks' relative positions. Visual landmarks detection algorithm is the crucial part of such solution. Study of ability of application of object detection system based on deep convolutional neural network to task of visual landmark detection will be discussed in this article. As an example, detection of visual landmarks on space docking images will be discussed. Neural network based detection system learned using images of International Space Station received in process of docking of cargo spacecrafts will be represented.

Keywords: Object detection · Deep neural networks · Convolutional neural networks · Faster R-CNN · Machine learning · Computer vision

1 Introduction

1.1 Relevance of the Problem

One of the most sophisticated and relevant problems in area of automated apparatus docking process is determination of relative position between one apparatus and another, or apparatus and docking position. If both apparatus, or at least one of them, equipped with video cameras, this problem can be solved using positions of visual landmarks in images from the camera. As an example, docking between spacecraft and International Space Station (ISS) will be discussed. Nowadays in process of docking this problem solving with special radio-electronic and optical systems, components of this systems must be placed on ISS and spacecraft. Also, all spacecrafts during last 40 years equipped with specialized television system that using in process of docking for additional visual control.

Earlier in articles [1, 2] described application of television system to determining relative position of spacecraft relative to ISS. Special computer applications developed

B. Kryzhanovsky et al. (eds.), *Advances in Neural Computation, Machine Learning, and Cognitive Research*, Studies in Computational Intelligence 736,
DOI 10.1007/978-3-319-66604-4_12

to solve this task. These applications can be installed on special laptop PC on the ISS or on desktop in Mission Control Center. Applications receive video signal from camera, installed on spacecraft, in process of docking, and performs simultaneous detection and tracking of visual landmarks that exists onboard ISS. Using data of such landmarks' positions, known model of camera, relative positions of landmarks determined by precise 3D model of ISS we able to precisely calculate relative positions of spacecraft and ISS by solving PnP problem.

1.2 Statement of the Problem

One of the most important components of developed system, that determines performance of the system and precision of relative position, determined by system is the module for simultaneous detection and tracking of visual landmarks. Other methods are fully mathematically described and their numerical result fully rely on precision of landmarks' pixel positions determined by this module.

Current television system has some specificities. All components on the way from cameras to PC where our system installed are analog, including components for radio signal transmitting and receiving on the ISS and Control Center. Each of these components have some different negative influence on the signal, and all of them may cause different distortions. Examples of distorted images received in the process of docking represented on Fig. 1 [3].

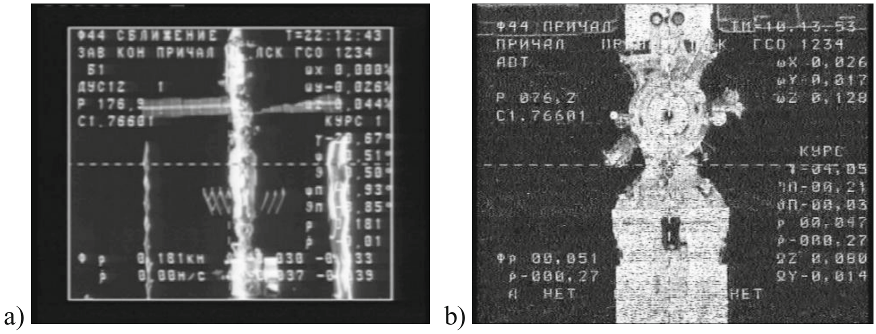

Fig. 1. Examples of distortions: (a) image size distortion, (b) camera matrix noise

On the other side, instead current approach, when to compensate image distortions we use algorithms of different complexity, we can use absolutely different approach. Neural networks at all and partially convolutional neural networks have very good ability to generalization of input learning information. Two years ago, object detection system based on deep convolutional neural network [4] have been introduced. This system utilizes combination of neural networks to detect different objects on images and results with mean average precision up to 78.8%.

We decided to try to apply this system in our problem to detect visual landmarks in images from docking video records, study results of this system in our task and decide, is it make any sense to use this system for object detection in space docking images or other images in our future works.

2 Description of Chosen Systems

2.1 Structure of the Faster R-CNN

To perform our studies, we decided to utilize ready-to-use realization of Faster R-CNN detection system [4] based on the neural networks that implemented using Caffe system and its Python language bindings, named py-faster-rcnn [5].

In this section, we will briefly discuss overall structure of Faster R-CNN and basic principles of how it work. Simplified scheme of the system represented on Fig. 2. Input of the system is the image. Firstly, input image processing by convolutional naurel network. This part of network contains convolutional layers with ReLU error function and pooling layers between them.

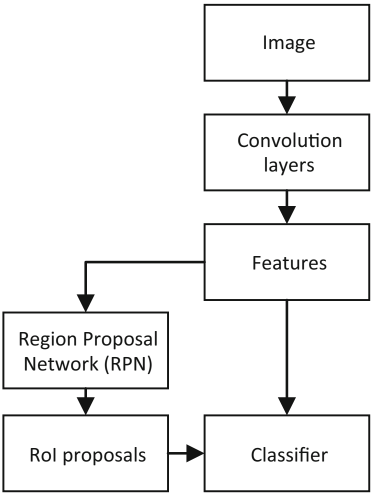

Fig. 2. Scheme of Faster R-CNN system

Convolutional layers perform operation of convolution of small kernels (usually square) with all channels, that passed to the input of the layer. Pooling layers usually choose one better output from each area in each output of previous layer, usually pooling performed in small squares like 2 × 2 or 3 × 3.

Outputs of convolutional neural network are feature maps, that firstly goes to the input of special network for generation propositions of regions of interest (RoI), that may contain or not contain some objects.

This network uses results of the convolutional layers to predict possible positions of objects in the source image and possibility of being an object for each such region.

After moment all RoIs are generated, they pass to the input of the classifier network part. Every RoI projects to the output of the last convolutional layer and resulting patch transforms to the vector of standard size and passes to the input of the classifier that is fully-connected neural network.

2.2 Learning of Faster R-CNN

Faster R-CNN standard way of learning is sequential learning of each part of the network. Because networks for RoI proposition and object classification share same convolutional layers in lower part of whole network this convolutional layer can be learned together. Classifier (fully-connected part) waits for fixed region proposals during process of learning, more precisely for RoIs that formed with similar rules for each image in each learning batch. While RPN learning, all weights are changing, rules of RoI selection are also changing and it is very hard for learning process to converge. Then in practice standard learning procedure contains few steps, where RPN and classifier learns sequentially. First, system leans RPN layers with convolutional layers from scratch. Then learned on previous step convolutional layer and RPN using to learn classifier. On third step weights of convolutional net are fixed already, and system performs RPN fine-tuning. And finally, on fourth step system using fixed fine-tuned RPN to finetune fully-connected classification layers.

In accordance with authors' instruction, before learning classifier and convolutional layer initializing with weight values pre-learned on 1000-classes ImageNet dataset and using this weight values in learning steps. Convolutional neural networks are "deep learning" nets, their weights forms in the process of learning from random values, each layer learn convolution kernels or feature detectors of different scale. To learn network to generate very good feature detectors network must be learned on very big count of examples. Because free datasets for object are very small if we compare them to object classification datasets authors decided to use weights of convolutional and classification parts previously learned on ImageNet dataset and for each new category set and last layer config they only fine-tune weights of all layer in assumption that new categories somehow similar to ImageNet categories. Authors showed that this approach significantly improve their result in PASCAL VOC Challenge.

3 Experimental Researches

For detection system learning we prepared dataset with images that received in process of docking of space apparatus that docked in different time to different docking nodes in International Space Station (ISS). Positions of all docking parts visible in each frame were marked by hands.

To improve quality of object detection we tried to apply augmentation to the data. Source list of images was increased five times, including original image and image with height and width increased by 5% and 10% respectively.

Py-faster-rcnn contains three different classification network models, RPN part is fixed for each model. First model is ZF (network from Zeiler-Fergus article), second is VGG-CMM-M-1024, same as ZF but parameters of some layers are modified. Final architecture is VGG-16, that showed best result in PASCAL VOC Challenge in detection discipline, in time when original article was published. All models are described properly in work [6] and deploy versions represented in BVLC Caffe Model Zoo [7].

Unfortunately, VGG-16 architecture not available to experiment because it needs more than 4 Gb of video memory to perform learning. Models ZF and VGG-CNN-M-1024 was both tested on different versions of our datasets, and will be compared in the tables below (Table 1).

Table 1. United table of the results

	Node 1	Node 2	Node 3	Node 4
VGG-1024, all objects, no changes	0,790	0,363	0,309	0,394
VGG-1024, augmented	0,765	0,345	0,297	0,508
VGG-1024, high contrast	0,779	0,377	0,250	0,454
ZF, all objects, no changes	0,708	0,316	0,219	0,345
ZF, high contrast	0,727	0,365	0,217	0,36

We learned both network using three different datasets. First dataset is source set, where all objects from 4 docking nodes are collected in one dataset. Second set is source, that extended using data augmentation, described before. Images in third dataset are equals to first, but each image additionally processed by contrast filter, because all images in source set have low contrast, some of them extremely low.

To receive some comparable quality metrics for our results we decided to use Mean Average Precision quality metrics, that was calculated according to rules, that can be found in full PASCAL VOC Challenge rules document [8]. This metric for now is standard to quality measurement in object detection systems. To make our results short, Precision-Recall curves and mAP calculated for all objects in test set together.

Tests shown us that increasing of learning set multiple times has very low positive effect on detection quality. Contrast increasing has low positive effect for one separate object type, and no effect for the rest objects. Most of best results shown by source network learned without filtering and/or augmentation.

If we compare results to similar with another architect, object detection mAPs with VGG-CNN-M-1024 detection network are a bit higher than respective values with ZF network. Some objects are detected better, another ones – worse.

4 Conclusions

In the process of our research, we performed different experiments of possibility pf application of Faster R-CNN neural-based object detection system to the task of object detection in process of docking, as an example we used space docking images. Received results show us, that system is able to detect objects with good contrast to surrounding frame area, such as special docking target (used by operator to lead spacecraft to the docking node) or large docking unit. Visible features, that has low difference from surroundings, or any other part of the station are almost never detected. Augmenting by deformation of the size of each image with small changed in ground-truth rectangles corners coordinates do not increased robustness of detection, or quality. Some results, that not represented here, show that increase of contrast and sharpness of testing set results in higher quality of detection. Influence of filtering will be tested more thoroughly in our future work.

Attempt to use simpler architecture with different layers' size resulted in lower detection quality, so in the future works we will test last of represented architects, that should result in best quality of detection.

References

1. Stepanov, D., Bakhshiev, A., Gromoshinskii, D., Kirpan, N., Gundelakh, F.: Determination of the relative position of space vehicles by detection and tracking of natural visual features with the existing TV-cameras. Analysis of Images, Social networks and Texts, Four International Conference, AIST 2015, Yekaterinburg, Russia, 9–11 April 2015, Revised Selected papers. Communications in Computer and Information Science, vol. 542, pp. 431–442 (2015)
2. Bakhshiev, A.V., Korban, P.A., Kirpan, N.A.: Software package for determining the spatial orientation of objects by TV picture in the problem space docking. In: Robotics and Technical Cybernetics, Saint-Petersburg, Russia, RTC, vol. 1, pp. 71–75 (2013)
3. InterSpace. The channel is hosting a record of all space launches in the world. https://www.youtube.com/channel/UC9Fu5Cry8552v6z8WimbXvQ. Accessed 19 Apr 2017
4. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Neural Information Processing Systems (NIPS) (2015)
5. Girshick, R.: Faster R-CNN (Python implementation). https://github.com/rbgirshick/py-faster-rcnn
6. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the Devil in the Details: Delving Deep into Convolutional Nets. British Machine Vision Conference (2014)
7. Model Zoo – BVLC. affe Wiki – GitHub. https://github.com/BVLC/caffe/wiki/Model-Zoo
8. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal Visual Object Classes (VOC) Challenge. http://host.robots.ox.ac.uk/pascal/VOC/pubs/everingham10.pdf