Visual features detection based on deep neural network in autonomous driving tasks

Ivan Fomin, Dmitrii Gromoshinskii, Dmitry Stepanov
Computer vision lab
Russian State Scientific Center for Robotics and Technical Cybernetics (RTC), Saint-Petersburg, Russia
{i.fomin, d.gromoshinskiy, dnstepanov}@rtc.ru

Abstract

Navigation using visual features is one of the ways to determine automated vehicle position. In autonomous automated driving system this way can be used when other systems do not work or unavailable. Vehicle position and orientation can be determined using 2D top-down map with marked visual features, known camera and vehicle models, information about camera position on the vehicle.

This paper proposes application of deep learning neural network Faster R-CNN in object detection for the autonomous driving task.

Keywords: *Faster R-CNN, object detection, autonomous driving, neural networks, deep learning.*

1. INTRODUCTION

Nowadays autonomous navigation of vehicle is interesting and hard problem. Vehicle is a car or a mobile robot, also a mobile robot inside a group of robots. To solve autonomous navigation problem, we need to localize vehicle in known map. One of approaches to vehicle localization is detection of position and orientation of vehicle relative to map, using camera, installed on vehicle, and information about natural visible features positions. Road signs, unique billboards or any other unique objects that have stable appearance and remain in one position for a long time can be used as visible features.

In earlier works we have developed algorithm [4] which performs vehicle orientation detection using trained cascade detector and video camera with known model (calibration). Algorithm needs at least three detected natural landmarks. Map of the scene consists of a set of known visual landmarks. Only relative positions of the landmarks are required to determine orientation of the vehicle. Exact coordinates of landmarks in arbitrary selected coordinate system and measure units are required to determine coordinates of vehicle in same units and coordinate system.

In this algorithm we used special visual landmark cascade detector based on well-known Viola-Jones detector. There is more detailed description of current detector in [6]. Algorithm of object detection utilizes sliding window and Statistically Effective Multi-scale Block Local Binary Patterns (SEMB-LBP) as descriptors and also especially trained cascade classifier.

One of the challenges in the task is using of camera with ultra wide-angle (Fisheye) lens for navigation. This type of lens provides very large distortions when object is close to peripheral zone of the image. Detection algorithm should be able to deal with distortions.

In this work we explore application of deep learning convolutional neural network Faster R-CNN [5] for object detection on our own dataset. Neural network with this architecture has good generalization ability and is able to detect objects with very different appearances and very large distortions only if such distortions represented in training dataset.

Neural network requires fixed amount of time to perform object detection on every single image. This time changes according to image resolution and global settings of the neural network. Current cascade detector performs detection in amount of time according to number of objects on the image, significantly grows when the number of objects is growing. Neural network based detector has better performance when image contains many different objects and worse otherwise.

Faster R-CNN is used for object detection on the images extracted from test video records. Records were captured while system of automated vehicle position detection was tested. We compare results from neural network detector with results from current cascade detector.

2. RELATED WORK

2.1 Convolutional neural networks

This type of networks differs from common networks which contain fully connected layers. Convolutional neural networks (CNNs) have different organization and learning principles. CNN contains so-called convolutional and pooling layers. Convolutional layers perform convolution of different small-size kernels with all output channels of previous layer. Then all output channels from current layer are used as input of the next layer and so on. Feature of this network type is learning of convolution kernels in unsupervised mode with relatively little pre-processing. All weights in convolution kernels are evaluated with special modification of backward propagation of errors algorithm. With several convolutional layers put one by one, detectors of high order features can be learned. The deeper network is – the more complicated features can be detected. Quality of feature detectors learned by neural network is significantly higher compared to detectors, formed by different researchers in earlier articles.

Also convolutional neural networks include pooling layers. Pooling layer takes output from convolutional layer as an input and selects neuron with maximal activation function in each small square region (usually 2x2) and passes it to the next layer. This operation decreases layers' dimensions and increases algorithm performance, also makes detectors invariant to feature position.

2.2 Fast R-CNN

Application of neural network as primary part of object detection system is shown in work [1], that describes architecture of object detection system based on convolutional neural network R-CNN. This neural network is used not directly for the object detection, but for classification of rectangles, detected by external algorithm. The input of the network is an image with marked regions of interest, RoI (rectangles). Each RoI is used as an input of convolutional layers of the R-CNN network, and then outputs of the convolutional layers is used as input of fully connected part of R-CNN, that performs classification of selected RoI - if RoI contains an object or background. If RoI contains object it will be referred to one of object classes passed to classifier in training time. This approach demonstrates applicability of neural networks to the problem of object detection.

Fast R-CNN [2] is critical improvement of R-CNN architecture, developed to make object detection faster. One of special algorithms generates RoI as well as for R-CNN. Then convolutional layers of network decreases input image resolution and generates feature search result outputs. Each RoI projects on each of this outputs according to image changes in convolutional layers. Then for each RoI pooling layer extracts constant-size feature vector. Each feature vector is fed into a fully-connected classification network which returns result as a list of possible objects with probabilities. Result with maximum probability is passed to bounding box fitting part of network that improves rectangle of RoI position to better fit the object position.

Fast R-CNN has better performance because convolution of image is performed not for each RoI separately but for whole image. All parts of network share results of convolution. This also results in better convolutional layers feature detection quality, because convolutional layers are not trained on small parts of each source image but on all images as a whole.

3. SELECTION OF NETWORK ARCHITECTURE

3.1 How Faster R-CNN works

After joint convolutional layers training and image processing were introduced in Fast R–CNN, the next logical step is to exclude external region proposal tool from the detector. Using special neural network for forming region proposals allows to improve quality and performance of the detector. In this way method becomes fully based on neural networks and fully learnable. Also, RoI detector used in R-CNN and Fast R-CNN was performed on CPU, that significantly restricted performance of detection in a whole, because this part of method was very slow compared to neural network part performed on GPU. RoI detection with neural network allows to increase algorithm performance.

Faster R-CNN is classification part of Fast R-CNN united with specially developed Region Proposal Network (RPN). RPN and classification parts utilize the same convolutional layers, it results in speedup and quality improvements. Scheme of different parts of neural network connection is represented in figure 1.

New part in Faster R-CNN is RPN. Let's describe this part more thoroughly. This neural network uses sliding window principle for generating candidate regions of interest (rectangles) in each position and for each convolutional layer's output.

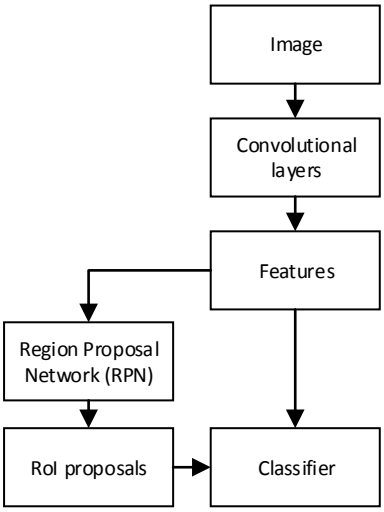

Figure 1. Connection between convolutional layers, RPN and classificator

RPN uses sliding window to generate proposals of object bounding boxes. For each sliding window position in last convolutional layer sliding window convolves with filter and result projected to low dimensional space to acquire fixed-size vector for each bounding box position. Then feature vector is passed to RoI fitting layer and RoI classification layer. Fitting layer generates some variants of bounding box form for each sliding window position. For each form classifier network calculates two outputs – probability of fact that bounding box contains an object (object probability), and probability that bounding box is a background. Non-maxima suppression by object probability score is applied to decrease the number of bounding box forms, for ones that have very big intersection rate with each other. Then remaining bounding boxes from all sliding window positions combined in one list ranked again by the object probability, fixed number of top ranked results passes to a classification layers.

Main principle of RPN learning is similar to Fast R-CNN learning, both networks use same convolution layers and similar backpropagation methods for fully-connected layers. Only inputs and outputs for fully-connected layers are different. Proposal of RoI is marked as good example if intersection over union (IoU) of generated proposal with one of the ground truth object frame more than some selected high threshold, or the rectangle have maximal IoU. Rectangle of RoI is marked as bad example if IoU with any ground truth rectangle is lower than another threshold that is usually low. Having large difference between two thresholds we can avoid training on bad examples.

All positive learning RoI examples are passed to the input of the classifier from Fast R-CNN network. Afterwards rectangles are classified by scheme described in Section 2.2.

Convolution layers in Faster R-CNN are used for RPN and classifier together. Since classifier network requires fixed rectangle regions as an inputs at the training stage for every training image it is not possible to learn RPN and classifier together. Rules of region proposals detection are changing during the RPN iterative learning. Fast R-CNN classification part cannot be learned in iterative process with different input regions in each

iteration. So, iterative sequence of learning steps is selected to train all networks. The first step is to separate convolutional network learning followed by RPN learning on ground truth examples. The second step is using RPN, learned on previous stage to learn Fast R-CNN part – to train convolutional layers and classification layers with RPN region proposals from scratch. At the third step convolutional network layers, learnt on second stage remain fixed and RPN is learnt again using fixed convolution results and data from classifier network. At the fourth step convolution layers remain fixed, also RPN layers become fixed and final classifier learning and tuning are performed.

3.2 Selection of network architecture

"Py-faster-rcnn" [3] package includes three network configurations used in described detection system: ZF, VGG-CNN-M-1024 and VGG-16 from [7], [2] and [5] respectively. These configurations have different complexity of convolutional part and links between network layers, consequently different learning difficulty, resources required for training, performance and quality of detection. First net named ZF consists of 5 convolutional layers and 3 fully-connected layers (classification layers). Second net named VGG-CNN-M-1024 has the same amount of convolutional layers but with different kernel field size and linking rules for layers. Net VGG-16 is the most complex of all considered configurations; it has 13 convolutional layers, needs many hours for training, processing takes more than half a second per image. Thereby this configuration is not suitable for real-time application with more than 5 FPS. The net cannot be learned on the current hardware configuration. As a result, network architecture VGG-CNN-M-1024 is chosen for primary tests of visual feature detection in automated driving tasks.

4. EXPERIMENTAL RESULTS

4.1 Dataset description

Special test set of images was prepared for experimental evaluation of quality of visible landmarks detection using detector based on neural network. Dataset consists of images extracted from video files used for learning of current cascade detector. We use 36 videos with different duration from two cameras. Most of videos (27 from 36) were captured by PointGrey camera with Fisheye lens, the angle of view is 185 degrees. Frames are grayscale, resolution is 1040x776 pixels. The rest of the videos were captured by Basler camera with the same lens and angle of view. Frames are color, resolution is 1600x1200 pixels.

Videos frame rate is 60 FPS. Frames are extracted in a special way, so objects have significant offset in every subsequent frame.

We used 7087 extracted and marked images, 18483 training examples for learning, as shown in Table 1. We train neural network of selected configuration using source code written by author of "Py-faster-rcnn" pack with standard parameters. According to instruction approved by author of pack dataset was not separated into two parts: training part and testing part which is purposed only for fine-tuning.

Table 1: Training and testing datasets

Objects	Amount of examples	
	Training dataset	***Testing dataset***
Fire Shield	1689	-
Circle Marker	1376	379
Chess board	1258	-
Traffic Light Triple	770	-
Crosswalk Light Sign	2241	261
Crosswalk Blue Sign	2335	357
Bricks	679	-
Traffic Light	1651	455
Children Sign	1842	328
Box	872	-
Triangle Marker	1533	346
Bus Stop Sign	1776	418
Electrical Shield	461	-
All	18483	2544

4.2 Experiments description

For comparing detection quality of cascade detector and neural network with VGG-CNN-M-1024 architecture we chose and marked sequence of images extracted from video captured by PointGrey camera. We extracted 942 images and marked 2544 examples of objects of 7 classes.

The sequence is used as testing dataset and images of this sequence were never used in training dataset. Images in sequence have the same type as images from training set. Explicit results of marking are shown in Table 1.

We tested both detectors in equal conditions on the same PC, operation system and hardware configuration. Results of comparison are shown in Tables 2 and 3.

Table 2: Comparison of precision and recall

Objects	Neural network		Cascade detector	
	Precision	***Recall***	***Precision***	***Recall***
Children Sign	0.50	0.55	0.67	0.33
Bus Stop Sign	0.47	0.30	0.54	0.33
Crosswalk Blue Sign	0.44	0.09	0.68	0.49
Crosswalk Light Sign	0.47	0.31	0.49	0.49
Circle Marker	0.73	0.37	0.52	0.46
Triangle Marker	0.61	0.44	0.54	0.70
Traffic Light	0.33	0.27	0.21	0.21

Table 3: Characteristics of detection quality

Objects	Neural network		Cascade detector	
	FPR	***Hit per img***	***FPR***	***Hit per img***
Bus Stop Sign	0.53	0.30	0.46	0.43
Crosswalk Light Sign	0.53	0.31	0.51	0.53
Crosswalk Blue Sign	0.56	0.09	0.32	0.51
Circle Marker	0.27	0.37	0.48	0.46
Traffic Light	0.67	0.27	0.79	0.27
Triangle Marker	0.39	0.44	0.46	0.75
Children Sign	0.50	0.54	0.33	0.46
All	0.49	0.31	0.48	0.47

In precision and recall terms shown that precision of neural network detector is higher than precision of cascade detector. Recall of cascade detector is a bit higher due to difference of calculating positive detections count for both detectors.

False positive rate (FPR) differs from one object to another, this value depends on count of training and testing examples for each object. Hit rate per image (true positive rate per image) is higher for cascade detector due to very small amount of false detections.

To improve object detection by neural network we should use synthetic dataset for training with amount of images several times more than in current work. Furthermore, we should consider object position on the previous frame to decrease search region and time.

5. CONCLUSION

This article describes application of neural network detector for natural landmark detection in problem of unmanned vehicle navigation.

Neural network demonstrates results comparable with cascade detector results despite the restricted size of learning dataset. Performance of neural network for object detection is higher than cascade detector performance when number of objects is large. Neural network detector has good ability to improve the performance and detection quality. This proves applicability of neural network detector for solving task of natural landmarks detection in problem of unmanned vehicle autonomous navigation.

Future research includes extension of training dataset for increasing detection quality and precision using a lot of synthetic images; investigation of different network configurations with different hyper-parameters to figure out influence of parameters change on results.

6. REFERENCES

[1] Ross Girshick, Jeff Donahue, Trevor Darrel, and Jitendra Malik, "Rich feature hierarhies for accurate object detection and semantic segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[2] Ross Girshick, "Fast R-CNN," Proceedings of the International Conference on Computer Vision (ICCV), 2015.

[3] Ross Girshick, "Faster R-CNN (Python implementation)," https://github.com/rbgirshick/py-faster-rcnn

[4] A.M. Korsakov, I.S. Fomin, D. A. Gromoshinskii, A.V. Bakhshiev, D.N. Stepanov and E. Y. Smirnova, "Determination of an unmanned mobile object orientation by natural landmarks," Analysis of Images, Social networks and Texts, Four International Conference, AIST 2015, Yekaterinburg, Russia, April 9-11, 2016, Revised Selected papers, *unpublished*

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," Neural Inormation Processing Systems (NIPS), 2015.

[6] D. Stepanov, A. Bakhshiev, D. Gromoshinskii, N. Kirpan, and F. Gundelakh, "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, April 9-11, 2015, Revised Selected papers. Communications in Computer and Information Science, vol. 542, pp. 431-442.

[7] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. CoRR, abs/1311.2901, 2013.