Automatic detection of objects on star sky images by using the convolutional neural network

A I Bobrovsky¹, M A Galeeva³, A V Morozov², V A Pavlov³ and A K Tsytsulin²

¹ State Research Institute of Instrument Engineering, Naberezhnaya Obvodnogo Kanala, 29, St. Petersburg, Russia
² Television Research Scientific Institute, Politeknicheskaya ul., 22, St. Petersburg, Russia
³ Peter the Great St. Petersburg Polytechnic University, Politeknicheskaya ul., 29, St. Petersburg, Russia

E-mail: pavlov_va@spbstu.ru

Abstract. The automatic detection methods of mobile objects on moving star background in the presence of high-speed image blurring are considered. The approach based on convolutional neural network YOLO v2 for detection and classification of two kinds of objects images is analysed. Estimations of detection accuracy and speed for the approach are based on convolutional neural network of a simplified structure, indicating further prospects for use of such networks.

1. Introduction

The task of automatic detection has a wide range of applications: video monitoring; analysis of a target objects movement (including artificial space objects, ASO); control of a process of convergence of a spacecraft (SC) [1] etc. The choice of automatic detection method depends on the mobility of a camera and a background. Two types can be distinguished: fixed camera – constant background and moving camera – changing background.

In the first type, there are four groups of detection methods [2]: background subtraction methods [3]; time difference methods [4]; probability method [5]; optical flow methods [6]. In the second type, automatic detection priory needs more computing resources. It requires a suitable classifier that should be preliminary trained (support vector machine [10, 15, 16], artificial neural network [10, 17], decision trees [10], k – nearest neighbours algorithm [10]) on a specific set of images and features.

In the second type, two categories of methods are used [12]. In the first category, the problem of classifying objects, according to a set of features, is solved. An image is divided into rectangular regions of interest (ROI). A set of features is extracted from each ROI, for example: Haar features [7], histograms of oriented gradients [8, 9], convolutional features [10] etc. At the next stage, a set of features is passed to the input of the classifier [8, 9] for identification of ASO in a feature space.

The second category of methods is based on convolutional neural networks [11]. The image is passed to the input of the convolutional neural network entirely. The result is the coordinates of a bounding boxes and a probability of belonging of objects to specific classes.

The aim of the work is to develop a neural network method of automatic detection of space objects on a starry background for a moving camera and a changing background. All objects recorded in image frames and located at long distances (>1000 km) are „circle“ (point-type) and belong to two
classes: „mobile” (objects with speed blur [19]) and „stationary” (in the instrument coordinate system). Examples of object images are shown in figure 1.

The review of publications [13, 14] showed that the algorithms of the YOLO family [11-13] allowed to achieve the best accuracy of detection and classification. YOLO v2 architecture was chosen for further research [12]. In YOLO v2, the fully connected layers are removed, so the input network resolution can be changed upon detection. YOLO v3 [13] shows the best results in accuracy. However YOLO 3 is inferior to YOLO v2 in processing speed due to the high complexity of the architecture.

Figure 1. Images of objects at long ranges. Left image – mobile object (approximate size 7 × 40 pixels). Right image – stationary object (approximate size 11 × 11 pixels).

2. Description of the algorithm
Classifier in YOLO v2 is based on convolutional neural network DarkNet19 [12]. DarkNet19 consists of 19 convolutional layers, 5 max-pooling layers, global average pooling layer and softmax layer. DarkNet19 was trained by authors [12] for 160 epochs in the dataset ImageNet 1000 [12]. After training in 224×224 resolution images, authors increased the resolution to 448×448 and trained the network for another 10 epochs.

For detection of object, authors in [12] modified DarkNet19 by removing the last convolutional layer and instead adding on three 3×3 convolutional layers with 1024 filters each followed by a final 1×1 convolutional layer with the number of outputs that are required to detect and classify of classes C.

The input image is divided by a grid with size S×S cells (in work [12] S=13). For each cell, the network predicts anchor boxes. They are formed before training using the k-means method [12] on the basis of the training dataset. For each anchor box, the network predicts (figure 2): 4 parameters (x, y, w, h) to correct the size and position of the anchor box, 1 confidence score and C probabilities by classes. For one cell (C+5)×B values is formed, where B – number of anchor boxes.

Figure 2. The prediction scheme for result of detection.
During operation, an image is passed to the network input, for which the prediction of the position of objects and their classes is performed. As an activation function, authors used „Leaky ReLU” (Leaky Rectified Linear Unit). To detect objects with different sizes during YOLO v2 training, the resolution of the input images was being changed randomly every 10 batches.

Architecture was chosen experimentally, consisting of 6 convolutional layers and 4 max-pooling layers. The structure of the neural network detector and classifier based on tinyYOLO v2 is shown on figure 3.

When compressing the original frame of a large image to the size of the input of the detector network (for example, 416×416 pixels), reduction in size and distortion of the images of objects are occurred. The original image is divided into overlapping fragments to eliminate compression. The overlap width of neighboring fragments of 416×416 pixels was chosen to be 83 pixels, which approximately corresponds to a value of 0.2 for the overlap coefficient. Figure 4 shows the structure of the process of detecting and classifying objects in a large-sized image using a neural network. A fragment of an image with a size of 416×416 is passed to the input of the detector. The result of the network on all fragments of the image is a set of \( M \) vectors describing the detected and classified \( M \) objects.

Each vector contains 6 elements: the position of the rectangle \((x, y, w, h)\), object class \((c)\) and the probability of an object belonging to a certain class \((p)\). Then, the resulting image is formed from the image fragments, on which the bounding rectangles are applied.

![Figure 3. The structure of the neural network detector and classifier based on tinyYOLO v2.](image)

![Figure 4. Illustration of the parametric structure of process of detection and classification of objects by the network in the high-resolution image.](image)

The training was conducted on fragments obtained from 100 original images. The training dataset consists of 153 fragments and contains 1104 images of objects of class 1 and 2. Training parameters: batch size – 64 images, initial value of learning rate – \( 10^{-3} \), momentum – 0.9; decay – 0.0005.

Parameters of the computing platform on which the network was trained and tested: GPU NVIDIA GTX 1050 TI, processor Intel Core i5–4570, 3.20 ГГц; 8 GB of RAM. The training software is implemented in C / C++ with support for CUDA technology. Testing was carried out on the same software, but without the support of CUDA technology.
3. Experiments
For evaluating the characteristics and testing of the algorithm of detection and classification of small-size space objects, the special technique was developed. The technique includes the following main stages: detection of frames for testing; marking of frames and their fragments by the expert; submission of test frames for processing and registration of detection results; obtaining intermediate estimates; obtaining final estimates.

For the experiments $NT = 10$ different frames of the starry sky were detected. They contained images of stars, interstellar space, planets, spacecraft and interference. The frames were selected from a set provided by the Pulkovo Observatory. These frames were not included part of the image set used to train the network.

The sighting axis of the camera is stable relative to the local vertical. Images of the observed luminous objects have „high-speed blur” due to long exposure time. Only a small number of geostationary or „camera stationary” objects have images in the form of a circle or point. Accordingly, there are two classes of images of point glowing objects („blur”, „circle”).

The test detection algorithm work with a rectangular image size $N_{i\alpha} \times N_{i\beta} \times N_d$ ($N_{i\alpha}$ – number of pixels in row, $N_{i\beta}$ – in column and $N_d$ – number of bits per pixel). At the next stage, the expert advisor selects images of two classes on the fragments („blur”, „circle”) and for them, defines the values of 5 parameters: $x, y, w, h, c$, where $x, y - a$ coordinates of the center of a rectangle (for class 1) or square (for class 2), sets its position in a frame with a precision of 1 pixel; $w, h$ – width and height of a rectangle with a precision of 1 pixel; $c$ – image class (1 or 2).

As a result of the network processing a fragment $f$ of a frame $k$, which contains $M_f = M_1 + M_2$ true images of objects of two classes, an array will be obtained. This array contains information about the $MO_{ij}$ of detected and classified object images. After processing the entire frame will be gotten an array of $MO_{ij}$ number of detected objects.

The $MO$ value for the entire test frames set and for each test frame ($MO_k$), can be represented as the sum of the following values (the index $k$ is removed for simplicity): $MC_1$ – number of correctly detected and classified objects of type 1 on all fragments; $MC_2$ – number of correctly detected and classified objects of type 2 on all fragments; $ME_{12}$ – number of erroneous classified objects of type 1 (type 2 is classified by type 1) on all fragments; $ME_{21}$ – number of erroneous classified objects of type 2 (type 1 is classified by type 2) on all fragments; $MS_1$ – number of skipped (skip) objects of type 1 on all fragments; $MS_2$ – number of skipped objects of type 2 on all fragments; $MF_1$ – number of false detections (type 1 detected) on all fragments; $MF_2$ – number of false detections (type 2 detected) on all fragments.

The procedure for determining the correctness, error and falsity of object detection and classification is based on the calculation of „proximity” of the values of the elements of the results array corresponding to the elements of the array of reference results. For two classes of objects there are four possible types of decisions: „correct detection”, „wrong classification”, „object skip” and „false detection”. For evaluation of „proximity” true and predicted result of the detector the measure of the intersection of rectangles is applied:

$$I = A(R_i \cap R_j)$$  \hspace{1cm} (1)

where $R_i$ – a reference rectangle, $R_j$ – a predicted rectangle.

When considering pair $i, j$ of rectangles following cases are possible:

- If $I_{ij} > 0$, $c_i = c_j = 1$, – reference object detected and correctly classified; value of $MC_1$ is incremented;
- If $I_{ij} > 0$, $c_i = c_j = 2$, – reference object detected and correctly classified; value of $MC_2$ is incremented;
- If $I_{ij} > 0$, $c_i ! = c_j$, – reference object detected and correctly classified – if $c_i = 1$, value of $ME_{12}$ is incremented; else value of $ME_{21}$ is incremented.

Final evaluation:
- „probability” of correct detection of objects of the first type: $P_1 = MC_1 / NO_1$;
• “probability” of correct detection of objects of the second type: \( P_2 = MC_2 / NO_2 \);

• “probability” of false detection of objects:
  
  • If \( MC_1 + MC_2 + ME_{21} + ME_{12} = MO \): \( P_{EF} = (ME_{21} + ME_{12} + MF) / (MO_1 + MO_2) \);
  
  • If \( MC_1 + MC_2 + ME_{21} + ME_{12} < MO \): \( P_{EF} = (MO - MC_1 - MC_2) / (MO_1 + MO_2) \);

If \( I_{ij} > 0 \) and \( c_i = c_j \) we can estimate the following parameters:

• absolute error in determining of a centre of a rectangle \( R_{Cij} \):
  
  \[ \Delta = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \]  \( (2) \)

where \( x_1, y_1 \) – coordinates of the centre of a reference bounding box, \( x_2, y_2 \) – coordinates of the centre of a resulting bounding box;

• relative error of determining the centre of the rectangle:
  
  \[ \delta = \Delta / \sqrt{A(R_i)} \]  \( (3) \)

where \( A(R_i) \) – an area of a reference rectangle.

4. Results

An example of the result of the algorithm of automatic detection of objects is shown in figure 5.

![Figure 5. The result of the detector on 2 different fragments.](image)

For the entire test dataset (120 fragments, 293 objects of the 1st type and 54 objects of the 2nd and second type) the following estimates were obtained: \( P_1 = 0.86 \); \( P_2 = 0.97 \); \( P_{EF} = 0.07 \); \( \Delta = 6.7 \) pixels; \( \delta = 0.19 \); the average processing time of a single fragment with a size of 416×416 is 0.1 s.

The computational complexity is determined by the following equation:

\[ O = m \times N \times S_f \times S_f \times c \times H_{out} \times W_{out} \]  \( (4) \)

where \( N \) – filters number; \( S_f \) – (filter size); \( c \) – channels; \( H_{out} \) – output height of feature map; \( W_{out} \) – output width of feature map; \( m = 2 \) – operations number (addition and multiplication). For this object detector, it is \( 1.556 \times 10^9 \) floating-point operations.

In comparison to the results of the work [18], the proposed method shows a better probability of correct classification. In addition, the proposed method allows the detection and the classification of two or more object classes.
5. Conclusions
A promising approach based on a convolutional neural network of YOLO v2 type for detection and classification of small-size images ASO on starry sky background is considered. A simulation of the operation of the detector - classifier of „simple” images of small-sized ASO on starry sky background was performed. The estimates of accuracy and speed of work on the test set of images are obtained. In further studies, it is advisable to expand the set of classes of selectable objects and the set of simulated images with deep convolutional network images. For this, a promising approach based on a convolutional neural network of YOLO v2 type for detection and classify small-size images was performed. The simulation of the operation of the detector - classifier of „simple” images of small sized ASO on starry sky background was performed. The estimates of accuracy and speed of work on the test set of images are obtained. In further studies, it is advisable to expand the set of classes of selectable objects and the set of simulated images with deep convolutional network images.

References
[1] Fomin I, Orlova S, Gromoshinskii D and Bakhshiev A 2018 Object detection on docking images with deep convolutional network Studies in Computational Intelligence 799 136-143
[2] Maggio E and Cavallaro A 2011 Video Tracking: Theory and Practice
[3] Cucchiara R, Grana C, Piccardi M and Prati A 2003 Detecting Moving Objects, Ghosts and Shadows in Video Streams IEEE Transactions on Pattern Analysis and Machine Intelligence 25 1337–1342
[4] Bevilacqua A, Di Stefano L and Lanza A 2004 An Efficient Change Detectioin Algorithm Based on a Statistical Non-parametric Camera Noise Model Proc. International Conference on Image Processing 4 2347–2350
[5] Friedman N and Russell S 1997 Image Segmentation in Video Sequences: A Probabilistic Approach Proc. Annual Conference on Uncertainty in Artificial Intelligence 175–181
[6] Barron J, Fleet D and Beauchemin S 1994 Performance of Optical Flow Techniques International Journal of Computer Vision 12 43–77
[7] Viola P and Jones M 2001 Rapid Object Detection using a Boosted Cascade of Simple Features IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001) 511
[8] Dalal N and Triggs B 2005 Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. IEEE Computer Society Conference 1 886–893
[9] Felzenszwalb P et al 2010 Object detection with discriminatively trained part based models IEEE Transactions on Pattern Analysis and Machine Intelligence 32 1627–1645
[10] Haykin S 1998 Neural Networks: A Comprehensive Foundation (Prentice Hall) 842
[11] Redmon J et al 2016 You Only Look Once: Unified, Real-Time Object Detection IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 779-788
[12] Redmon J and Farhadi A 2017 YOLO9000: Better, Faster, Stronger IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 7263-7271
[13] Redmon J and Farhadi A 2018 Yolov3: An incremental improvement (Preprint arXiv:1804.02767)
[14] Lin T et al 2017 Focal loss for dense object detection (Preprint arXiv arXiv:1708.02002).
[15] Utkin L and Zhuk Y 2017 Interval SVM-Based Classification Algorithm Using the Uncertainty Trick International Journal on Artificial Intelligence Tools 26
[16] Utkin L 2019 An imprecise extension of SVM-based machine learning models Neurocomputing 331 18-32
[17] Sonkin K, Stankevich L, Khomenko Y, Nagornova Z, Shemyakina N, Koval A and Perets D 2016 Neurological classifier committee based on artificial neural networks and support vector machine for single-trial EEG signal decoding Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9719 100-107
[18] Morozov A 2017 Star sky images adaptive binarization Voprosy radioelektroniki ser. Tekhnika televiziony 1 23–30
[19] Korobeynikov A, Grishentsev A, Velichko E, Korikov C, Alekssanin S, Fedosovskii M and Bondarenko I 2016 Calculation of regularization parameter in the problem of blur removal in digital image Optical Memory and Neural Networks (Information Optics) 3 184-191