Real-time Recognition Method of The Ship Object Based on Video Image

Yupeng Xing*, Xiaojun Li and Fengtang Fang

CSSC SYSTEMS ENGINEERING RESEARCH INSTITUTE, Beijing, 100094, China

*Corresponding author’s e-mail: Xingyupeng818@sina.com

Abstract. Target recognition has always been a hot topic in the field of computer vision. In the military field, computer vision has a good application in the research of ship target detection methods based on SAR images, but this type of research is limited to the detection of ship targets, and there is no further identification of ship types. Shipborne tracking radar parasitic TV can obtain clear images of sea surface targets in sea and air background. In order to improve the combat capability of ship combat systems, this paper proposes a video-based ship target recognition method, based on the identification of ship targets. This paper takes the hand-labeled TV video as the research object, and uses the Camshift and Kalman filter to improve the Faster R-CNN calculation framework for the automatic recognition and tracking of the video image ship target in the sea and sky background. The experimental results show that the correct recognition rate of the target model of the ship based on this method is above 90%.

1. Introduction

In the complex environment of the sea battlefield, efficient intelligence gathering methods can provide strong support for operational command. Ship automatic identification technology provides an important guarantee for accurately tracking ship targets and achieving accurate missile guidance. The correct identification of ship models is conducive to the development of targeted operational plans. Therefore, research on automatic identification of ship-based targets based on video is important. In intelligence acquisition, current research mainly focuses on target detection methods based on SAR images, and no further identification of key information such as target types 0.The traditional ship identification algorithm includes background difference method and recognition algorithm based on BP neural network. The algorithm steps mainly include image pre-processing, ship feature selection and classifier design. The main challenge is that the specific operations of each step need to be specific to the specific application scenario, and the final classification accuracy does not exceed 85% [3].

Deep learning forms a more abstract high-level representation attribute category or feature by combining low-level features. Its application is not constrained by specific scenes, and by adjusting parameters, high classification accuracy can usually be achieved. Convolutional Neural Networks (CNN) is a common method of deep learning and has a good performance in many recognition tasks. Among them, Faster R-CNN applied to real-time detection has achieved quite good results on benchmark datasets such as ImageNet, VOC and KITTI. In this paper, Faster R-CNN is used for real-time target detection, crawler is used to crawl training data from Fleetmon, and Camshift and Kalman filter algorithms are used for moving target tracking. Experiments show that the proposed method can effectively support video-based ship target recognition.
2. Detailed network design

This section describes the detailed design of the network and describes the key principles and parameters of the network.

2.1. Common convolutional layer

The public convolutional layer can use the open source architecture, as shown in Figure 1. Taking VGG-16 as an example, Conv layer contains 13 convolution (conv) layers and 13 Rectified Liner units (ReLU), 4 pooling layers. For each conv layer, there are:

$$\text{kernel}_{\text{size}} = 3$$

$$\text{pad} = 1$$

For each pooling layer, there are:

$$\text{kernel}_{\text{size}} = 2$$

$$\text{stride} = 2$$

Therefore, the image of size $M \times N$ is still $M \times N$ after being processed by the conv layer and the ReLU layer. After the pooling layer, the size becomes:

$$\frac{M}{2} \times \frac{N}{2}$$

The size of the feature map output is:

$$\frac{M}{16} \times \frac{N}{16}$$

2.2. RPN architecture

The role of the RPN network is to recommend candidate regions, which are structured as full convolutional networks that can be end-to-end trained.

2.2.1. Candidate regions (anchors). A candidate area can be understood as a frame of a preset size whose size is related to the size of the input image, and a frame represents an area on the input image. Since the original image and the feature map have a proportional relationship of 16×16, two adjacent pixels on the feature image are separated by 16 pixels on the original image. For each pixel position on the feature map, consider $k = 9$ possible windows, as shown in Figure 1, for 3 areas (128 x 128, 256 x 256, 512 x 512) and 3 aspect ratios (2: 1, 1, 2, 1:1) combination. Therefore, each point on the feature map corresponds to 9 anchors.

![Figure 1. Anchors feature](image)

2.2.2. Box classification (cls). Conditions for assigning a positive label (foreground) anchor: 1) Has the highest IoU overlap with a ground truth box (GT box); 2) It has an IoU overlap greater than $\alpha$ (0.7) for any GT box. Conditions for assigning a negative label (background) anchor: The IoU overlap with
all GT boxes is less than 1-α. Since each point corresponds to k anchors, each anchor corresponds to two positive and negative labels, so after cls, the output becomes 2×k. As shown in Figure 2, the green box is the GT box of the foreground target (ship), and the red is an anchor, apparently meeting the condition 1) for assigning the positive label.

Figure 2. Cls feature

2.2.3. Box regression(reg). After classification, you can get an anchor that may be foreground, as shown in Figure 3, but there is still a gap between the anchor and the ground truth box, and the target may not be detected correctly. Therefore, the box regression is used to transform the anchor so that the anchor is closer to the ground truth box. The target frame is generally represented by a 4-dimensional vector, which represents the coordinates, width, and height of the center point of the target frame:

\[(x, y, w, h)\]

If the initial anchor is represented as \(A = (A_x, A_y, A_w, A_h)\) and GT box is represented as \(G = (G_x, G_y, G_w, G_h)\), then the regression process is to find a mapping relationship \(F\) such that \(A\) maps to a sum. The real box \(G\) is closer to the regression window \(G' = (G'_x, G'_y, G'_w, G'_h)\), as shown in figure 3, the transformation step includes translation and scaling.

Figure 3. Mapping relationship

Pan:

\[
\begin{align*}
G'_x &= A_x + A_w \cdot D_x(A) \\
G'_y &= A_y + A_h \cdot D_y(A)
\end{align*}
\]

Zoom:

\[
\begin{align*}
G'_w &= A_w \cdot \exp(D_w(A)) \\
G'_h &= A_h \cdot \exp(D_h(A))
\end{align*}
\]

Therefore, \(F\) needs to learn four parameters \(D_x(A), D_y(A), D_w(A)\) and \(D_h(A)\). When \(A\) and \(G\) are small, the \(F\) is considered to be a linear transformation, so linear regression can be used to observe the anchor. Make fine adjustments.

Regression process:

Input feature map \(\phi\), \(\phi(A)\) represents the feature vector composed of the feature map corresponding to the anchor, input the real parameter values of the \(A\) to \(G\) transform \((t_x, t_y, t_w, t_h)\),
$W_c^T$ is the parameter to be learned, $d_c(A)$ is the predicted value obtained, then the objective function can be expressed as:

$$d_c(A) = W_c^T \phi(A)$$

The cost function is as follows:

$$\text{Loss} = \sum_{i}^{N} (t_i^c - \hat{W_c}^T \phi(A^i))^2$$

Optimization goal is:

$$W_c^T = \arg\min_{W_c^T} \sum_{i}^{N} (t_i^c - \hat{W_c}^T \phi(A^i))^2 + \lambda \|W_c^T\|^2$$

Where $i$ represents the $i$-th anchor.

2.2.4. Frame filtering (anchorfiltering). After regression and classification, the number of candidate anchors is very large. In order to reduce the subsequent computational complexity, the anchor needs to be optimized. First, determine whether the anchor is beyond the boundary after mapping to the original image, and remove the severely out of the boundary; then sort the softmax according to the region classification score of the anchor, extract the first N regions, and perform non-maximumsupression (NMS). Operation, taking the first M as a candidate area (proposal) input to the RoI Pooling layer.

2.3. RoI Pooling layer

For traditional CNN (such as VGG-16), when the network is trained, the input image size must be a fixed value, and the network output is also a fixed size. If the size of the input image is variable, there are two solutions: the original image is cropped or warp and sent to the network. As shown in Figure 4, both methods will cause loss to the original image.

![Crop and Warp](image)

Figure 4. Crop and warp

RoI Pooling is used to solve the problem that the proposal input from the RPN has different sizes and shapes. The method is: 1) The feature map corresponding to each proposal is divided into P parts horizontally and vertically, and the maximum pooling process is performed for each copy, and the feature value of each proposal becomes P×P; 2) The Top-N fixed outputs are connected to form a feature vector, and the number of samples is Top-N, and the feature dimension of each sample is P×P, which is sent to the fully connected layer.

2.4. Secondary classification and regression

Through the RoI Pooling layer, we have obtained the feature vectors composed of all candidate regions, and then sent to the fully connected layer and softmax to calculate which category each candidate box belongs to, and output the score of the category; at the same time, use box regression to obtain the relative actuality of each candidate region. The offset prediction value of the position is used to correct the candidate frame to obtain a more accurate target detection frame.
3. Video target recognition Algorithm

In order to achieve real-time detection and tracking of video targets, this paper uses Camshift combined with Kalman filter to track moving targets. The algorithm flow is shown in Figure 6. Since the target detection of the Faster R-CNN is running in the GPU, the target tracking algorithm runs in the CPU, so the recognition framework defines dual-threaded real-time calculation, that is, the detection thread and the tracking thread, due to the algorithm The two threads are computed in parallel, so the trace thread does not have to wait for the result of the test thread.

The global variable Flag is used to mark whether a new target has been detected. The algorithm flow is as follows:

Step1: Target Recognition. Starting from the initial frame of the video, the detection thread is used to detect the target in the image, and the extraction target frame is used as the initialization of the Kalman filter of the tracking thread;

Step2: Based on the target tracking result of the first frame, using Kalman filter to predict the position of the target frame in the current frame;

Step3: According to the target frame position obtained in step Step2, the search window is initialized, and the back projection image is calculated, and then iterated using the Camshift algorithm until it matches the target frame position, and the position of the search window is Track the position of the target in the current frame;

Step4: Correcting the Kalman filter based on the target position of the current frame, and setting the current frame as the initial frame;

Step5: Returning to Step1 until the current frame is the last frame of the video, the algorithm ends.

4. Experiment

4.1. Experimental environment and data set

The experimental environment of this paper is Ubuntu14.04 LTS system, hardware configuration CPU is Intel i7-6700, GPU is 8G memory GTX980M. Based on the TensorFlow framework, it is developed using the Python language.

Using the image of the ship in the Fleetmon.com database as experimental data, six types of aircraft carriers, patrol ships, small frigates, container ships, passenger ships and fishing boats were manually extracted and labeled as training and test samples, including 380 training samples. There are total of 120 test samples, and each type of image resolution is above 200 × 200 pixels. Sample classification labels and partial image samples are shown in Table 1.

| Aircraft carrier | Destroyer | Small frigate | Container ship | Passenger ship | Fishing ship |
|------------------|-----------|---------------|----------------|----------------|--------------|

Table 1. Sample classification image

Taking the video stream data in the KITTI data set as the calculation framework verification set, KITTI is the largest computer vision algorithm evaluation data set of the automatic driving scene, including the video stream data collected by the scenes of urban, rural and highway, and the samples collected in this paper. The data is 1280 x 720 and the frame rate is 30 frames/second.

4.2. Experimental result

This section verifies the proposed algorithm by two experiments: 1) Verify the accuracy of identification of ship targets on the FleetMon dataset; 2) The algorithm identifies the tracking and tracking results of the moving targets in the video on the KITTI dataset.
4.2.1. Experiment 1. In order to verify the recognition accuracy of the algorithm on the ship target, the target detection part of the algorithm is first trained and tested. The number of training times is 1000 times, the learning rate is 0.02, and a total of 6 repeated experiments are performed. The final classification accuracy is shown in Table 2. According to the data in Table 2, the accuracy of the algorithm for the classification of the six types of ships is above 94%.

| Frequency | Aircraft carrier | Patrol ship | Small frigate | Container Ship | Passenger ship | Fishing boat |
|-----------|------------------|-------------|---------------|---------------|----------------|--------------|
| 1         | 98.33            | 97.14       | 93.58         | 96.28         | 93.33          | 91.43        |
| 2         | 97.62            | 98.15       | 97.63         | 93.65         | 97.52          | 97.63        |
| 3         | 98.14            | 97.26       | 95.40         | 98.30         | 93.54          | 97.22        |
| 4         | 98.02            | 97.65       | 98.30         | 98.14         | 96.23          | 91.74        |
| 5         | 95.93            | 96.84       | 93.48         | 97.21         | 93.48          | 93.45        |
| 6         | 97.35            | 95.47       | 93.55         | 98.30         | 94.74          | 98.07        |
| Average   | 97.57            | 97.09       | 95.32         | 96.98         | 94.81          | 94.92        |

4.2.2. Experiment 2. Due to the lack of data flow targeted by the ship as experimental data, the KITTI data set verification algorithm is used to identify and track the moving targets in the video. Figure 5 is a cumulative distribution function of the detection time and tracking time of a single-frame image. It can be seen that about 80% of the target detection time of a single-frame image falls within the interval [0.09, 0.11], and the target tracking time of a single-frame image falls within the interval [0.003, 0.025] range.

![Figure 5](image)

**Figure 5.** The detection time and tracking time of a single-frame image

Table 3 is the result of intercepting part of the target detection and tracking. Faster R-CNN has outstanding performance advantages in target detection, and the combination of Camshift and Kalman filter can effectively deal with environmental changes such as illumination, perspective and background during target tracking.

| Frequency | Aircraft carrier | Patrol ship | Small frigate | Container Ship | Passenger ship | Fishing boat |
|-----------|------------------|-------------|---------------|---------------|----------------|--------------|
| 1         | 98.33            | 97.14       | 93.58         | 96.28         | 93.33          | 91.43        |
| 2         | 97.62            | 98.15       | 97.63         | 93.65         | 97.52          | 97.63        |
| 3         | 98.14            | 97.26       | 95.40         | 98.30         | 93.54          | 97.22        |
| 4         | 98.02            | 97.65       | 98.30         | 98.14         | 96.23          | 91.74        |
| 5         | 95.93            | 96.84       | 93.48         | 97.21         | 93.48          | 93.45        |
| 6         | 97.35            | 95.47       | 93.55         | 98.30         | 94.74          | 98.07        |
| Average   | 97.57            | 97.09       | 95.32         | 96.98         | 94.81          | 94.92        |

5. Conclusion
This paper proposes a video-based ship target recognition method, which can effectively track moving targets based on the identification of ship targets. Shipborne tracking radar parasitic TV can obtain clear images of sea surface targets in sea and sky background. In order to improve the combat capability of ship combat systems, this paper uses hand-labeled TV video as research object, and uses Camshift and Kalman filter to improve Faster R-CNN calculation. The framework is used to automatically identify and track the ship image target in the air and sea background, and the
effectiveness of the algorithm is verified by experiments. The correct recognition rate of the ship target model based on this method is above 90%.

References
[1] Jianwei Li, Changwen Qu and Shujuan Peng. Ship detection in SAR images based on an improved Faster R-CNN[C]. 2017BIGSARDATA, Beijing.
[2] Chen Wenting. Research on feature extraction and classification and recognition of SAR image ship target [D]. National University of Defense Technology, 2012.
[3] Liang Jinxiong, Wang Qiqi. Classification of ship target recognition based on BP neural network[J]. Ship Science and Technology, 2015, 37(3): 206–209.
[4] Zhai Zhong, Liu Qing, NieLinhong, et al. Research on texture classification based on convolutional neural network[J]. Journal of Frontiers of Computer Science and Technology, 2016, 10(3): 389–397.
[5] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6):1137-1149.
[6] R. Girshick. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, pages 1440–1448, 2015.
[7] Zhang C, Qiao Y, Fallon E and Xu C 2009 An improved camshift algorithm for target trackingin video surveillance. [C]. 9th IT&T Conference 1-10.
[8] Huang S and Hong J 2011 Moving object tracking system based on camshift and Kalman filter. [C]. IEEE Int. Conf. on Consumer Electronics, Communications and Networks 1423-6
[9] KITTI benchmark, http://www.cvlibs.net/datasets/kitti/eval_object.php