Postprint of Small and Weak Target Detection Technology in Complex Large-Scale Traffic Scenes

Authors: Huaxia, Wang Xinqing, Ma Zhaoye, Wang Dong, Shao Faming

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

Abstract

To address the issues of poor recognition performance for low-resolution small targets in high-resolution complex large-scale scenes by existing big data and deep learning-based object detection frameworks, and the difficulty in balancing accuracy and real-time performance for multi-target detection, we improved the deep learning-based object detection framework SSD (Single Shot MultiBox Detector) and proposed an enhanced multi-target detection framework called DRZ-SSD (DRZ), specifically tailored for multi-target detection in complex large-scale traffic scenarios. The detection follows a coarse-to-fine strategy: a low-resolution coarse detector and a high-resolution fine detector are trained separately. High-resolution images are downsampled to obtain low-resolution versions. We designed a Dynamic Region Zoom-in Network framework (DRZN) based on reinforcement learning that dynamically zooms in low-resolution weak target regions to high resolution for detection and recognition using the fine detector, while the remaining image regions are processed by the coarse detector. This approach yields significant improvements in detection and recognition accuracy for weak targets as well as computational efficiency. An adaptive threshold adjustment strategy using fuzzy thresholding is employed to enhance the model' s decision-making capability while avoiding overfitting to the dataset, significantly reducing both missed detection and false alarm rates. Experimental results demonstrate that the improved DRZ-SSD achieves favorable performance in challenging detection scenarios such as weak targets, multiple targets, cluttered backgrounds, and occlusions. Testing on designated datasets shows that compared to other deep learning-based object detection frameworks, the average accuracy for various target categories improved by 4-15%, the mean average precision (mAP) increased by approximately 9-16%, the multi-target detection rate improved by 13-34%, and the detection and recognition speed reached 38 frames/s, thereby achieving a balance between algorithmic accuracy and operational speed.

Full Text

Preamble

Detection of Dim and Small Targets in Complex Large Traffic Scenes

Hua Xia[1], *Wang Xinqing*[1], *Ma Zhaoye*[1], *Wang Dong*[1,2], *Shao Faming*[1]

[1]Army Engineering University, Nanjing 210007, China;
[2]Second Institute of Engineering Research & Design, Southern Theatre Command, Kunming 650222, China

Abstract: Existing target detection frameworks based on big data and deep learning exhibit poor recognition performance for low-resolution small targets in high-resolution complex large-scale scenes, and struggle to balance the accuracy and real-time performance of multi-target detection. This paper improves the deep learning-based Single Shot MultiBox Detector (SSD) framework and proposes an enhanced multi-target detection framework called DRZ-SSD (Dynamic Region Zoom-in SSD), specifically designed for multi-target detection in complex large traffic scenes. Detection proceeds via a coarse-to-fine strategy: we train a low-resolution coarse detector and a high-resolution fine detector separately, obtain a low-resolution version through downsampling of high-resolution images, and design a dynamic region zoom-in network framework based on reinforcement learning (DRZN) that dynamically zooms low-resolution dim target regions to high resolution for detection and recognition by the fine detector, while the remaining image regions are processed by the coarse detector. This approach significantly improves detection and recognition accuracy for dim targets while enhancing computational efficiency. We employ a fuzzy threshold method to adjust the adaptive threshold strategy, which improves model decision-making capability while avoiding overfitting to the dataset, substantially reducing both missed detection and false alarm rates. Experiments demonstrate that the improved DRZ-SSD achieves favorable results when handling challenging scenarios such as dim targets, multiple targets, cluttered backgrounds, and occlusions. Testing on designated datasets shows that compared to other deep learning-based target detection frameworks, the average precision for various target categories improves by 4–15%, mean average precision (mAP) improves by approximately 9–16%, multi-target detection rate increases by 13–34%, and detection/recognition speed reaches 38 frames/s, achieving a balance between algorithmic accuracy and runtime speed.

Keywords: machine vision; deep learning; neural network; traffic scene multi-target detection; reinforcement learning; self-adaptation

---

0 Introduction

Target detection and recognition of pedestrians and vehicles in traffic scenes represents an important branch of target detection technology, serving as a

core component in research domains such as autonomous driving, robotics, and intelligent video surveillance [1]. Deep learning constitutes a learning methodology based on deep artificial neural networks, and deep learning-based target detection algorithms can be applied to diverse detection scenarios with strong comprehensiveness, enabling simultaneous detection and recognition of multiple target categories with high initiative. Among various artificial neural network architectures, deep convolutional networks possess powerful feature extraction capabilities. Increasingly sophisticated network structures for image classification have been proposed, continuously enhancing the advantages of deep convolutional networks in feature extraction and achieving excellent performance in visual tasks including image recognition, image segmentation, target detection, and scene classification [2].

The Faster R-CNN [4] framework replaces the time-consuming selective search method, improving speed while maintaining high-quality region proposals from the RPN that enhance accuracy (mAP). However, region proposals generated at image edges are discarded. YOLO [5] formulates object detection as a regression problem, resulting in a simplified detection pipeline and enabling single-stage training. YOLO can "see"the entire image during training and inference, yielding low background false positive rates, whereas region proposal-based methods (e.g., Fast R-CNN) only "see" local information within candidate boxes during detection. Nevertheless, YOLO suffers from poor localization accuracy, low recall, and particularly subpar performance for small and densely packed targets.

SSD (Single Shot MultiBox Detector) [3], proposed by Liu Wei at ECCV 2016, remains one of the primary detection frameworks to date. Compared to Faster R-CNN [4], SSD offers significant speed advantages, while providing clear mean average precision (mAP) benefits over YOLO [5]. SSD' s main characteristics include: inheriting YOLO' s regression-based detection approach with single-stage training; adopting prior boxes similar to Faster R-CNN' s anchors; and incorporating pyramidal feature hierarchy-based detection [6], representing a partial FPN [6] concept. Although SSD has achieved high accuracy on specific datasets with good real-time performance, its training process remains extremely time-consuming and heavily dependent on training sample quality and quantity. Relying on image color and edge information for target detection, SSD exhibits poor performance for targets lacking sufficient image information, such as dim targets and heavily occluded objects. Furthermore, its detection efficiency requires improvement to meet real-time operational requirements for deployed systems.

This paper addresses the characteristics and requirements of pedestrian and vehicle target detection tasks in complex large traffic scenes by introducing two key improvements to the traditional SSD algorithm: (1) Utilizing reinforcement learning and sequential search methods tailored to large traffic scene target detection, we propose a Dynamic Region Zoom-in Network framework (DRZN). This framework significantly reduces computational load through image downsampling while maintaining detection accuracy for targets of various sizes in

high-resolution images via dynamic region zoom-in, markedly improving detection and recognition precision for low-resolution dim targets and reducing missed detection rates. (2) To address SSD' s inflexible fixed confidence threshold, we employ a fuzzy threshold method to adjust adaptive threshold strategies, enhancing model decision-making capability while avoiding dataset overfitting and significantly reducing both missed detection and false alarm rates.

1 Dynamic Region Zoom-in Network Framework

SSD employs a feature pyramid structure for detection, utilizing feature maps of different scales including conv4-3, conv-7 (FC7), conv6-2, conv7-2, conv8_2, conv9_2 for simultaneous softmax classification and localization regression across multiple feature maps, achieving reasonable detection accuracy for small targets [3]. However, its performance for low-resolution dim targets in complex large traffic scenes remains inadequate.

To address SSD's difficulties with low-resolution dim target detection in complex large scenes, this paper proposes a Dynamic Region Zoom-in Network framework (DRZN). This framework reduces target detection computational cost by downsampling high-resolution large-scene images while preserving detection accuracy for low-resolution dim targets through dynamic region zoom-in, significantly improving detection and recognition precision for dim targets. Detection proceeds via a coarse-to-fine strategy: first performing detection on downsampled image versions to reduce computational load and improve runtime efficiency, then sequentially selecting regions likely to contain low-resolution small targets for zoom-in operations and detailed analysis to ensure recognition accuracy for low-resolution small targets. Using reinforcement learning, we model the zoom-in reward from both detection accuracy and computational cost perspectives, dynamically selecting a series of regions to zoom to high resolution for analysis. The overall algorithmic framework is illustrated in Figure 1 [Figure 1: see original paper].

1.1 Amplification Precision Gain Regression Network R-net

Sequential Search. The strategy for processing high-resolution large-scene images avoids processing the entire image, instead sequentially detecting small regions suspected to contain targets.

Reinforcement Learning (RL). RL serves as a general mechanism for learning sequential search strategies, as it enables the model to consider the effects of a sequence of actions rather than just individual actions [8]. RL learns to select actions that yield maximum reward in specific contexts through trial-and-error. In many scenarios, current actions affect not only immediate rewards but also subsequent states and reward sequences. RL' s three most important characteristics are: it operates fundamentally in a closed-loop form; it does not directly specify which action to select; and a sequence of actions and reward signals produces long-term effects on subsequent actions. RL employs a sample-

acquisition-and-learning approach, updating its learning model after obtaining samples, using the current model to guide subsequent actions, and iterating this process until convergence.

Our algorithm adopts a coarse-to-fine detection strategy, applying a coarse detector at low resolution and using its output to guide deeper searches for high-resolution targets. While the coarse detector will be less accurate than the fine detector, it identifies image regions requiring further analysis, thereby incurring high-resolution detection computational costs only in promising regions. The algorithm primarily employs two mechanisms: (a) a mechanism for learning statistical relationships between coarse and fine detectors to predict which regions require zoom-in given coarse detector output; and (b) a mechanism for sequentially selecting regions for high-resolution analysis given coarse detector output and regions requiring fine detector analysis.

This strategy can be formulated as a Markov Decision Process. At each step, the system first observes the current state, estimates the potential cost-aware reward for different actions, and selects the action with the maximum long-term cost-aware reward [9]. The components include:

a) Actions. The algorithm sequentially analyzes regions with high zoom-in returns at high resolution. In this context, actions correspond to selecting regions for high-resolution analysis. Each action can be represented by a vector indicating region location and size. At each step, the algorithm scores a set of potential actions (a list of rectangular regions) based on potential long-term rewards.

b) State Space. The state representation encodes two types of information: predicted precision gains for regions to be analyzed, and the history of regions already analyzed at high resolution (the same region should not be zoomed multiple times). We design an Amplification Precision Gain Regression Network (R-net) to learn an information precision gain map (AG map) as the state representation. The AG map has the same width and height as the input image, where each pixel' s value estimates how much detection accuracy would improve by including that pixel in the input image. Thus, the AG map provides detection accuracy gains for selecting different actions. After taking an action, values corresponding to the selected region in the AG map are reduced accordingly, allowing the AG map to dynamically record action history.

c) Loss-Reward Function. The state encodes predicted precision gains for zooming each image subregion. To maintain high accuracy under limited computation, we define a loss-reward function as shown in Equation 1. Given a state and action, the loss-reward function scores each action (zoom region) by considering both cost increment and accuracy improvement:

$$R_{gp} = \sum_{k \in B_s} \left(\lambda \cdot (p_h^k - p_l^k) - \frac{|a|}{|A|} \right)$$

where action k indicates that target k is included in the region selected by the action. p_h^k and p_l^k represent the detection scores from the coarse and fine detectors for the same target, respectively, and p_g^k is the corresponding ground truth label. The variable $|a|$ denotes the total number of pixels in the selected region, while $|A|$ represents the total number of pixels in the input image. The first term represents detection accuracy improvement, and the second term represents zoom-in cost. The balance between accuracy and computation is controlled by parameter λ.

The Amplification Precision Gain Regression Network (R-Net) predicts specific zoom-in precision gains based on coarse detection results. R-Net is trained on coarse-fine detection data pairs so it can observe their correlations to learn appropriate precision gain relationships [7]. Due to SSD' s success in many computer vision applications, we use SSD as the base detector. Two SSD models are trained separately on training sets composed of high-resolution fine images and low-resolution coarse images, subsequently serving as black-box coarse and fine detectors. Applying the two pre-trained detectors to a set of training images yields two sets of detection results: low-resolution detections $\{d_l^i, p_l^i, f_l^i\}$ in downsampled images and high-resolution detections $\{d_h^j, p_h^j\}$ in each image' s high-resolution version, where d is the detection bounding box, p is the probability of being a target object, and f represents the feature vector of the corresponding detection. Superscripts h (High) and l (Low) denote high-resolution and low-resolution (downsampled) images, respectively.

To enable the model to determine whether high-resolution detection improves overall results, we introduce a matching layer to associate detection results from both detectors. In this layer, if possible object i in the downsampled image and possible object j in the high-resolution image have sufficiently large intersection over union ($IoU(d_l^i, d_h^j) > 0.5$), we define i and j as corresponding to each other. Following rules to match coarse and fine detection proposals generates a set of correspondences $\{(d_l^k, p_l^k, f_l^k, d_h^k, p_h^k)\}$. Given this correspondence set, we can estimate the zoom-in precision gain for coarse detection.

Detectors can only process objects within a certain size range, so applying the detector to high-resolution images does not always yield optimal accuracy. For example, if a detector is primarily trained on small target datasets, its detection accuracy for larger targets will not be high. Therefore, we use $|p_g^k - p_h^k| < |p_g^k - p_l^k|$ to measure which detection result (coarse or fine) is closer to ground truth, where p_g^k serves as the ground truth metric. When the high-resolution score p_h^k is closer to ground truth than the low-resolution score p_l^k, this function indicates the target is worth zooming in; otherwise, applying the coarse detector on the downsampled image may produce higher accuracy, and we should avoid zooming in on that target.

We introduce a Correlation Regression (CR) layer to estimate the zoom-in precision gain g_k for target k:

$$g_k = \phi_1(W_1, f_l^k)$$

where ϕ represents the regression function and W_1 represents the parameter set. This layer' s output is the estimated accuracy gain. The CR layer comprises two fully connected layers: the first with 4096 units and the second with a single output unit.

Based on the learned accuracy gain for each target, we can generate the AG map (accuracy gain map). Assuming each pixel within a candidate bounding box contributes equally to its accuracy gain, the AG map is generated as:

$$\widehat{AG}(x,y) = \begin{cases} \phi_2(W_2, f_l^k) & \text{if } (x,y) \in b_l^k \\ 0 & \text{otherwise} \end{cases}$$

where α is a constant and W_2 represents the estimated parameters of the CR layer.

The AG map serves as the state representation, naturally containing information about coarse detection quality. After zooming in and detecting a region, all values within that region are set to 0 to prevent future zooming in the same area. The structure of the Amplification Precision Gain Regression Network R-net is shown in Figure 2 [Figure 2: see original paper].

1.2 Dynamic Amplification Region Selection Algorithm

To reduce computational cost for region zoom-in and fine detection, effectively improving algorithm efficiency and real-time performance while ensuring good coverage of selected regions, we construct amplification screening regions centered on each region center block comprising 3$×$3 rectangles. If multiple rectangular regions within the same amplification screening region satisfy the pixel value threshold condition, we select the one with the maximum pixel value as the region center.

The region center block selection process is as follows: First, the AG map is divided into equal rectangular regions using an 8$×8$grid.We calculate the sum of pixel values in each rectangle, set a amplification screening region.

Within the amplification screening region, we construct 4 prediction bounding boxes with different aspect ratios centered at the amplification screening region center point. By comparing construction metrics (pixel values, ratios, areas) across different prediction bounding boxes, we select the optimal amplification region bounding box. The total pixel value $sumpx_i$ of rectangle rtg_i in the gridded AG map is:

$$sumpx_i = \sum_{j \in rtg_i} px_j$$

where px_j represents the pixel value of the j-th pixel point in region rtg_i. A larger $sumpx_i$ value indicates greater zoom-in benefit for rectangle rtg_i, which aligns with human visual perception of block neighborhood correlation. We adaptively select the pixel value threshold using a second-order difference method for initial region center block screening. The second-order difference can represent the magnitude of change trends in discrete arrays, useful for determining thresholds within a set of pixel values. Detecting an AG map yields 64 candidate regions by default, with each candidate region obtaining one overall pixel value $sumpx_i$ representing zoom-in benefit, resulting in a 64$×$1 array. We discard elements smaller than 0.1 (judged as containing no target), obtaining an $n \times 1$ array C. Let $f(g)$ be the function estimating the decreasing trend of $sumpx_i$:

$$f(C_k) = C_{k+1} + C_{k-1} - 2C_k, \quad k = 1, 2, 3, \ldots, n-1$$

We select C_k that maximizes $f(C_k)$ as the $sumpx_i$ threshold for this AG map image.

After obtaining the AG map, each pixel' s value estimates how much detection accuracy would improve by including that pixel in the input image. Thus, the AG map provides detection precision gains for selecting different actions. After taking an action, values corresponding to the selected region in the AG map are reduced accordingly, allowing the AG map to dynamically record action history. Based on the AG map, we propose a dynamic amplification region selection algorithm, with the specific algorithm flow shown in Figure 3 [Figure 3: see original paper].

Within the amplification screening region, centered at the amplification screening region center point, we predict 6 fixed-size prediction bounding boxes according to different aspect ratios. The area of the amplification screening region is S_Z. The area of each prediction bounding box is calculated as:

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m-1} \cdot (k-1), \quad k = 1, 2, \ldots, 5$$

where $s_{\min} = 0.1 \times S_Z$, $s_{\max} = 0.7 \times S_Z$, and $m = 5$. Different aspect ratios are assigned to different prediction bounding boxes:

$$\left\{ \frac{w_r}{h_r} = a_r, a_r \in \left\{ \frac{1}{2}, 1, 2, 3 \right\} \right\}$$

where w_r and h_r represent the width and height of the bounding box, respectively. The corresponding width and height of the prediction bounding box are:

$$w_r = \sqrt{s_k / a_r}, \quad h_r = \sqrt{s_k \cdot a_r}$$

When $a_r = 1$, there is an additional prediction bounding box with scale $s'_k = \sqrt{s_k \cdot s_{k+1}}$, resulting in a total of 6 prediction bounding boxes.

For any bounding box b_l, we calculate the total pixel value within the box:

$$sumpx_i = \sum_{j \in b_l} px_j$$

The region area S is:

$$S_b = W \times L$$

where W and L represent the width and length of the box, respectively. The proportion of high zoom-in benefit pixels within the region is:

$$P_n = \frac{pn_1}{pn}$$

where pn_1 represents the total number of pixels with zoom-in benefit (pixel values greater than 0.1) in box b_l, and pn represents the total number of pixels in box b_l region.

Each prediction bounding box b_l has a feature vector $(x, y, sumpx, W, L, P)$, where x and y represent the horizontal and vertical coordinates of b_l' s center point.

Using manually annotated training samples, we train a Logistic classifier [10] to evaluate the box selection effectiveness of each prediction bounding box. The evaluation results are divided into two categories: prediction bounding boxes that meet zoom-in requirements and those that do not. After evaluating each prediction bounding box via the Logistic classifier, we obtain a corresponding box selection evaluation score, followed by non-maximum suppression to obtain the final prediction as the ultimate amplification bounding box.

After selecting the amplification bounding box, we set all pixel values within the amplification screening region to 0 to avoid efficiency loss from repeated selection, simultaneously updating the corresponding region in the AG map. We then check whether all high zoom-in benefit regions have been detected on the AG map (whether the total pixel value of the AG map is 0). If so, detection is complete; otherwise, the detection process continues iteratively.

Before sending the original image portion of the amplified fine detection candidate region to the fine detector, we first perform bilinear interpolation zoom-in to the minimum size required by the fine detector for candidate regions (the minimum candidate region size is set to 10$\times$10 in this paper).

2 Confidence Adaptive Threshold Improvement

In SSD' s final classification stage using Softmax, candidate regions receive confidence scores for each category (i.e., probability of belonging to each category). When the confidence for a particular category exceeds a set threshold, the candidate region is classified as that target type. If multiple categories exceed the threshold for the same candidate region, the category with the highest confidence is selected. When target scale is small or occluded, confidence is relatively low. Using a fixed threshold, setting it too high excludes many true targets, while setting it too low introduces many false targets. The common practice involves repeatedly adjusting thresholds and performing multiple tests on the dataset to calculate average precision at different thresholds, selecting the threshold with maximum average precision as the final model threshold. However, this approach tends to overfit to the dataset, and even large datasets cannot cover all real-world scenarios. This paper employs adaptive thresholds to improve model decision-making capability while avoiding dataset overfitting [11].

Whether fixed or adaptive, threshold setting requires reference to target score distributions in the dataset. With well-trained detection models, confidence scores for correct detection results often differ by one or two orders of magnitude between true and false targets, with true target confidence typically above 0.7. Although false targets differ from true targets in confidence, they may still achieve high confidence above 0.7 due to certain target-like features, making pure fixed thresholds unable to distinguish targets from background [11]. To address SSD' s inflexible fixed confidence threshold, we adopt a fuzzy adaptive threshold method [12] to adjust the adaptive threshold strategy, reducing missed detection and false alarm rates.

Fuzziness is determined by a fuzzy rate function; segmentation achieves optimal effect when the fuzzy rate is minimized. The fuzzy rate relates to membership functions, with the fundamental idea of fuzzy mathematics being the concept of membership degree. The key to applying fuzzy mathematics for modeling lies in establishing membership functions that conform to reality [12].

Based on the formula, we obtain parameter θ using gradient descent. First, we differentiate parameter θ:

$$\frac{\partial}{\partial \theta_j} l(\theta) = \sum_{i=1}^{n} \left(y^{(i)} - h_\theta(b_l^{(i)}) \right) \cdot b_{l_j}^{(i)}$$

The parameter update formula is:

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{n} \left(y^{(i)} - h_\theta(b_l^{(i)}) \right) \cdot b_{l_j}^{(i)}$$

where α is the learning rate.

SSD detects N candidate regions per image by default, with each candidate region receiving M confidence scores representing belonging to M categories, resulting in N arrays of size $M \times 1$. We extract the maximum value from each array and sort them in descending order, discarding values smaller than 0.1 (if all N values are below 0.1, we judge no target present), obtaining an $N \times 1$ array C.

Let $\mu(C_k)$ be the membership degree of regions with confidence C_k in array C. The fuzzy rate $\gamma(C)$ of array C is a measure of array C' s fuzziness. Let $h(C_k)$ be the number of elements with confidence C_k in array C. The fuzzy rate $\gamma(C)$ of array C is defined as:

$$\gamma(C) = \frac{2}{n} \sum_{k=1}^{n} \min\left(\mu(C_k), 1 - \mu(C_k)\right)$$

The fuzzy rate $\gamma(C)$ depends on the membership function $\mu(x)$. If we select the membership function as an S-function:

$$\mu(x) = \begin{cases} 0, & 0 \le x \le q_0 - \Delta \\ 2\left(\frac{x - q_0 + \Delta}{2\Delta}\right)^2, & q_0 - \Delta < x \le q_0 \\ 1 - 2\left(\frac{x - q_0 + \Delta}{2\Delta}\right)^2, & q_0 < x \le q_0 + \Delta \\ 1, & q_0 + \Delta < x \le C_n \end{cases}$$

where parameter q determines the threshold. Once the coefficient c is selected (commonly set to 0.3 based on previous research), the threshold depends only on parameter q. The fuzzy threshold method solves for the adaptive threshold by pre-setting window width, calculating the fuzzy rate curve by varying q to make the membership function $\mu(x)$ pass through the confidence interval $[0, 1]$. The valley point of this curve, which minimizes $\gamma(C)$, corresponds to the desired adaptive threshold.

The overall improved detection algorithm framework is shown in Figure 4 [Figure 4: see original paper]. The algorithmic flow is as follows: (a) Input a single-frame image from the video to be detected, downsample the image to obtain a low-resolution version to reduce computational load; (b) Through DRZN, sequentially select and zoom in regions requiring high-resolution detection and recognition, perform target detection and recognition with the fine detector to obtain results R_1, and set all pixel values to 0 in already-detected regions of the low-resolution image; (c) Input the remaining low-resolution image to the coarse detector for target detection and recognition to obtain results R_2; (d) Fuse detection results R_1 and R_2 to obtain final results R_3.

3 Experiments

3.1 Experimental Conditions and Datasets

Our experiments use a DELL Precision R7910 (AWR7910) graphics workstation with an Intel Xeon E5-2603 v2 (1.8 GHz/10M) processor and NVIDIA Quadro K620 GPU for accelerated computation. SSD runs on the Caffe deep learning framework, which supports CPU and GPU parallel computation, enabling deep learning with massive computational requirements to be completed within a short period.

We conduct experiments on traffic scene datasets collected from YFCC100M (Web dataset) and the KITTI dataset. For KITTI, we select the first image set "Download left color images of object data set" and annotation file "Download training labels of object data set," containing 7,481 training images. KITTI images have annotation information, while test images do not. SSD training scripts are based on the Pascal VOC dataset format, requiring conversion of KITTI dataset to Pascal VOC format. Pascal VOC contains 20 categories total; we configure 3 categories for our dataset: 'Car' , 'Cyclist' , and 'Pedestrian' . Since the original annotations include other vehicle and person types, we merge 'Van' , 'Truck' , and 'Tram' into the 'Car' category, merge 'Person_{sitting}' into 'Pedestrian' , and directly ignore 'Misc' and 'Dontcare' categories. We select 100 images containing low-resolution small targets (defined as targets smaller than 10$×$10 pixels) from the test set to construct the KITTI low-resolution small target test set.

The YFCC100M dataset contains nearly 100 million images along with summaries, titles, and tags. To better demonstrate our method, we collect 1,000 high-resolution test images from YFCC100M by searching keywords "pedestrian" , "road" , and "vehicle" . For this dataset, we annotate all targets with at least 16-pixel width and less than 50% occlusion. Images are rescaled on the longer side to 2,000 pixels to fit GPU memory. We select 100 images containing low-resolution small targets (targets smaller than $10\$\times 10 pixels) from the test set to construct the Web dataset low-resolution small target test set. In experiments, all image sizes are normalized to $320\times\$320$.

3.2 Experimental Parameter Settings

We select SSD512 from the SSD family for improvement. SSD512 provides large, medium, and small-scale deep convolutional neural network models; we choose the medium-scale VGG_{CNN}M_{1024} model as the base model, modifying parameters related to target category count (the original model recognizes 20 target categories while ours recognizes only 3). Small sample datasets can represent original datasets to some extent, and optimal hyperparameters obtained through small sample training can adapt to original datasets to a certain degree [13]. Through small sample parameter tuning, without using adaptive thresholds, the threshold is set to 0.1 (default is 0.7). The number of candidate regions retained after non-maximum suppression is set to 100 (default is

300) across all experiments. Other settings remain at default values, and all subsequent experiments are conducted based on these settings.

3.3 Evaluation Metrics

In multi-target classification, let n be the number of target categories. For single-category discrimination, we still follow the four possibilities of binary hypothesis testing, where each hypothesis has two outcomes. Let H_0^j represent the null hypothesis that target j is absent, and H_1^j represent the alternative hypothesis that target j is present. Let D_0^j represent the decision that target j is absent, and D_1^j represent the decision that target j is present. In any binary hypothesis testing problem, we consider four possibilities [14]: (a) H_0^j true, decide D_0^j; (b) H_0^j true, decide D_1^j; (c) H_1^j true, decide D_0^j; (d) H_1^j true, decide D_1^j.

Cases (a) and (d) represent correct target j decisions; (b) is a Type I error called false alarm (no target detected as present); (c) is a Type II error called missed detection (present target missed). Additionally, in multi-target recognition, misclassifying target i as target j ($i \neq j$) constitutes a classification error.

In target classification, we focus on recognition performance for present targets, where recognition rate generally refers to detection rate. By definition, false alarm rate, detection rate, missed detection rate, and misclassification rate sum to 1. In practical calculations, we first compute recognition rate, then calculate false alarm rate and missed detection rate. For remaining targets identified by the system that do not actually exist, we count them to compute classification false alarm rate. For multi-target recognition, false alarm rate should be calculated as accumulated false alarm rate over a certain time period. For datasets, we use averaging to compute overall false alarm rate, missed detection rate, detection rate, and misclassification rate.

Deep learning adjusts neural network weights through backpropagation of errors to achieve modeling. Backpropagation iterations gradually increase from tens of thousands to hundreds of thousands until training error converges. Model quality is finally evaluated by calculating average precision (AP) and mean average precision (mAP) on the test set. AP measures detection algorithm accuracy from both recall and precision perspectives. AP is the most intuitive standard for evaluating deep detection model accuracy and can analyze detection effectiveness for individual categories. mAP is the average of AP across all categories; higher mAP indicates better comprehensive detection performance across all categories [11].

3.4 Experimental Design

We first combine each strategy individually with SSD512 for comparative experiments to demonstrate each strategy' s effect, then combine all strategies with SSD512 for comprehensive evaluation of the final improved algorithm.

We train the original SSD512 on the training set, denoting this model as M0. Based on M0, we add the adaptive threshold strategy to generate model M1. Based on M0, we add the dynamic local region zoom-in strategy to generate model M2. Finally, we combine M0 with all strategies to generate model M3. We test and compare M0, M1, and M3 using test sets from both databases. To highlight low-resolution small target detection performance, we separately test and compare M0 and M2 using the constructed small target test sets.

Experimental results are shown in Tables 1 and 2, comparing recognition and detection performance of models M0, M1, and M3 on regular test sets from KITTI and WD datasets.

Table 1 Comparison of Recognition Precision (AP %)

Model	Dataset	mAP (%)	Person	Cyclist	Car
M0	KITTI				
M1	KITTI				
M3	KITTI				

Table 2 Comparison of Detection Performance

Model	Dataset	Pf (%)	Pm (%)	Pd (%)	Pe (%)
M0	KITTI				
M1	KITTI				
M3	KITTI				

Comparing M0 and M3 results in Tables 1 and 2, on the KITTI dataset, AP for various target categories improves by 14–19%, mAP improves by approximately 16.11%, false alarm rate decreases by 13.88%, detection rate increases by 32.13%, missed detection rate decreases by 10.65%, and misclassification rate decreases by 7.6%. On the WD dataset, AP improves by 7–11%, mAP improves by approximately 7.24%, false alarm rate decreases by 10.01%, detection rate increases by 31.33%, missed detection rate decreases by 11.19%, and misclassification rate decreases by 10.13%. All metrics show significant improvement, demonstrating the overall effectiveness of our strategy in compensating for SSD512' s deficiencies.

Comparing M0 and M1 results, on the KITTI dataset, AP improves by 1–4%, mAP improves by approximately 2.67%, false alarm rate decreases by 7.90%, detection rate increases by 16.52%, missed detection rate decreases by 6.05%, and misclassification rate decreases by 2.57%. On the WD dataset, AP improves by 1–3%, mAP improves by approximately 1.62%, false alarm rate decreases by 4.08%, detection rate increases by 13.62%, missed detection rate decreases by 6.89%, and misclassification rate decreases by 2.65%. Model M1 is trained by

adding the adaptive threshold strategy to M0. Comparing M1 with M0 on both databases reveals that M1 significantly improves multi-target detection rate and noticeably reduces false alarm and missed detection rates, demonstrating that the adaptive threshold strategy effectively distinguishes low-confidence true targets from high-confidence false targets, successfully reducing SSD512's missed detection and false alarm rates for multi-target detection.

Table 3 and Table 4 compare detection performance of models M0 and M2 on low-resolution small target test sets from KITTI and WD datasets.

Table 3 M0 and M2 Low-Resolution Small Target Recognition Precision (AP %)

Model	Dataset	Person	Cyclist	Car	mAP (%)
M0	KITTI				
M2	KITTI				

Table 4 M0 and M2 Low-Resolution Small Target Detection Performance

Model	Dataset	Pf (%)	Pm (%)	Pd (%)	Pe (%)
M0	KITTI				
M2	KITTI				

Comparing M0 and M2 results in Tables 3 and 4, on the KITTI dataset, AP improves by 49–64%, mAP improves by approximately 57.86%, false alarm rate decreases by 22.3%, detection rate increases by 50.34%, missed detection rate decreases by 19.26%, and misclassification rate decreases by 8.78%. On the WD dataset, AP improves by 44–57%, mAP improves by approximately 51.68%, false alarm rate decreases by 22.24%, detection rate increases by 45.58%, missed detection rate decreases by 15.63%, and misclassification rate decreases by 6.71%. Model M2 is trained by adding the dynamic local region zoom-in strategy to M0. Comparing M2 with M0 on low-resolution small target test sets from both databases reveals that M2 significantly improves recognition precision and detection rate for multi-target low-resolution small targets, with noticeable reductions in misclassification, false alarm, and missed detection rates. This demonstrates the effectiveness of the dynamic local region zoom-in strategy for low-resolution small target detection and recognition. Since low-resolution dim targets are difficult to classify, M2' s detection errors are mostly classification errors (high misclassification rate), while M0' s multi-target detection rate is extremely low, indicating that SSD512' s deep convolutional network causes severe information loss for low-resolution dim targets during hierarchical feature extraction.

Figure 5 [Figure 5: see original paper] validates the effectiveness of R-Net' s gain evaluation in model M3. The blue numbers indicate the confidence that the red box contains a target. c represents coarse detector results, F represents fine detector results, and red numbers represent R-Net' s precision gain. Positive and negative values are normalized to [0,1] and [-1,0), respectively. Comparison reveals that for regions where coarse detection is sufficiently good or better than fine detection, R-Net gives low precision gain scores (columns 1 and 2), while for regions where fine detection is much better than coarse detection (column 3), R-Net gives high precision gain scores.

We test using regular test sets from the Web Dataset and KITTI dataset. Detection and recognition performance is shown in Table 5 , where FPS represents algorithm runtime speed.

Table 5 Comparison of Detection and Recognition Performance Across Algorithms

Method	Dataset	mAP (%)	Person AP (%)	Car AP (%)	Cyclist AP (%)	Pd (%)	FPS
Faster R-CNN	KITTI	83.26	74.13	75.42			
DSOD300	KITTI	77.43	72.26	68.38			
DSSD513	KITTI	75.46	69.53	68.34			
YOLOv2 544	KITTI	79.43	71.25	67.32			
M3 (DRZ-SSD)	KITTI	87.42	86.73	84.38		38	

Comparing M3 with other deep learning-based detection algorithms in Table 5, on the KITTI dataset, AP for various target categories improves by 4–16%, mAP improves by approximately 9–15%, and detection rate increases by 13–28%. On the WD dataset, AP improves by 5–12%, mAP improves by approximately 4–9%, and detection rate increases by 10–34%. Although detection and recognition speed does not match that of DSOD300, DSSD513, or YOLOv2 544, the FPS reaches 38 frames/s, meeting real-time requirements.

4 Conclusion

To address the problems of poor recognition performance for low-resolution small targets in high-resolution complex large-scale scenes and the difficulty in balancing accuracy and real-time performance in multi-target detection for existing big data and deep learning-based target detection frameworks, this paper improves the deep learning-based SSD target detection framework and proposes an enhanced multi-target detection framework called DRZ-SSD, specifically designed for multi-target detection in complex large traffic scenes. Experimental validation demonstrates that the improved strategy effectively compensates

for traditional SSD' s deficiencies, achieving favorable results when handling challenging detection scenarios such as dim targets, multiple targets, cluttered backgrounds, and occlusions, while balancing algorithmic accuracy and runtime speed. Since convolutional neural networks are not well-suited for processing temporal information, combining them with recurrent neural networks [21] (a class of neural networks with memory capabilities) to address video target detection and tracking problems will be the focus of future work.

References

[1] Chi Xiaojun, Meng Qingchun, Chen Peng. Application of Bayesian decision-making method based on minimum risk in traffic detection [J]. Application Research of Computers, 2005, 22(12): 204-205.

[2] Yu Kai, Jia Lei, Chen Yuqiang. Yesterday, Today and Tomorrow of Deep Learning [J]. Computer Research and Development, 2013, 50(9): 1799-1804.

[3] Liu Wei, et al. SSD: Single Shot MultiBox Detector [C]// Proc of European Conference on Computer Vision. Cham: Springer, 2016: 21-37.

[4] Ren Shaoqing, et al. Faster R-CNN: towards real-time object detection with region proposal networks [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2017, 39(6): 1137-1149.

[5] Joseph R, et al. You only look once: unified, real-time object detection [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2016: 779-788.

[6] Lin Tsuang Yi, et al. Feature Pyramid Networks for Object Detection [J]. arXiv preprint arXiv: 1612.03144, 2016.

[7] Gao Mingfei, et al. Dynamic Zoom-in Network for Fast Object Detection in Large Images [J]. arXiv preprint arXiv: 1711.05187, 2017.

[8] Glen Berseth, Cheng Xie, et al. Progressive Reinforcement Learning with Distillation for Multi-Skilled Motion Control [J]. arXiv preprint arXiv: 1802.04765, 2018.

[9] Chen Qianbin, He Xiaoqiang, Wu Pan, et al. A strategy for determining the sleep duration of micro base stations based on service awareness in partially measurable Markov decision processes [J]. Journal of Electronics and Information Technology, 2018, 40(1): 130-136.

[10] Zhao Peng, Li Dazhai, Wang Wei. Part Image Region Extraction Based on Logistic Regression [J]. Application Research of Computers, 2017, 34(4): 1265-1268.

[11] Feng Xiaoyu, Mei Wei, Hu Dashuai. Aerial target detection based on improved faster R-CNN [J]. Acta Optica Sinica, 2018, 38(6): 1-16.

[12] Chen Guo, Zuo Hongfu. Adaptive fuzzy threshold segmentation method for images [J]. Acta Automatica Sinica, 2003, 29(5): 791-796.

[13] Hu Cong, Qu Wei, Xu Chuanpei, et al. Apparel image recognition based on adaptive pooling neural network [J//OL]. Computer application, 2018, 1-8.

[14] Ma Chunting, Zheng Jian, Chen Donggen, et al. Evaluation index of multi-target recognition for ground battlefield reconnaissance system [J]. Journal of Detection & Control, 2006, 28(1): 6-9.

[15] Shen Zhiqiang, et al. DSOD: learning deeply supervised object detectors from scratch [C]// Proc of IEEE International Conference on Computer Vision. Washington DC: IEEE Computer Society, 2017: 1937-1945.

[16] Zhang Jianming, et al. A real-time Chinese traffic sign detection algorithm based on modified YOLOv2 [J]. Algorithms, 2017, 10(4): 127.

[17] Fu Chengyang, Liu Wei, Ranga A et al. DSSD: deconvolutional single shot detector [J]. arXiv preprint arXiv: 1705.09587, 2017.

[18] Zhou Feiyan, Jin Linpeng, Dong Jun. A review of convolutional neural networks [J]. Chinese Journal of Computers, 2017, 40(6): 1229-1251.

[19] Chang Liang, Deng Xiaoming, Zhou Mingquan, et al. Convolutional neural networks in image comprehension [J]. Acta Automatica Sinica, 2016, 42(9): 1300-1312.

[20] Tang Pengjie, Wang Hanli, Kwong S. G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition [J]. Neurocomputing, 2017, 225(2): 188-197.

[21] Munaro Matteo, et al. OpenPTrack: Open source multi-camera calibration and people tracking for RGB-D camera networks [J]. Robotics & Autonomous Systems, 2016, 75: 525-538.

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