Random Access Memories: A New Paradigm for Target Detection in High Resolution Aerial Remote Sensing Images

Zhengxia Zou and Zhenwei Shi*, *Member IEEE*

***Abstract*—We propose a new paradigm for target detection in high resolution aerial remote sensing images under small target priors. Previous remote sensing target detection methods frame the detection as learning of detection model + inference of class-label and bounding-box coordinates. Instead, we formulate it from a Bayesian view that at inference stage, the detection model is adaptively updated to maximize its posterior that is determined by both training and observation. We call this paradigm "Random Access Memories (RAM)". In this paradigm, "Memories" can be interpreted as any model distribution learned from training data and "Random Access" means accessing memories and randomly adjusting the model at detection phase to obtain better adaptivity to any unseen distribution of test data. By leveraging some latest detection techniques e.g. deep Convolutional Neural Networks and multi-scale anchors, experimental results on a public remote sensing target detection dataset show our method outperforms several other state of the art methods. We also introduce a new dataset "LEVIR", which is one order of magnitude larger than other datasets of this field. LEVIR consists of a large set of Google Earth images, with over 22k images and 10k independently labeled targets. RAM gives noticeable upgrade of accuracy (an mean average precision improvement of 1%∼4%) of our baseline detectors with acceptable computational overhead.**

***Index Terms*—High resolution aerial remote sensing image, Target detection, Convolutional neural networks, Random access memories.**

I. Introduction

THE rapid development of remote sensing technologies has opened a door for people to observe the earth. Automatically detecting targets of remote sensing images, e.g. the airplane, oilpot and ship, is one of the core tasks in remote sensing applications, and have been drawing more and more attentions in recent years [1].

Most of the early attempts of remote sensing target detection [2–6] are designed with the help of some specifically designed hand-crafted features and supervised classification algorithms. Recent advances in remote sensing target detection methods [7–13] have been primarily focusing on deep learning based detection methods, especially the ones based on convolutional neural networks. Despite their great variations, all previous methods frame the detection as two stages: 1) learning a detection model $\mathcal{H}$ from training data $\mathcal{D}_{tr}$ by finding local/global maximum of likelihood $p(\mathcal{D}_{tr}|\mathcal{H})$, and 2) making inference of category labels and bounding-box coordinates from the observed image $\mathcal{D}_{ob}$. Arguably, once the learning process has stopped, the model will be fixed at testing time. In this paper, we will re-examine this problem from another perspective.

Random Access Memories (RAM). Here we re-think the detection paradigm by assuming that detection model can be changed adaptively as the model receives different observations. This idea is inspired by a large group of detection algorithms of signal process field called the Constant False Alarm Rate algorithms [14] where the algorithm adapts the parameters of the model to the statistical characteristics of the observation at test time. Similar thoughts can also be found in quantum mechanics where the state of any quantum object not only depends on its states, but also on the measurement itself [15]. Based on the above ideas, we formulate the detection model as any certain probability distribution $p(\mathcal{H})$ in its hypothesis space at training phase, and at detection phase the model is further updated that is determined by both of the training and observation. We call this new paradigm "Random Access Memories". From a Bayesian perspective, "Memories" can be interpreted as any model distribution $p(\mathcal{H})$ learned from training data, while "Random Access" may be interpreted as accessing to its memories and reaching its maximum of posterior after a random observation at detection phase. Fig. 1 shows the overview of our proposed new paradigm.

Approximate Inference. Most of the CNN based detectors are built on basis of frequentist statistics, where their detection model can be explicitly determined by the local/global maximum of the likelihood during the training phase. Instead, the proposed paradigm is established based on a complete probability distribution of the detection model. In Bayesian machine learning, Laplace Approximation [16, 17] acts as a simple but useful tool to find a Gaussian approximation of the posterior distribution near the MAP solution. In information theory, Fisher Information [18] has the same essence but different interpretations. It describes the amount of information obtained by the Maximum Likelihood Estimation (MLE) of a set of training samples, namely, a measure of the flatness and sharpness of the the model's distribution at MLE solution $\hat{\mathcal{H}}$.

The work was supported by the National Natural Science Foundation of China under the Grants 61671037, the Beijing Natural Science Foundation under the Grant 4152031, the funding project of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University under the Grant BUAA-VR-16ZZ-03 and the Excellence Foundation of BUAA for PhD Students under the Grant 2017056. *(Corresponding author: Zhenwei Shi.)*

Zhengxia Zou (e-mail: zhengxiazou@buaa.edu.cn) and Zhenwei Shi (Corresponding Author, e-mail: shizhenwei@buaa.edu.cn) are with Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China, and with Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, China, and also with State Key Laboratory of Virtual Reality Technology and Systems, School of Astronautics, Beihang University, Beijing 100191, China.

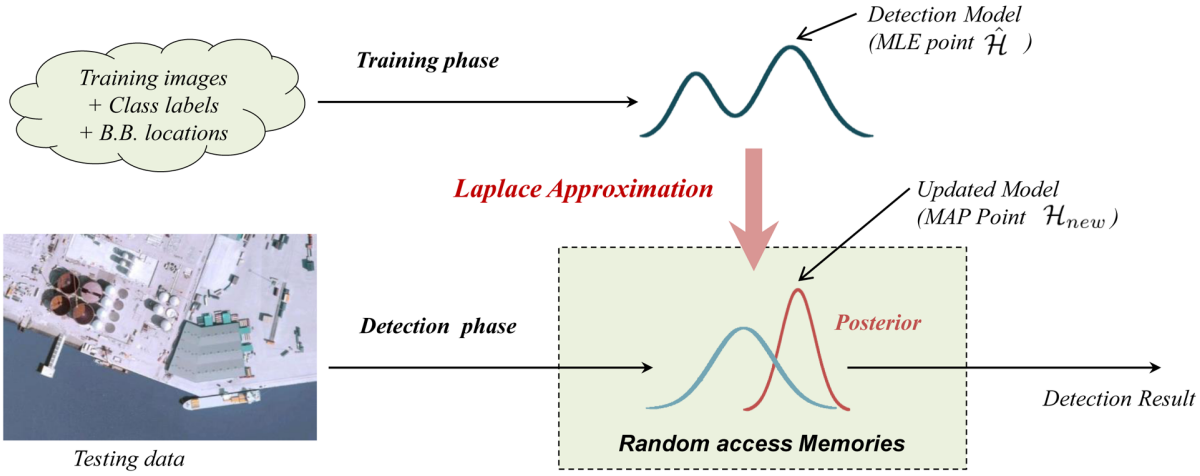

Fig. 1. An overview of our proposed new detection paradigm. Different from pervious approaches that the detection model is explicitly determined by the MLE point of training data, the proposed paradigm formulates the detection from a Bayesian view that at detection phase, the model is updated to maximize its posterior that contemporarily determined by both training and observation.

By computing fisher information and using laplace approximation, we can bridge that gap between the mainstream CNN detectors and "Random Access Memories" paradigm.

The Small Target Prior in Remote Sensing Image. For high resolution aerial remote sensing images, the targets of interest are usually sparse distributed and only occupy a very small number of pixels. In the case of sliding window based detectors, the imbalance between background and desired target could be as extreme as 10^7 background windows to every target window. This could be even true for wide-scale remote sensing images. The complex background distribution leads to a higher demand on the capacity of the model. Larger models are able to capture more complex background distributions, while it may suffer from a higher computation cost, which is especially infeasible for some on-orbit remote sensing applications. An advantage of our method lies that the small target prior can be very easily integrated to the proposed paradigm, as we are able to compress model's capacity to meet the speed requirement while maintaining the detection accuracy.

Contributions. The contributions of our work can be summarized as follows:

1) The key innovation of Random Access Memories lies that the detection model can be adaptively changed during detection phase that contemporarily determined by both training and the latest observations. Such paradigm can be easily integrated with the current CNN based detectors without complex changes. With RAM, a comparable or even higher detection accuracy of a larger model can be obtained with less parameters and a faster detection speed.

2) We introduce a new dataset "LEVIR"[1] for remote sensing target detection task, which is more challenging and one order of magnitude larger than existing datasets. We have made LEVIR open access at http://levir.buaa.edu.cn/Code.htm.

The rest of this paper is organized as follows. In section II, we will review the recent advances in general object detection for natural images. In section III, we will give a detailed introduction to our proposed detection paradigm. In section IV, we will introduce LEVIR, a new dataset for remote sensing target detection. Some experimental results are given in section V, and the conclusions are drawn in Section VI.

[1]LEVIR is the name of the authors' laboratory: LEarning, VIsion and Remote sensing laboratory.

II. Recent Advances in Natural Image Object Detection

In computer vision, there has been great progress of natural image object detection methods in recent years. Recent advances of deep CNN [19–21] has made a great improvement on the general object detection tasks for natural images. While early approaches [22–24] simply formulate the detection into an sliding window traversal + classification problem (background VS objects of interest), recent CNN based detection methods mainly focus on the following four aspects: 1) efficient algorithms for multi-scale detection, 2) accurate bounding-box prediction 3) training with imbalance data and 4) speedup strategy.

For multi-scale detection, the most straight forward way is to build feature map pyramids [25, 26]. Recent progress includes using external object proposals [27–29], multi-scale anchors [30] and integrated multi-scale detection [31]. For bounding-box prediction, efforts have been made to improve the accuracy by stepwise bounding-box correction [32] and the probabilistic inference of location [33]. For the problem of imbalance data, some useful strategies include cascaded training/detection [22, 23, 26] and hard-negative mining [24–26, 34]. Time efficiency is another key factor in the object detection, especially for remote sensing images. Early approaches, which are designed based on sliding window techniques, usually search for objects exhaustively at different locations and scales [22–26]. While it may be possible for fast detection of certain object categories (e.g. face and pedestrian [22–24]) with extensive speed up strategies (e.g. integral image/channel [22, 23, 35], feature pyramid approximation

[36] and cascaded-detection [22, 23, 37, 38]), it is still hard to extend such ideas to the fast detection of multiple categories. Some of the recent CNN based detection methods [30, 39, 40], with larger model capacity and stronger representation ability, apply a fixed set of filters with multiple bounding-box references on a fixed set of convolutional feature maps to speed up the detection process.

Despite that great efforts have been made, for some important remote sensing applications such as wide-scale remote sensing monitoring or even on-orbit target detection, these algorithms are still far from being practical at present due to the complex background features, drastically changes of the target scales [41] and the computational cost requirements. Although some multi-scale techniques have largely improved the detection performance of small objects, such as detecting objects from the feature maps of different depth [39, 42], or using multi-scale feature fusion [43–45] to improve the representation ability of small objects, in this paper, we try to study this problem from another point of view.

III. Random Access Memories

In this section, we will give a detailed description to our detection paradigm and explain how it works with a CNN based detector.

A. Fisher Information

The training process of any target detector is essentially a likelihood estimation process. Here we follow a classical learning paradigm that the optimal model $\hat{\mathcal{H}}$ can be determined by the i.i.d. training data $\mathcal{D}_{tr}$ and the model's hypothesis space $\mathcal{F}$

$$\hat{\mathcal{H}} = \text{argmax}_{\mathcal{H}\in\mathcal{F}} p(\mathcal{D}_{tr}|\mathcal{H}). \tag{1}$$

For any early sliding window based detectors [22–26], $\mathcal{D}_{tr}$ means the collection of image data and label within any sliding window region, while for the recent CNN based detectors [30, 31, 39, 40], $\mathcal{D}_{tr}$ corresponds to that of the respective field. Arguably, under the view of Bayesian statistics, $\hat{\mathcal{H}}$ is not unique since for any perturbations of training data, the estimated model maybe different. To describe its potential variants as the data $\mathcal{D}_{tr}$ changes, here we introduce *fisher information* as a basic metric. The fisher information is a way of measuring the amount of information that an observable training data $\mathcal{D}_{tr}$ carries about an unknown model $\mathcal{H}$. For a model with single parameter θ, the fisher information can be defined as follows:

$$\mathcal{I}(\hat{\theta}) = -E\{\frac{\partial^2}{\partial\theta^2} log(p(\mathcal{D}_{tr}|\theta))\}, \tag{2}$$

which is equivalent to the second derivative (if it exists) of the negative log likelihood function. $\mathcal{I}(\theta)$ can be viewed as a measurement of the "curvature" of the support curve near the MLE point $\hat{\theta}$, where a "blunt" support curve (one with a shallow maximum) could have a low negative expected second derivative, and thus low information, while a sharp one could have a high information.

For a detection model $\mathcal{H}$ with multiple parameters $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_N]^T$, the fisher information can be written as its matrix form $\boldsymbol{\mathcal{I}}(\hat{\mathcal{H}})$, where its element-wise representation is

$$\boldsymbol{\mathcal{I}}(\hat{\mathcal{H}})_{i,j} = -E\{\frac{\partial^2}{\partial\theta_i\partial\theta_j} log(p(\mathcal{D}_{tr}|\boldsymbol{\theta}))\}. \tag{3}$$

$\boldsymbol{\mathcal{I}}(\hat{\mathcal{H}})$ is equivalent to the Hessian matrix (if it exists) of the negative log likelihood function. It can be also understood as a metric of an appropriate changes of any variables induced from the Euclidean metric.

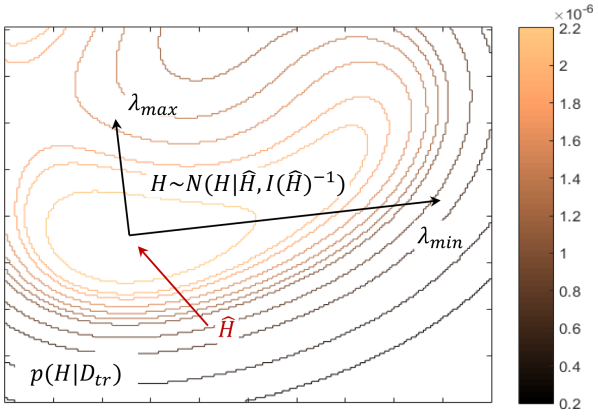

Fig. 2. Contour line of $p(\mathcal{H}|\mathcal{D}_{tr})$ on a group of toy data. In our method, the laplace approximation acts like a regularization when updating the memories at detection phase. Those parameters with small engine values of fisher information matrix turn out to be more likely to be updated, while those with large engine values are relatively stable.

B. Laplace Approximation

Here we use a simple but widely used framework called the Laplace approximation, that aims to find a Gaussian approximation to a probability density defined over a set of continuous variables [16]. Laplace Approximation can be implemented by fitting the model with a second order Taylor expansion of the log likelihood function $log(p(\mathcal{H}|\mathcal{D}_{tr}))$ around the its maximum point $\hat{\mathcal{H}}$. The estimated distribution finally is conducted by $\hat{\mathcal{H}}$ as the its mean and $\boldsymbol{\mathcal{I}}(\hat{\mathcal{H}})$ as its covariance matrix

$$\mathcal{H} \sim \mathcal{N}(\mathcal{H}|\hat{\mathcal{H}}, \boldsymbol{\mathcal{I}}(\hat{\mathcal{H}})^{-1}). \tag{4}$$

In this way, the probability value at any changes of the model can be represented as

$$p(\hat{\mathcal{H}} + \delta\mathcal{H}) \propto exp(-\frac{1}{2}\|\delta\mathcal{H}\|^2_{\boldsymbol{\mathcal{I}}(\hat{\mathcal{H}})}), \tag{5}$$

where $\|\cdot\|^2_{\boldsymbol{\mathcal{I}}(\hat{\mathcal{H}})}$ represents the $\boldsymbol{\mathcal{I}}(\hat{\mathcal{H}})$ norm metric of the model changes.

C. Updating Memories

The fisher information can be regarded as a very important prior that guides the update of the model. During detection phase, the detection model $\hat{\mathcal{H}}$ will be changed according to both of the approximated distribution (4) and the observation $\mathcal{D}_{ob}$. The posterior distribution of $\mathcal{H}$ can be represented as

$$\begin{aligned} p(\mathcal{H}|\mathcal{D}_{ob}) &= \frac{p(\mathcal{H})p(\mathcal{D}_{ob}|\mathcal{H})}{p(\mathcal{D}_{ob})} \\ &\sim p(\mathcal{H})p(\mathcal{D}_{ob}|\mathcal{H}). \end{aligned} \tag{6}$$

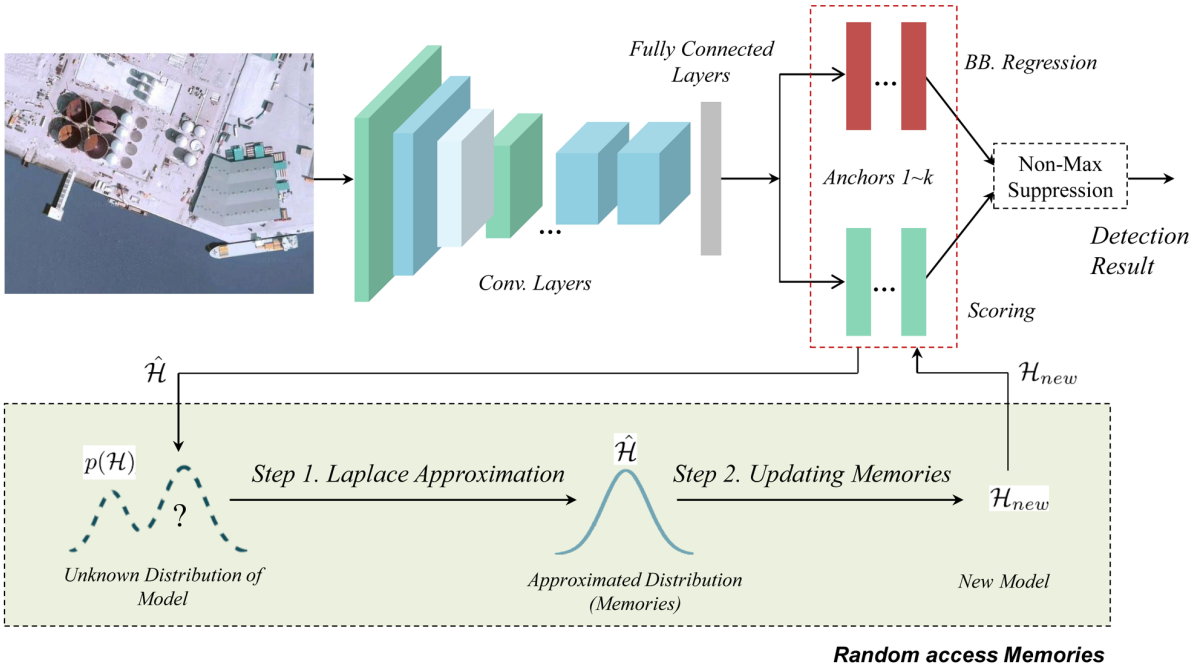

Fig. 3. A detailed Overview of our proposed new detection paradigm: backbone of the detector and random access memories. The RAM contains three steps: 1) Approximating the unknown distribution $p(\mathcal{H})$ from $\hat{\mathcal{H}}$ by Laplace Approximation, 2) updating the memories to get the new detection model $\mathcal{H}_{new}$.

By dividing the likelihood function $p(\mathcal{D}_{ob}|\mathcal{H})$ into the positive part (target of interest $\mathcal{D}_{ob}^{+}$) and negative part (undesired background $\mathcal{D}_{ob}^{-}$), and substituting (5) into $p(\mathcal{H})$, the posterior can be expanded as

$$\begin{aligned} p(\mathcal{H}|\mathcal{D}_{ob}) &\sim p(\mathcal{H})p(\mathcal{D}_{ob}^{+}|\mathcal{H})p(\mathcal{D}_{ob}^{-}|\mathcal{H}) \\ &= exp(-\frac{1}{2}\|\mathcal{H}-\hat{\mathcal{H}}\|^2_{\boldsymbol{\mathcal{I}}(\hat{\mathcal{H}})})p(\mathcal{D}_{ob}^{+}|\mathcal{H})p(\mathcal{D}_{ob}^{-}|\mathcal{H}) \end{aligned} \tag{7}$$

For any latest observations, the model should be updated to access the maximum of posterior to best fit the training and testing data at the same time

$$\hat{\mathcal{H}}_{new} = \text{argmax}_{\mathcal{H}\in\mathcal{F}} p(\mathcal{H}|\mathcal{D}_{ob}). \tag{8}$$

Clearly, for some directions of $\boldsymbol{\mathcal{I}}(\hat{\mathcal{H}})$, a slight change of the model $\delta\mathcal{H}$ would cause a rapid decline of the probability $p(\mathcal{H}|\mathcal{D}_{ob})$, while for some other directions the probability may keep stable. Fig. 2 shows the contour line of $p(\mathcal{H}|\mathcal{D}_{tr})$ on a group of toy data. If we move a step further and make an eigenvalue decomposition of the fisher information matrix $\boldsymbol{\mathcal{I}}(\mathcal{H}) = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T$, where $\boldsymbol{\Sigma} = diag(\lambda_1, \lambda_2, \ldots, \lambda_N)$, an important conclusion can be easily derived that the direction of the eigenvector with the largest eigenvalue λ_{max} will be the fastest dropping direction of $p(\hat{\mathcal{H}})$. We call those associated parameters *permanent memories*. The direction with the smallest eigenvalue λ_{min} will be that of the slowest one. We call those associated parameters *short time memories*. The *permanent memories* acts as an important regularization at inference stage, while *short time memories* are more likely to be updated according to the observations.

D. Integrating The Small Target Priors

It is hard to optimize (8) directly since it contains latent variables $\mathcal{D}_{ob}^{+}$ and $\mathcal{D}_{ob}^{-}$, where "+" and "-" represent the latent labels. Since the amount of the negatives are usually far greater than the positives for a remote sensing image: $N^{-} \gg N^{+}$, we are reasonable to neglect the positive latent variables. In this way, (8) can be relaxed to its negative logarithmic likelihood form

$$\text{min}_{\mathcal{H}} L(\mathcal{H}|\mathcal{D}_{ob}) = -log(p(\mathcal{D}_{ob}^{-}|\mathcal{H})) + \alpha\|\mathcal{H}-\hat{\mathcal{H}}\|^2_{\boldsymbol{\mathcal{I}}(\hat{\mathcal{H}})}, \tag{9}$$

where $\|\mathcal{H}-\hat{\mathcal{H}}\|^2_{\boldsymbol{\mathcal{I}}(\hat{\mathcal{H}})}$ can be seen as a constraint which controls the update range of the model. $\alpha > 0$ is a positive number controls the degree of the constraint.

Despite the latent variables has been eliminated, another problem lies that the high dimensional parameter space of a deep CNN makes it still very difficult to directly compute and store the fisher information matrix $\boldsymbol{\mathcal{I}}(\hat{\mathcal{H}})$. A further assumption can be made that only a small portion of the memories are updated during detection phase. Here we take a simple and straightforward manner, that only optimizing the parameters of the final output layer while keeping those of previous layers fixed as feature extractor. More specifically, when a testing image is fed into the network, passing through all convolutional layers and finally comes to the full-connected layer, the RAM operation should be performed. Since the regularization $\|\mathcal{H}-\hat{\mathcal{H}}\|^2_{\boldsymbol{\mathcal{I}}(\hat{\mathcal{H}})}$ is always convex at any local/gobal maximum point, any types of the convex loss functions of $-log(p(\mathcal{D}|\boldsymbol{\theta}))$, e.g. square loss, smoothed L1 loss [28, 30], L2 loss or cross-entropy loss [46] would lead to a final unique solution. In this way, (9) can be efficiently optimized by any convex optimization algorithms, e.g. gradient decent or

Newton method, and global optimal solution will always be guaranteed. For some special cases, saying that using square loss or smoothed L1 loss, (9) would degenerate into a non-constraint quadratic programming problem, thus a closed form solution can be simply obtained, as we will see it later.

E. Implementation Details

Fig. 3 shows the backbone of our detector and the random access memories operation during detection phase. Our implementation details are given as follows.

Backbone. Current CNN based object detection methods can be divided into two important branches based on their processing flow. The first branch is cascaded detectors where the detection is performed from a coarse to fine manner [27–30], while the second branch is integrated ones where the detection is evaluated only once [31, 39, 40]. In this paper, we take the second one as the backbone of our detector which is faster and may have larger rooms of improvements in the future. Here we have designed three types of networks, a tiny one, a medium one and a large one. Their configurations are listed as follows ("Layer-name(Number of Filters, Size/Stride)"), where the ReLU layer between a convolutional layer and a maxpooling layer is omitted for simplicity:

***Tiny CNN**: conv.(256, 3×3/1) + maxpool.(-, 3×3/3) + conv.(256, 3×3/1) + maxpool.(-, 3×3/3) + conv.(512, 3×3/1) + maxpool.(-, 2×2/2) + full-cnect.(512× k(C+4)).*

***Medium CNN**: conv.(256, 3×3/1) + maxpool.(-, 2×2/2) + conv.(512, 3×3/1) + maxpool.(-, 2×2/2) + conv.(512, 3×3/1) + maxpool.(-, 2×2/2) + conv.(1024, 3×3/1) + conv.(1024, 1×1/1) + full-cnect.(1024×k(C+4)).*

***Large CNN**: VGG-f [20] (decision layer removed) + full-cnect.(4096×k(C+4)).*

For tiny and medium size networks, similar to the VGG network, we use mostly 3×3 filters [20] except for the last fully-connection layer (1×1 convolution). The output depth of the last layer is $k(C+4)$, which depends on the number of multi-scale anchors k, number of target categories C, and 4 coordinates of the bounding box. Our implementation of the network is based on matconvnet-1.0 beta23 [47].

Loss Layer Design. We follow the idea of pre-defined anchors for multi-scale detection [39, 40]. The output dimensions are organized as in Fig. 4. There are 3 predefined anchors: 0.9×0.9, 0.7×0.7 and 0.5×0.5 for each detection window. For each anchor scale, a multi-task loss is designed which consists of a category scoring loss and a bounding box prediction loss

$$\begin{aligned} L(\mathbf{p},\mathbf{p}^*,\mathbf{t},\mathbf{t}^*) &= L_{scoring}(\mathbf{p},\mathbf{p}^*) + \beta I(\mathbf{p}) L_{pred.}(\mathbf{t},\mathbf{t}^*) \\ I(\mathbf{p}) &= \begin{cases} 1 & \text{IoU}\{\text{Anchor}(\mathbf{p}),\mathbf{t}^*\} > 0.5 \\ 0 & \text{else} \end{cases} \end{aligned} \tag{10}$$

where $\text{IoU}\{\cdot\}$ refers to the intersection over union overlap between two regions. Anchor($\mathbf{p}$) means the anchor box of a certain scale. $I(\mathbf{p})$ is an indicator controls whether it is an target of interest under a certain anchor scale.

For bounding box prediction, we use the smoothed L1 loss [28, 30] that is less sensitive to outliers than the L2 loss used in [27]

$$L_{reg.}(t_i,t_i^*) = \begin{cases} 5(t_i-t_i^*)^2 & |t_i-t_i^*| \leqslant 0.1 \\ |t_i-t_i^*| - 0.05 & \text{else.} \end{cases} \tag{11}$$

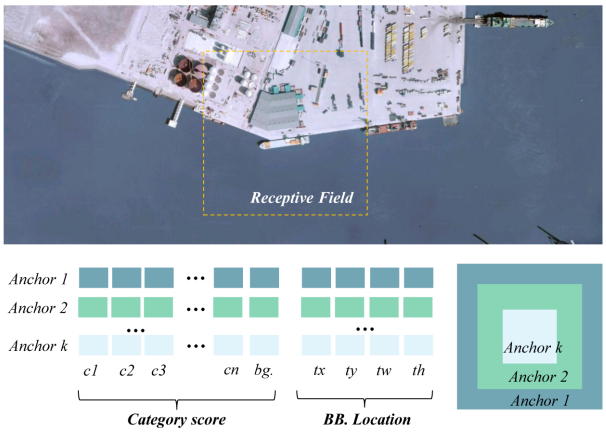

Fig. 4. An example of respective field and its multi-anchor boxes. For each anchor scale, the ground-truth label is arranged as two parts: category score and bounding box location coordinates. The final ground-truth label of the respective field can be formed by end-to-end connection of the label of each anchor scale.

We use the parameterized coordinates as it was used in [30]. For category scoring, we also simply see it as a score regression problem with the smoothed L1 loss. Similar ideas have been used in [31]. Since weights of the fully connected layer are learnt with independent loss. The memory update operation can be individual performed with corresponding anchor scale and target category. When we use the smoothed L1 loss, the negative log likelihood loss term in (9) can be represented as

$$-log(p(\mathcal{D}_{ob}^-|\mathcal{H})) = E_{(\mathbf{x},y)\sim\mathcal{D}_{ob}}\{L_{reg}(\boldsymbol{\theta}^T\mathbf{x},y)\}, \tag{12}$$

where $\mathbf{x}$ refers to the features of the last convolutional layer, $\hat{\boldsymbol{\theta}}$ and y refer to the convolutional filters and ground-truth label of specific category and anchor-scale. Then the optimization problem (9) can be further written as

$$\begin{aligned} \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}|\mathcal{D}_{ob}) &= E_{(\mathbf{x},y)\sim\mathcal{D}_{ob}}\{L_{reg}(\boldsymbol{\theta}^T\mathbf{x},y)\} \\ &\quad + \alpha\|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\|^2_{\boldsymbol{\mathcal{I}}(\hat{\boldsymbol{\theta}})}. \end{aligned} \tag{13}$$

The above optimization problem has an approximated closed form solution:

$$\boldsymbol{\theta}_{new} = (\mathbf{C}+\alpha\boldsymbol{\mathcal{I}}(\hat{\mathcal{H}}))^{-1}(E_{(\mathbf{x},y)\sim\mathcal{D}_{ob}}\{y\mathbf{x}\}+\alpha\boldsymbol{\mathcal{I}}(\hat{\mathcal{H}})\hat{\boldsymbol{\theta}}), \tag{14}$$

where $\mathbf{C}$ and $\boldsymbol{\mathcal{I}}(\hat{\mathcal{H}})$ has the following expression:

$$\begin{aligned} \boldsymbol{\mathcal{I}}(\hat{\mathcal{H}}) &= E_{(\mathbf{x},y)\sim\mathcal{D}_{tr}}\{\xi(\mathbf{x},y)\mathbf{x}\mathbf{x}^T\} \\ \mathbf{C} &= E_{(\mathbf{x},y)\sim\mathcal{D}_{ob}}\{\xi(\mathbf{x},y)\mathbf{x}\mathbf{x}^T\}, \\ \xi(\mathbf{x},y) &= \begin{cases} 1 & |\hat{\boldsymbol{\theta}}^T\mathbf{x}-y| < 0.1 \\ 0 & \text{else.} \end{cases} \end{aligned} \tag{15}$$

In this way, the model can be updated easily without any iterative operations.

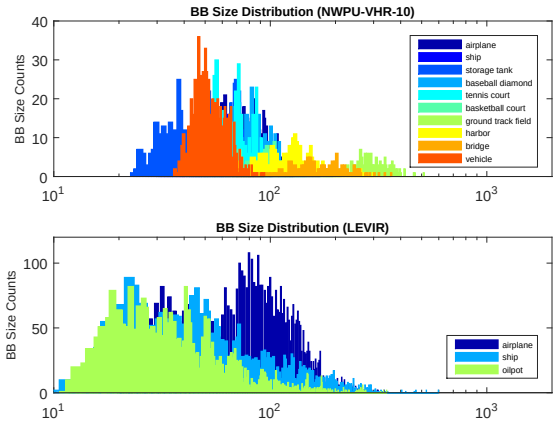

Fig. 5. Distribution of BB size of LEVIR and NWPU-VHR-10.

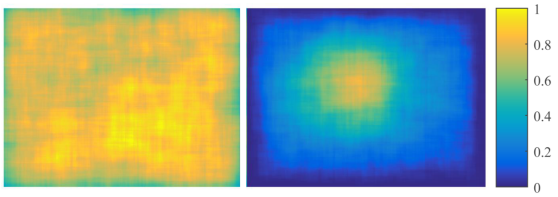

Fig. 6. Heat map of BB location for LEVIR (left) and NWPU-VHR-10 (right).

IV. LEVIR: A New Dataset for Remote Sensing Target Detection

Large and challenging datasets are necessary for the progress in remote sensing applications. Our goal in introducing the LEVIR detection dataset is to provide a better benchmark and statistically meaningful evaluations for those current and future detection methods.

LEVIR consists of a large number of high resolution Google Earth images with over 22k images of 800×600 pixels and 0.2m∼1.0m/pixel's resolution. LEVIR covers most types of ground features of human living environment, e.g. city, country, mountain area and ocean. Extreme land environments such as glacier, desert and gobi are not considered in our dataset. There are 3 types of targets in our dataset: airplane, ship (including both inshore ships and offshore ships) and oilpot. We label all the images for a total of 11k independent bounding boxes (BB) including 4,724 airplanes, 3,025 ships and 3,279 oilpots. The average number of targets per image is 0.5. For every image in which a given target of interest is visible, we draw a tight BB that indicates the full extent of the entire target. It should be noticed that for those targets that are partially outside the image boundary or occluded by other objects, this involves estimating the location of hidden parts. A summary of our dataset is given in Table. I.

During the last decades, some efforts have been made to develop public dataset for target detection from aerial and satellite images. Here we list the detailed overview of three public available datasets:

NWPU-VHR-10 [5] is a ten-class dataset which contains 800 images. There are a total of 757 airplanes, 302 ships, 655 storage tanks, 390 baseball diamonds, 524 tennis courts, 159 basketball courts, 163 ground track fields, 224 harbors, 124 bridges, and 477 vehicles manually annotated with BBs for ground truth.

TABLE I
A Summary of LEVIR dataset

Type	Item	Value
Image Info.	# total images	21,952
	image size	600×800 pixels
	image resolution	0.2m∼1.0m/pixel
	modality	RGB image
BB Info.	# individual BB	11,028
	# airplanes	4,724
	# ships	3,025
	# oilpots	3,279
	# BB with hidden parts	2,940
	BB size	10∼600 pixels

TAS Aerial Car Detection Dataset (TACDD) [48] consists of 30 images acquired from Google Earth with the size of 729×636 pixels. 1319 vehicles are manually labeled with BBs for groundtruth.

Overhead Imagery Research Dataset (OIRDS) [49] is designed for vehicle detection with the collection of about 900 images captured by aircraft-mounted camera. The total number of labeled vehicles is about 1,800.

Table II provides a detailed comparison of LEVIR and other existing remote sensing target detection datasets. NWPU-VHR-10 has helped drive recent advances [1, 4, 5, 10] in target detection of remote sensing images. TACDD and OIRDS remain the most widely used for vehicle dtection [48–50]. However, their defects are also obvious. Firstly, all these datasets are too limited to obtain statistically meaningful training and evaluation result for the most current deep learning based detection methods. Secondly, NWPU-VHR-10 also shows a strong bias toward large, unoccluded targets. Last but not least, these datasets also have the "center biased" problem, saying that the target tends to appear near the center of a image. In fact, unlike natural images where their contents are limited and the objects of interests are usually near the center of an image, remote sensing images, if not manually selected and cropped, usually have no specific region of interest. In Fig. 5, we histogram the size of all BBs of LEVIR and NWPU-VHR-10. We can see LEVIR contains targets with a larger range of scales, especially for small target. Fig. 6 shows the heat-map of the BB location probability of LEVIR (left) and NWPU-VHR-10 (right). As we can see, LEVIR shows a more evenly distributed BB location than that of NWPU-VHR-10.

V. Experimental Results and Analysis

In this section, we have made an extensive evaluation and comparison with several variants of our model and other detection methods. Our experiments are performed on LEVIR and NWPU-VHR-10. For LEVIR, 70% images are used for training and rests are used for test. For NWPU-VHR-10, we use the same training-testing split criterion with other comparison methods as they originally used in their paper.

TABLE II
A COMPARISON BETWEEN LEVIR AND OTHER REMOTE SENSING DETECTION DATASETS

Dataset	# Imgs	# Classes	# BBs	Property		
				no selec.bias	occ.target	estim.BB.bound.
NWPU-VHR-10 [5]	0.8k	10	3.8k	×	×	×
TACDD [48]	0.03k	1	1.3k	×	×	×
OIRDS [49]	0.9k	1	1.8k	×	×	×
LEVIR	22.0k	3	11.0k	✓	✓	✓

A. Training Details

Data Augmentation. To detect targets with different directions, each individual target is randomly rotated for several times and then randomly resized and translated to make sure that there are sufficient information for each anchor-scale to learn. We use the similar criterion that is used in [30] that we assign a target label to two kinds of anchors: 1) the anchor with the highest IoU overlap with a ground-truth box, or 2) an anchor whose IoU overlap is higher than 0.5 with any ground-truth box.

Pre-training. For our large-sized model, the initial weight is transferred from that of VGG-f which is trained on the ImageNet. For our tiny-sized and medium-sized models, the networks are first pre-trained with targets and randomly generated backgrounds. The training is performed for 20 epochs by back-propagation and stochastic gradient descent (SGD) [46].

Hard-negative Mining. After pre-training, we fine-tune the model, where in each mining iteration the training set is augmented with hard negative examples. We assign a background label to any detected false negative anchor if its IoU overlap is lower than 0.2 for all ground-truth BBs. Anchors that are neither positive nor negative do not contribute to the training. In each mining iteration, the detector collects over 500 hard negatives and the SGD is performed for 2 epochs.

Multi-octave Detection. Although recent advances [30, 31, 39, 40] advocate using single scale feature map to detection target of multiple scales since it offers trade-off between accuracy and speed, detection on pyramid feature maps still can be proved with better performance, especially for small sized targets [21, 43]. Since there is a large scale variation of remote sensing targets, we build a sparse image pyramid with octave stride $os = 2$ as the network's inputs. For scale variations within a single octave, the scale range can be well captured by the detector of multi-scale anchors. The regularization parameter α of (9) is set to 100 for all experiments. All the important parameters we used are listed in Table III.

TABLE III
PARAMETER SETTINGS

Stage	Parameter	Value
Pre-training	epoch num.	20
	learning rate	10^{-3}
	batch num.	100
	momentum	0.9
	weight decay	5×10^{-4}
Hard-Neg. Mining	IoU thresh	0.2
	learning rate	10^{-4}
	mining iter. num.	50
Detection	regularizer α	100
	octave stride	2

TABLE IV
COMPARISONS OF THE PROPOSED METHODS AND THEIR BASELINES.

Method	plane	ship	oilpot	mAP
TINY-BASE	45.0%	17.2%	39.6%	33.9%
TINY-RAM	50.5%	19.3%	43.7%	37.8%
MEDIUM-BASE	73.7%	48.4%	46.4%	56.2%
MEDIUM-RAM	76.0%	50.0%	48.1%	58.0%
LARGE-BASE	70.6%	60.5%	42.1%	57.7%
LARGE-RAM	71.7%	60.8%	43.0%	58.5%

B. Overall Results Statistics

Comparisons with Baselines. For fair comparisons with our detection paradigm, we have designed three baseline methods:

- Baseline 1: tiny-networks (*TINY-BASE*),
- Baseline 2: medium-networks (*MEDIUM-BASE*),
- Baseline 3: VGG-f based networks (*LARGE-BASE*),

and their improved variants with memories-updating:

- Proposed 1: tiny-networks+ram (*TINY-RAM*),
- Proposed 2: medium-networks+ram (*MEDIUM-RAM*),
- Proposed 3: VGG-f based networks+ram (*LARGE-RAM*).

During evaluation, some ambiguous instances are excluded from our dataset. There are two kinds of situations that we identify it as an ambiguous target. The first type is that the targets bounding-box are partially clustered or outside (larger than 3/4 of its area) the image. The second type refers to the instance with very small size, whose length is smaller than 20 pixels. To detect smaller target, we suggest using higher resolution images. Any false detection or missing of these targets will not be taken into account neither as a false positive nor as a true positive.

All comparisons use the same settings including networks' hyperparameters, training parameters and detection parameters. Table IV shows the three baseline methods and their variations on LEVIR dataset[2]. We can clearly observe RAM gives an overall improvement on all of the three classes and three varies sized models. We also observed that the performance enhancement of the small sized model is more remarkable than that of the medium and large sized model. This is simply because a smaller model tends to be saturated when training with background of complex distribution. RAM

[2]Sine *LARGE-RAM* model takes too much time to in RAM process, we only sample a subset of our test data for this model. For other models, the full test set is used.

TABLE V
PERFORMANCE COMPARISON OF DIFFERENT METHODS ON NWPU-VHR-10 DATASET.

Method	plane	ship	stor.tank	baseb.	tenn.	basket.	gr.trac.	harbor	bridge	vehicle	mAP
FDDL[4]	29.2%	37.6%	77.0%	25.8%	2.8%	3.6%	20.1%	25.4%	21.5%	4.5%	24.7%
COPD[5]	62.3%	68.9%	63.7%	83.3%	32.1%	36.3%	85.3%	55.3%	14.8%	44.0%	54.6%
RICNN[10]	88.4%	77.3%	85.3%	88.1%	40.8%	58.5%	86.7%	86.6%	61.5%	71.1%	72.6%
TINY-RAM	83.5%	30.4%	77.3%	62.7%	20.2%	6.3%	34.1%	16.1%	4.5%	42.5%	37.8%
MEDIUM-RAM	89.6%	71.7%	72.1%	86.4%	27.6%	30.7%	52.5%	26.4%	23.3%	51.0%	53.1%
LARGE-RAM	94.1%	85.5%	85.9%	89.6%	67.1%	63.9%	48.9%	62.9%	58.7%	81.6%	73.8%

Fig. 7. Selected examples of our detection results on LEVIR dataset (categories: airplanes, ships, oilpots).

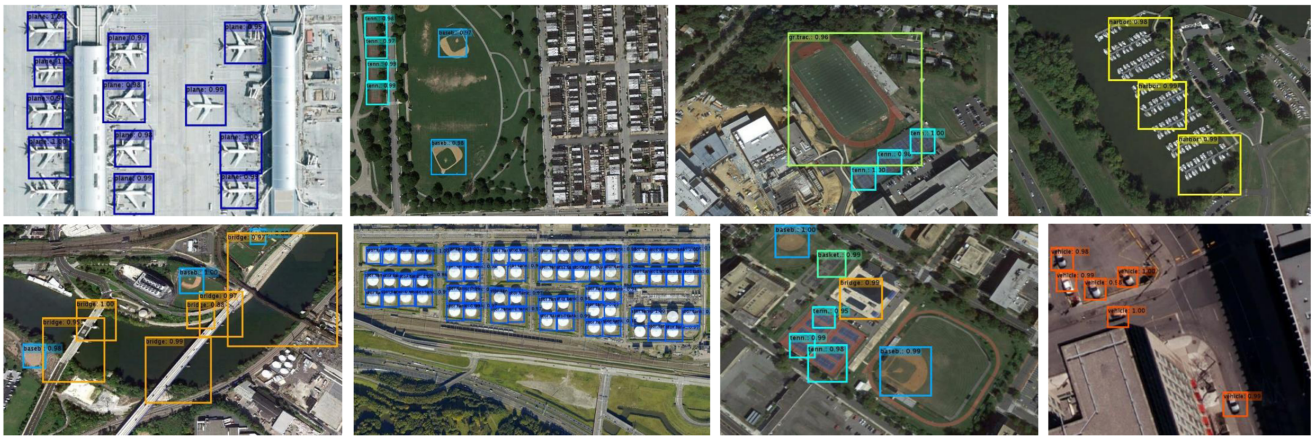

Fig. 8. Selected examples of our detection results on NWPU-VHR-10 dataset (categories: airplanes, ships, storage tanks, baseball diamonds, tennis courts, basketball courts, ground track fields, harbors, bridges and vehicles).

TABLE VI
COMPARISONS OF THREE DIFFERENT WAYS OF FISHER MATRIX APPROXIMATION: 1) WITHOUT APPROXIMATION, 2) DIAGONAL MATRIX APPROXIMATION AND 3) IDENTITY MATRIX APPROXIMATION.

Method	$\mathcal{I}(\hat{\mathcal{H}})$ Approximation	mAP
TINY-RAM	*Fisher*	37.8%
TINY-RAM	*Diagonal*	33.9%
TINY-RAM	*Identity*	29.4%
MEDIUM-RAM	*Fisher*	58.0%
MEDIUM-RAM	*Diagonal*	56.1%
MEDIUM-RAM	*Identity*	52.6%
LARGE-RAM	*Fisher*	58.5%
LARGE-RAM	*Diagonal*	57.7%
LARGE-RAM	*Identity*	53.4%
TINY-BASE	–	33.9%
MEDIUM-BASE	–	56.2%
LARGE-BASE	–	57.7%

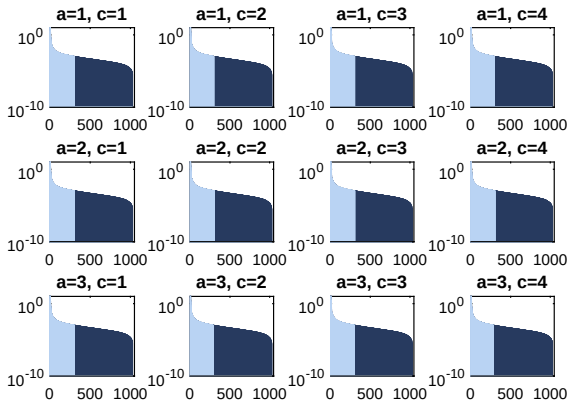

Fig. 9. Each sub-figure shows the eigenvalues of the fisher information matrix with a specific anchor scale "a" and class id "c". The fisher information matrices are accumulated with both pre-training data and hard negatives at MLE point. Since most of the eigenvalues are very close to zeros (less than 10^{-3}, marked as dark blue bars) except for a few large ones (marked as light blue bars), our model will have a free update on those eigenvectors at the detection phase.

helps small model concentrate on the newly observed data and "forget" part of the useless memories previously learned from the training data. Fig. 7 and Fig. 8 shows some examples of detection results on LEVIR dataset and NWPU-VHR-10 dataset.

Fig. 9 shows the eigenvalues of the fisher information matrices accumulated on our training data (include pre-training data and hard negatives) at MLE point. As we can see, most of the eigenvalues stay almost close to zero (marked as dark blue bars) except for only a few large ones (marked as light blue bars). This phenomenon indicates that although our model is fully trained, their uncertainty is still large at these corresponding eigenvector directions and any disturbance on these directions will not make clear decrements of the training objective function. These eigenvectors serve as the main paths for updating the model.

Comparisons on NWPU-VHR-10. We also evaluate the performance of our method on NWPU-VHR-10 dataset and compare with other state-of-the-art remote sensing detection methods e.g. RICNN [10], COPD [5] and FDDL [4]. Among them, RICNN learns a rotation-invariant CNN model by introducing and learning a new Rotation-Invariant layer on the basis of the existing CNN architectures to detect remote sensing target with different orientations. COPD is designed based on the COllection of Part Detectors with a sliding window approach on Histograms of Oriented Gradients map and linear support vector machine classifier. FDDL is designed based on Fisher Discrimination Dictionary Learning method, where a set of target candidate regions are firstly generated by a saliency detection method and then a sparse representation based classifier is adopted on each candidate to perform multi-class detection.

We use the same training-testing split criterion as those was used in [10] for a fair comparison. Table V lists their performance. Since the training data is limited for large networks (vgg based model) to obtain statistically meaningful training results, we have added some external data in training set, including the LEVIR data and some other samples from Google Earth images. The results of RICNN, COPD and FDDL are reported by [10]. We can see our model (LARGE-RAM) obtains the best detection results.

How Important is Fisher Information? The way to compute the fisher information matrix is a key point of our method that it describes how well the model distribution is approximated at its MLE point. In this experiment, we give two approximation forms to its original one: 1) the diagonal matrix approximation where only its diagonal elements are left while others are set zeros and 2) the identity matrix approximation where the matrix is further simplified as an identity matrix. Table VI shows their comparisons based on the three proposed variations. For identity matrix approximation, updating memories under such priors means that the pre-trained model has an equivalent probability of updating their parameters in any directions in the parameter space. The updating process can be simply viewed as a maximization of likelihood of observations under a Euclidean distance constraint. We can see there is a little drop of accuracy compared with their fisher-approximated models in this condition. For diagonal matrix approximation, a similar explanation can be given while the only difference is that the dimensions are weighted by its diagonal elements when computing the Euclidean distance constraint. The accuracy also drops at this time.

Hyper-parameter Stability. The regularization coefficient α of (9) serves as a very important hyper-parameter in our method. Fig. 10 and Fig. 11 show the performances of the above three approximation ways and their baselines respectively with different α. Performance enhancement can be observed with a wide range of the choice of parameter. When α is set too small, the accuracy is lower than the baseline method as we expected. This is because the constraint on the model near the MLE point is so weak that some targets are over-suppressed. When α is set too large, the model will be bounded at a very small feasible region near its MLE point thus the improvement is very little. When $\alpha \to \infty$, $\mathcal{H}_{new}$ equals to $\hat{\mathcal{H}}$.

Speed Performance We test our method on an Intel i7 PC with a Nvidia GTX 1080Ti graphics card. We use the GPU to accelerate the training and detection process. The training

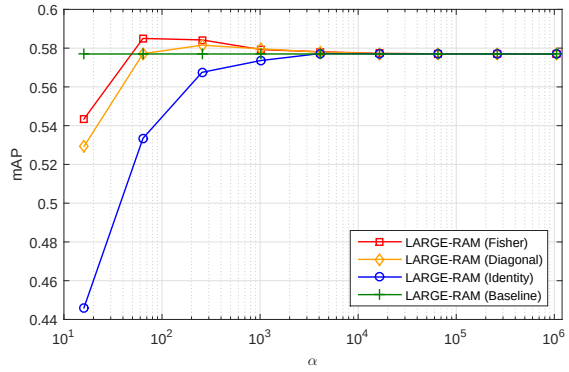

Fig. 10. Performances of our method on LEVIR dataset with different values of regularization coefficient α.

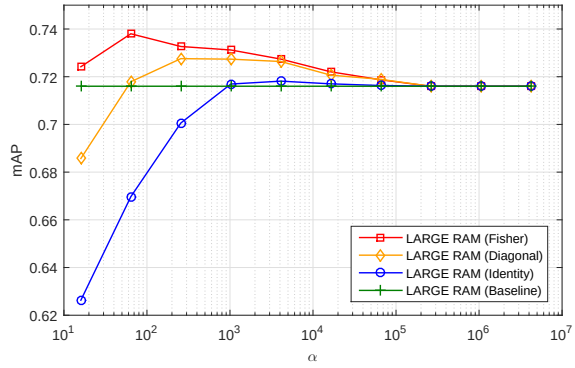

Fig. 11. Performances of our method on NWPU-VHR-10 dataset with different values of regularization coefficient α.

process takes a few hours to days on various sized models. For a 800×600 sized input image, a fast version of our method only takes about 0.1s (10fps) to finish the forward pass (0.02s) + memories updating process (0.10s). Table VII lists the total detection time and memory updating time of the baseline methods and the proposed four variants. For RICNN and FDDL, the running time is reported by their authors, and the testing image is about the same size. The authors of COPD did not report their time. All our models run much faster in spite that the memory updating process takes the most of the running time. The main time cost when updating memories is from the matrix inversion operation in (15), which has a cubic computational complexity of the number of weights in fully connected layer.

TABLE VII
TIME PERFORMANCE OF DIFFERENT METHODS FOR A 600×800 IMAGE.

Method	Mem. Updating Time	Total Detec. Time
TINY-RAM	0.100s	0.122s
MEDIUM-RAM	0.408s	0.492s
LARGE-RAM	4.001s	4.711s
TINY-BASE	–	0.022s
MEDIUM-BASE	–	0.084s
LARGE-BASE	–	0.710s
FDDL[4]	–	$\geq$40.0s
RICNN[10]	–	8.700s

With random access the memories, we can obtain a comparable or even higher detection accuracy of a larger model with less parameters and a faster detection speed. For example, in Table III and Table VI, a lighter model, MEDIUM-RAM achieves higher mAP than a VGG-based baseline model LARGE-BASE (mAP 58.0% VS 57.7%), meanwhile, the former one has faster detection speed (speed: 0.492s VS 0.710s). Notice that the vgg-based model takes quite long time to update memories, this is because the dimension of its full-connected layer is 4096, which is much higher than other two models. Nevertheless, as long as the parameter number of the fully connected layer is well configured, the calculation time can be controlled within an acceptable range.

VI. CONCLUSION

We propose a new paradigm called "Random Access Memories" for target detection for high resolution aerial remote sensing image. We also provide a new challenging dataset for remote sensing target detection which is one order of magnitude larger than existing datasets. Experiments have confirmed the validity of the proposed paradigm where noticeable improvements over a CNN based detectors can be observed. The proposed method outperforms several other state-of-the-art remote sensing target detection methods. Besides, RAM may open some novel opportunities of investigation in other applications under small target priors, such as the fast detection of natural image objects, instance segmentation and even image retrieval tasks.

REFERENCES

[1] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *Isprs Journal of Photogrammetry and Remote Sensing*, vol. 117, pp. 11–28, 2016.

[2] Z. Shi, X. Yu, Z. Jiang, and B. Li, "Ship detection in high-resolution optical imagery based on anomaly detector and local shape feature," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 8, pp. 4511–4523, 2014.

[3] L. Liu and Z. Shi, "Airplane detection based on rotation invariant and sparse coding in remote sensing images," *Optik - International Journal for Light and Electron Optics*, vol. 125, no. 18, pp. 5327–5333, 2014.

[4] J. Han, P. Zhou, D. Zhang, G. Cheng, L. Guo, Z. Liu, S. Bu, and J. Wu, "Efficient, simultaneous detection of multi-class geospatial targets based on visual saliency modeling and discriminative learning of sparse coding," *Isprs Journal of Photogrammetry and Remote Sensing*, vol. 89, no. 1, pp. 37–48, 2014.

[5] G. Cheng, J. Han, P. Zhou, and L. Guo, "Multi-class geospatial object detection and geographic image classification based on collection of part detectors," *Isprs Journal of Photogrammetry and Remote Sensing*, vol. 98, no. 1, pp. 119–132, 2014.

[6] X. Yu and Z. Shi, "Vehicle detection in remote sensing imagery based on salient information and local shape feature ," *Optik - International Journal for Light and Electron Optics*, vol. 126, no. 20, pp. 2485–2490, 2015.

[7] F. Zhang, B. Du, L. Zhang, and M. Xu, "Weakly supervised learning based on coupled convolutional neural networks for aircraft detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 9, pp. 5553–5563, 2016.

[8] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, "Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning," *IEEE*

Transactions on Geoscience and Remote Sensing, vol. 53, no. 6, pp. 3325–3337, 2015.

[9] L. Zhang, Z. Shi, and J. Wu, "A hierarchical oil tank detector with deep surrounding features for high-resolution optical satellite imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 10, pp. 4895–4909, 2015.

[10] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7405–7415, 2016.

[11] Z. Zou and Z. Shi, "Ship detection in spaceborne optical image with svd networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 5832–5845, 2016.

[12] Z. Shi and Z. Zou, "Can a machine generate humanlike language descriptions for a remote sensing image?" *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 6, pp. 3623–3634, 2017.

[13] H. Lin, Z. Shi, and Z. Zou, "Maritime semantic labeling of optical remote sensing images with multi-scale fully convolutional network," *Remote Sensing*, vol. 9, no. 5, pp. 480–500, 2017.

[14] D. Manolakis, D. Marden, and G. A. Shaw, "Hyperspectral image processing for automatic target detection applications," *Lincoln Laboratory Journal*, vol. 14, no. 1, pp. 79–116, 2003.

[15] R. P. Feynman, R. B. Leighton, M. Sands, and R. B. Lindsay, *The feynman lectures on physics, vol. 3: Quantum mechanics*. AIP, 1966.

[16] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[17] C. E. Rasmussen, *Gaussian processes for machine learning*. Citeseer, 2006.

[18] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Conference and Workshop on Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[22] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2001.

[23] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[24] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2005, pp. 886–893.

[25] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2008, pp. 1–8.

[26] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[27] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[28] R. Girshick, "Fast r-cnn," in *International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Conference and Workshop on Neural Information Processing Systems (NIPS)*, 2015, pp. 91–99.

[31] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[32] D. Yoo, S. Park, J.-Y. Lee, A. S. Paek, and I. So Kweon, "Attentionnet: Aggregating weak directions for accurate object detection," in *International Conference on Computer Vision (ICCV)*, 2015, pp. 2659–2667.

[33] S. Gidaris and N. Komodakis, "Locnet: Improving localization accuracy for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 789–798.

[34] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 761–769.

[35] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *British Machine Vision Conference (BMVC)*.

[36] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[37] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2006, pp. 1491–1498.

[38] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Pedestrian detection at 100 frames per second," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 2903–2910.

[39] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 21–37.

[40] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016.

[41] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 22–40, 2016.

[42] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 354–370.

[43] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *arXiv preprint arXiv:1612.03144*, 2016.

[44] T. Kong, A. Yao, Y. Chen, and F. Sun, "Hypernet: Towards accurate region proposal generation and joint object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 845–853.

[45] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2874–2883.

[46] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[47] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 689–692.

[48] G. Heitz and D. Koller, "Learning spatial context: Using stuff to find things," in *European Conference on Computer Vision*, 2008, pp. 30–43.

[49] F. Tanner, B. Colder, C. Pullen, D. Heagy, M. Eppolito, V. Carlan, C. Oertel, and P. Sallee, "Overhead imagery research data

set - an annotated data library and tools to aid in the development of computer vision algorithms," in *Applied Imagery Pattern Recognition Workshop*, 2009, pp. 1–8.

[50] A. Kembhavi, D. Harwood, and L. S. Davis, "Vehicle detection using partial least squares," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1250–65, 2011.

Zhengxia Zou received his B.S. degree in the School of Astronautics, Beihang University, Beijing, China, in 2013. He is currently working toward his Ph.D. degree in the Image Processing Center, School of Astronautics, Beihang University. His research interests include remote sensing image processing and other related topics in computer vision and machine learning.

Zhenwei Shi (M13) received his Ph.D. degree in mathematics from Dalian University of Technology, Dalian, China, in 2005. He was a Postdoctoral Researcher in the Department of Automation, Tsinghua University, Beijing, China, from 2005 to 2007. He was Visiting Scholar in the Department of Electrical Engineering and Computer Science, Northwestern University, U.S.A., from 2013 to 2014. He is currently a professor and the chairman of the Image Processing Center, School of Astronautics, Beihang University. His current research interests include remote sensing image processing and analysis, computer vision, pattern recognition, and machine learning.

Dr. Shi serves as an Associate Editor for the Infrared Physics and Technology. He has authored or co-authored over 100 scientific papers in refereed journals and proceedings, including the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Neural Networks, the IEEE Transactions on Geoscience and Remote Sensing, the IEEE Geoscience and Remote Sensing Letters and the IEEE Conference on Computer Vision and Pattern Recognition. His personal website is http://levir.buaa.edu.cn/.