Building Regional Covariance Descriptors for Vehicle Detection

Xueyun Chen, Ren-Xi Gong, Ling-Ling Xie, Shiming Xiang, Cheng-Lin Liu, and Chun-Hong Pan

***Abstract*— We study the question of building regional covariance descriptors (RCDs) for vehicle detection from high-resolution satellite images. A unified way is proposed to build RCD features by constant convolutional kernels in the forms of 2-D masks. Two novel formulas are designed to construct different RCD types based upon one or two convolutional masks, obtaining ten novel RCD features by four simple constant convolutional masks. Experiments show that such convolutional-mask-based RCDs outperform the previous image-derivative-based RCDs, the popular local binary patterns (LBPs), the histogram of oriented gradients (HOGs), and LBP+HOG. Furthermore, feeding to nonlinear support vector machines (SVMs) of two kernel types [L_1 kernel and radial basis function (RBF)], these RCDs outperform four known deep convolutional neural networks: AlexNet, GoogLeNet, CaffeNet, and LeNet, as well as their fine-tuned models by their well-trained weights of imageNet classification. Among three popular classic classifiers we have tested in the experiments, nonlinear SVMs outperform BP and Adaboost obviously, and L_1 kernel exceeds RBF slightly.**

***Index Terms*— Deep convolutional neural networks (DCNNs), regional covariance descriptor (RCD), vehicle detection.**

I. Introduction

THE regional covariance descriptor (RCD) was first introduced by Tuzel *et al.* [1], [2] in 2006 to represent a region by the covariance matrix of image features, such as spatial location, intensity, first-order and second-order derivatives, and so on. From then on, it has been used in a wide variety of tasks, including face verification [3], human detection [4], object tracking, classifying, matching, and recognition [5]–[9]. It is known that RCD possesses a strong robustness against small disturbances [1], [2], [10]. However, we do not know its exact performance extents for general recognition and classification, and how many varieties of them can be explored. All these things remain a mystery as much as they were one decade ago.

Two factors impede the study of RCDs: one is the limited selection of its based features and the other is the uncertain choice of its distance metrics. Most researchers followed the original definition of the based feature and distance metric proposed in [2], a very few works have been done to expand its feature varieties and determine which distance metric works better.

Recently, Faulkner *et al.* [10] did a significant work to improve RCD by adding the lab color components to the based feature, and evaluating the performances of Euclidean, Log-Euclidean, and affine-invariant (the original) metrics by a broad range of geometric and photometric image transformations. They reported that Euclidean measure achieves much higher precision than the others in most situations (four out of five), including changes of brightness, Gaussian blur, Gaussian noise, and saturation. We argue that if Euclidean-like (L_1 and L_2) metric is adopted, the 2-D structure property of the covariance matrix should be ignored, which is just our way of treating RCDs in this letter.

In order to expand the forms of RCD, we proposed a unified way of constructing RCD forms by two novel formulas from one or two convolutional masks. Ten new RCD forms were built from four constant convolutional masks. Theoretically, our method leads to an unlimited way of building RCDs upon the infinite varieties of the convolutional masks.

Vehicle detection from satellite images coarsely includes two stages: the first is searching all candidate windows efficiently and the second is classifying the candidates precisely and outputting those may contain vehicles. It is the second stage that determining the precision of the detector mainly, and the feature used that deciding the accuracy (AC) of the classifier mostly.

Many features have been used in the past works of vehicle detection: multiscale histogram of oriented gradients (HOGs [18]) of color maps [11], HOG+Haar wavelets+local binary patterns (LBPs [19]) [12], pose-indexed feature [13] (an HOG-like feature), HOG+Haar wavelets features [14], deep convolutional features [15], sparse coding upon bag-of-words model [16], sparse representation of multiscale HOG [17], and so on.

We used BP, Adaboost, and nonlinear support vector machine (SVM) with two kernels to evaluate the RCD performances, just the L_1 kernel [Section IV, formula (7)] and the RBF.

Experiments upon vehicle database show that our novel RCDs outperform the previous RCDs [1], [2], [10], LBP, and HOG. Furthermore, the convolutional-mask-based RCDs + SVM(L_1) even outperform four known deep convolutional neural networks (DCNNs): AlexNet, GoogLeNet, CaffeNet, and LeNet.

The remainder of this letter is organized as follows. The previous works of RCD are presented concisely in Section II, our approach of building new RCDs is carefully explained in Section III, and implementation details and parameters are given in Section IV. Experimental results are listed in Section V. We concluded finally in Section VI.

Manuscript received July 14, 2016; revised October 17, 2016 and November 29, 2016; accepted December 27, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61661006, Grant 61561007, and Grant 91646207.

The authors are with Guangxi University, Nanning 530004, China, and also with the Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: cxy177@163.com; rxgong@gxu.edu.cn; xielingling1318@163.com; smxiang@nlpr.ia.ac.cn; liucl@nlpr.ia.ac.cn; chpan@nlpr.ia.ac-.cn).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2017.2653772

II. Previous Works of Regional Covariance Descriptor

The popular LBP and HOG features are easily to be disturbed by the position shifting and illumination varying in object detection. Based on the first-order and second-order statistical moments, RCD [1], [2] is thought to be somehow robust slightly, here we give its definition.

Given an n-dimension feature $\phi(x, y)$ of a region R, the covariance matrix of $\phi(x, y)$ of R is expressed as

$$\Lambda_R(\phi(x, y)) = \sum_{(x,y)\in R} (\phi(x, y) - \mu_R)^T (\phi(x, y) - \mu_R) \tag{1}$$

where $\mu_R = \frac{1}{|R|}\sum_{(x,y)\in R}\phi(x, y)$, $|R|$ is the number of pixels in R. When using Euclidean-like metrics, we define $\Lambda_R^-(\phi) = \text{vector}(\Lambda_R(\phi))$ as the vector composed by all nonrepeated elements of the symmetric $n \times n$ matrix Λ_R, it is easy to see that the dimension of Λ_R^- is $\frac{n\times(n+1)}{2}$.

Let $I(x,\ y)$ denote the grayscale function of the image pixel at (x,y), and I_x, I_y, I_{xx}, and I_{yy} denote the first-order or second-order differential operators of $I(x, y)$. Tuzel *et al.* [1], [2] presented the initial form of RCD as follows:

$$RCD(T) = \Lambda_R^-\left(\left[x, y, |I_x|, |I_y|, \sqrt{I_x^2 + I_y^2}\ , |I_{xx}|, |I_{yy}|, \tan^{-1}\left(\frac{|I_x|}{|I_y|}\right)\right]\right). \tag{2}$$

Faulkner *et al.* [10] proposed a new form of RCD as

$$RCD(F) = \Lambda_R^-\left(\left[x, y, r, g, b, |I_x|, |I_y|, |I_{xx}|, |I_{yy}|, |I_{xy}|, \sqrt{I_x^2 + I_y^2}, \tan^{-1}\left(\frac{|I_x|}{|I_y|}\right), l, a, b\right]\right) \tag{3}$$

where l, a, and b are the three components of LAB color space.

III. Our Approach of Improving Regional Covariance Descriptor

The importance of the convolutional kernels used in DCNN has been convincingly proved by the great progresses achieved by the deep learning methods in many artificial intelligence fields. Such kernels can be mathematically expressed by their convolutional masks. We argue that elaborated designed constant convolutional masks should have a promising performance like that in DCNN, when being used as the basic features of RCD.

We design four simple convolutional masks, named $Ci = \{Ci_x, Ci_y\}, 1 \le i \le 4$, where Ci_x and Ci_y are the two components along the x- and y-coordinate axes. Their geometrical forms are shown in Fig. 1.

In Fig. 1, C1$_x$ acts as the standard first-order differential mask I_x and C2$_x$ acts as the second-order differential mask I_{xx}. C3 is designed to detect stripes and belts, and C4 is designed to detect the corners and ends.

We introduce an operator $\odot$ denoting the mask inner product. If $\text{A} = [a_{ij}]_{n\times n}, \text{B} = [b_{ij}]_{n\times n}, \text{C} = [c_{ij}]_{n\times n}$, then

$$A \odot B = C. \tag{4}$$

Means that $\text{c}_{ij} = \text{a}_{ij}\cdot\text{b}_{ij} (1 \le i \text{ and } j \le \text{n})$. Thus, C1$_x$⊙C1$_y$ is equivalent to the derivative I_{xy}, C2$_x$⊙C2$_y$ equals I_{xxyy}, and C1$_x$⊙C2$_y$ equals I_{xyy}.

$$\begin{bmatrix}-1&-1&1&1\\-1&-1&1&1\\-1&-1&1&1\\-1&-1&1&1\end{bmatrix}\quad\begin{bmatrix}-1&1&1&-1\\-1&1&1&-1\\-1&1&1&-1\\-1&1&1&-1\end{bmatrix}\quad\begin{bmatrix}-1&1&-1&1\\-1&1&-1&1\\-1&1&-1&1\\-1&1&-1&1\end{bmatrix}\quad\begin{bmatrix}1&1&-1&-1\\-1&-1&1&1\\-1&-1&1&1\\1&1&-1&-1\end{bmatrix}$$

(a) C1x (b) C2x (c) C3x (d) C4x

$$\begin{bmatrix}-1&-1&-1&-1\\-1&-1&-1&-1\\1&1&1&1\\1&1&1&1\end{bmatrix}\quad\begin{bmatrix}-1&-1&-1&-1\\1&1&1&1\\1&1&1&1\\-1&-1&-1&-1\end{bmatrix}\quad\begin{bmatrix}-1&-1&-1&-1\\1&1&1&1\\-1&-1&-1&-1\\1&1&1&1\end{bmatrix}\quad\begin{bmatrix}1&-1&-1&1\\1&-1&-1&1\\-1&1&1&-1\\-1&1&1&-1\end{bmatrix}$$

(e) C1y (f) C2y (g) C3y (h) C4y

Fig. 1. First row lists the x-components of the four basic convolutional masks, and the second row lists their y-components, which can be got by applying a 90° clockwise turning upon the corresponding x-components. (a) C1$_x$. (b) C2$_x$. (c) C3$_x$. (d) C4$_x$. (e) C1$_y$. (f) C2$_y$. (g) C3$_y$. (h) C4$_y$.

We propose two novel formulas to construct RCDs, the first is that based on one convolutional mask as

$$RCD(Ci) = \Lambda_R^-\left(\left[x, y, Ci_x, Ci_y, Ci_x \odot Ci_y, \sqrt{Ci_x^2 + Ci_y^2}\ ,\quad \tan^{-1}\left(\frac{Ci_x}{Ci_y}\right)\right]\right) \tag{5}$$

where $1 \le i \le 4$, the second is that based on two convolutional masks as

$$\begin{aligned}&RCD(Ci, Cj)\\&= \Lambda_R^-\Bigg(\Bigg[x,\ y,\ Ci_x,\ Ci_y,\ Cj_x,\ Cj_y,\\&\qquad Ci_x \odot Cj_y, Ci_y \odot Cj_x, \sqrt{Ci_x^2 + Ci_y^2}, \sqrt{Cj_x^2 + Cj_y^2},\\&\qquad \sqrt{(Ci_x \odot Cj_y)^2 + (Ci_y \odot Cj_x)^2},\quad \tan^{-1}\left(\frac{Ci_y}{Ci_x}\right),\\&\qquad \tan^{-1}\left(\frac{Cj_y}{Cj_x}\right),\quad \tan^{-1}\left(\frac{Ci_y \odot Cj_x}{Ci_x \odot Cj_y}\right)\Bigg]\Bigg)\end{aligned} \tag{6}$$

where $1 \le i,\ j \le 4$, and $i \ne j$. By (5) and (6), four-novel *RCD(Ci)* and six-new *RCD(Ci, Cj)* are constructed, respectively.

IV. Implementation Details

Here, we give some important implementation details and parameters of our approach.

When computing the RCD's basic feature $\phi(x, y)$, following Dalal and Triggs's suggestion [18], we compute the value of the convolutional masks upon three RGB color channels and output the maximal magnitude, then normalizing all components of $\phi(x,\ y)$ of the Region R into an unified range [0,1]. Finally, the RCDs are constructed upon the normalized $\phi(x, y)$. For RCD(T), RCD(F), RCD(Ci), and RCD(Ci, Cj), the dimensions of their $\phi(x, y)$ are: 8, 15, 7, and 16, respectively.

LBP feature is computed just as described by Ojala *et al.* [19], where P = 8, and R = 1.5. All binary patterns are divided into 59 classes, meaning that the patterns with more than two 0–1 jumps belong to the nonuniform class. HOG feature is computed as suggested by Dalal and Triggs [18]. The [0, 180°] orienting range is divided into nine bins.

Every sample image is divided by five spatial pyramid grids, the number of the blocks is: $1\times1+2\times2+3\times3+4\times4+5\times5 = 55$. So, the total dimension of LBP, HOG, RCD(T), RCD(F), RCD(Ci), and RCD(Ci, Cj) are: $59\times55 = 3245$, $9\times55 = 495$, $8\times9/2\times55 = 1980$, $15\times16/2\times55 = 6600$, $7\times8/2\times55 = 1540$, and $16\times17/2\times55 = 7480$, respectively. The default feature norm is L1-norm. The influence of different feature norms is listed in Table VI.

Let x and z denote two points in the feature space and $K(x,z)$ denotes the kernel function of the SVM classifier. The L_1 kernel and RBF kernel are defined as follows

$$L_1 : K(x,z) = e^{-\frac{\gamma}{nDim}\|x-z\|_1} \tag{7}$$

$$RBF : K(x,z) = e^{-\frac{\gamma}{nDim}\|x-z\|_2^2} \tag{8}$$

where γ is the kernel parameter and $nDim$ is the dimension of the feature space. We use nonlinear SVM with L_1 kernel as the default classifier, set $\gamma = 45$, and the number of support samples $nsv = 800$ as the default parameters.

We used four known DCNN models: AlexNet [20], GoogLeNet [21], CaffeNet [22], and LeNet [23]. Their frames were downloaded from https://github.com/BVLC/caffe. Except of LeNet, their weight models of ImageNet database were downloaded from https://dl.caffe.berkeleyvision.org/.

The structural parameters of these known DCNN models are kept unchanged throughout our experiments, such as the parameters of the layers, of the convolutional and pooling kernels (number, type, size, stride, padding size, and so on). But, the data layer was revised slightly to support our vehicle database and linearly transforming the 48 × 48 image of the samples into the required size (227 × 227 or 224 × 224 for them). The learn_rate = 0.001, weight_decay = 0, and momentum = 0. Training process continued until the error rate was less than 0.001 or smaller. Testing was executed every 100 iterations, using the "Softmax" as the final classifiers.

We defined the deep convolutional feature (DCF) of a DCNN as the output of the highest pooling layer of the net. For AlexNet, GoogLeNet, and CaffeNet, the dimension of their highest pooling layer is 9216, 1024, and 9216, respectively. The source codes of Caffe [22] include a file named "extra -ct features.cpp." It enables us to extract the DCFs from the above three known DCNN models and evaluate their performance by nonlinear SVM in Table V.

V. Experiment

The searching stage of vehicle detection inevitably produces many similar positive samples, a huge quantity of meaningless negative samples, such repetitive or meaningless samples contribute very less to the detector training process. To avoid such a fault, we construct our database by many unique vehicle samples and difficult negative samples that containing complicated textures or vehicle-like objects. Training database includes 1500 positive samples and 1500 negative samples. Testing database contains the same numbers of different positive and negative samples. All image patches are cut from 111 high-resolution satellite images at San Francisco city streets via the Google-Earth software (Fig. 2).

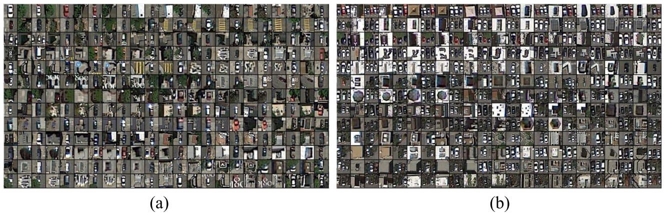

Fig. 2. Images of the samples used in our experiments. (a) Partial training samples. (b) Partial test samples.

TABLE I
Accuracies of Four known Deep convolutional Neural Nets

Training way	Deep convolutional neural nets			
	AlexNet	GoogLeNet	CaffeNet	LeNet
Non-fine-tuned	0.9286	0.9226	0.9236	0.9190
Fine-tuned	0.9440	0.9523	0.9533	-

The precision rate, recall rate, and AC are defined as

$$\mathrm{PR} = \frac{\text{number of detected vehicles}}{\text{number of detected objects}} \tag{9}$$

$$\mathrm{RR} = \frac{\text{number of detected vehicles}}{\text{number of all vehicles}} \tag{10}$$

$$\mathrm{AC} = 1.0 - \frac{\min\{\text{number of misclassified samples}\}}{\text{number of all vehicles}}. \tag{11}$$

In formula (11), the misclassified samples include miss-detected vehicles and false alarms. AC reflects the best performance that a detector can achieve.

Fig. 2 lists some typical sample images from our training and test databases. They are 48 × 48 colorful images, with normalized scale and orientation.

Table I lists the accuracies of the four known DCNNs upon our vehicle test database, where AlexNet is the nine-layer model described in [20], which had won the ILSVRC-2012 classification challenge, GoogLeNet is a large-scale deep neural networks [21] with about 40 layers, won the ILSVRC-2014 classification challenge, CaffeNet [22] is a revised net model from AlexNet, where pooling is done before normalization, and LeNet is the net-model described in [23], which is known to be working well on the hand-written digit classifying task. The nonfine-tuned way is the normal way that starting from small randomly initialized weight parameters and then trained by the stochastic gradient descent method, and the fine-tuned way is that tuned upon a preloaded well-trained weight model of imageNet database. Table I shows that fine tuning obviously improves the performances of the three net models. It also greatly accelerates the training processes. For an instance, GoogLeNet needs about 24–30 normal training hours, but only 4–5 fine-tuning hours.

Table II lists the test accuracies of nonlinear SVM classifier (L_1 kernel, $\gamma = 45$, and $nsv = 800$), using 15 different features. It shows clearly that RCD(Ci,Cj)+SVM(L_1) outperforms the above four known DCNN models. RCD(Ci,Cj) outperforms other features by an obvious margin. Among all RCD features, RCD(C2,C3) is the best, but RCD(C1,C2) achieves the second best by an almost ignorable margin (0.0006).

TABLE II

PERFORMANCES OF HOG, LBP, LBP+HOG, AND RCDs+SVM(L_1)

Features	Dim	Accuray
HOG	495	0.8833
LBP	3245	0.9190
LBP+HOG	3740	0.9346
RCD(T)	1980	0.9196
RCD(F)	6600	0.9363
RCD(C1)	1540	0.9430
RCD(C2)	1540	0.9316
RCD(C3)	1540	0.9480
RCD(C4)	1540	0.8913
RCD(C1,C2)	7480	0.9590
RCD(C1,C3)	7480	0.9543
RCD(C1,C4)	7480	0.9540
RCD(C2,C3)	7480	0.9596
RCD(C2,C4)	7480	0.9573
RCD(C3,C4)	7480	0.9580

TABLE III

PERFORMANCES OF RCD FEATURES WITH LBP+HOG, CLASSIFIED BY SVM(L_1)

Features	Dim	Accuracy
LBP+HOG+RCD(T)	5720	0.9476
LBP+HOG+RCD(F)	10340	0.9510
LBP+HOG+RCD(C1)	5280	0.9553
LBP+HOG+RCD(C2)	5280	0.9496
LBP+HOG+RCD(C3)	5280	0.9546
LBP+HOG+RCD(C4)	5280	0.9536
LBP+HOG+RCD(C1,C2)	11220	0.9623
LBP+HOG+RCD(C1,C3)	11220	0.9613
LBP+HOG+RCD(C1,C4)	11220	0.9626
LBP+HOG+RCD(C2,C3)	11220	0.9610
LBP+HOG+RCD(C2,C4)	11220	0.9590
LBP+HOG+RCD(C3,C4)	11220	0.9580

TABLE IV

AC OF VARIOUS CLASSIFIERS IN THE LBP+HOG+RCD(C1,C4) FEATURE SPACE

SVM(L_1)	SVM(RBF)	BP(8,2)	BP(16,2)	BP(32,2)	Adaboost
0.9626	0.9533	0.9413	0.9380	0.9406	0.9150

TABLE V

PERFORMANCE OF DCFS + SVM (L_1)

Nets	layer	Dim	AC	AC *	AC **
AlexNet	pool5	9216	0.9186	0.9423	0.9263
GoogLeNet	pool5-7x7-s1	1024	0.9220	0.9423	0.9493
Caffenet	pool5	9216	0.9180	0.9590	0.9343

AC : Accuracy of the Nets training normally.
AC*: Accuracy of the Nets loaded the imagenet weight-models without fine-tuning.
AC**: Accuracy of the Nets fune-tuned from the imagenet weight-models.

AC : Accuracy of the Nets training normally.
AC*: Accuracy of the Nets loaded the imagenet weight-models without fine-tuning.
AC**: Accuracy of the Nets fune-tuned from the imagenet weight-models.

TABLE VI

INFLUENCE OF FEATURE NORM UPON THE AC OF SVM(L_1) CLASSIFIER

		Feature Norm		
Features	Dim	L1-norm	L2-norm	L1-sqrt
HOG	495	0.8833	0.8853	0.8820
LBP	3245	0.9190	0.9290	0.9280
LBP+HOG	3740	0.9346	0.9373	0.9350
RCD(T)	1980	0.9196	0.9233	0.9216
RCD(F)	6600	0.9363	0.9376	0.9333
RCD(C3)	1540	0.9480	0.9493	0.9446
RCD(C2,C3)	7480	0.9596	0.9590	0.9526
LBP+HOG+RCD(T)	5720	0.9475	0.9473	0.9503
LBP+HOG+RCD(F)	10340	0.9510	0.9513	0.9493
LBP+HOG+RCD(C1)	5280	0.9553	0.9520	0.9563
LBP+HOG+RCD(C1,C4)	11220	0.9623	0.9610	0.9620

TABLE VII

INFLUENCE OF KERNEL PARAMETER UPON THE AC OF LBP + HOG + RCD(C1,C4) + NONLINEAR SVM

	kernel parameter: γ					
Kernel	20	30	40	50	60	70
L_1	0.9593	0.9586	0.9613	0.9630	0.9633	0.9640
RBF	0.9480	0.9533	0.9503	0.9520	0.9546	0.9523

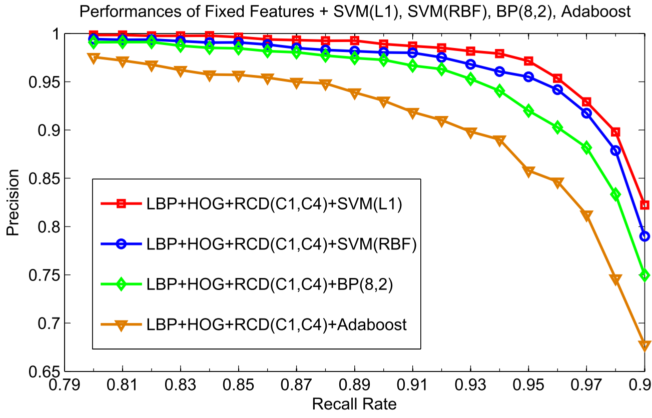

Fig. 3. P–R curves of nonlinear SVM versus BP and Adaboost on our vehicle test database.

Table III lists the accuracies of LBP+HOG+different RCDs+SVM classifier (L_1 kernel, $\gamma = 45$, and $nsv = 800$), it shows that the combination of RCD with LBP + HOG increases the former's AC obviously. Among all features, LBP+HOG+RCD(C1,C4) is the best one, and LBP+HOG+RCD(C1,C2) achieves the second best again with an ignorable margin (0.0003). It seems that RCD(C1,C2) has a better performance stability than other RCDs.

In Table IV and Fig. 3, BP(n,2) means the three-layer back-propagation neural network with n hidden neurons and two output neurons, where the rectified linear unit function is used for the neurons of the hidden layer, and Tanh function for the output layer. Adaboost used weak classifiers based on all components of the input feature, with their threshold parameters and directions optimized in the training set in advance. Table IV clearly exhibits that SVM outperforms BP(n,2), and BP(n,2) outperforms Adaboost.

In Table V, the DCFs are extracted from the highest pooling layers of the DCNNs, and the dimension of the highest pooling layer is decided by its number of maps and its map size. All DCFs are sent to a nonlinear SVM(L_1, $\gamma = 45$, and $nsv = 800$) classifier to do training and testing. Table V clearly shows that these DCFs could not outperform LBP + HOG + RCD(Ci,Cj). It seems that those DCFs for imageNet classification are not suitable for vehicle detection.

Table VI shows the slight influence of the tree feature norms upon the accuracies of fixed features + SVM(L_1).

Table VII lists the influence of the selection of kernel parameter γ upon the accuracies of LBP + HOG + RCD(C1, C4) + nonlinear SVM. It reveals that L_1 kernel

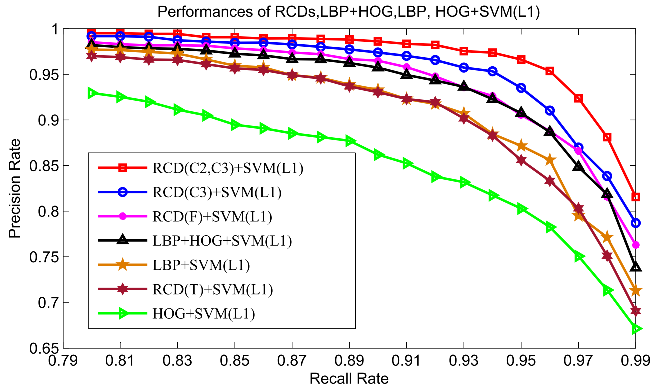

Fig. 4. P–R curves of seven vehicle detectors: RCD(C2, C3), RCD(C3), RCD(F), RCD(T), LBP + HOG, LBP, and HOG with SVM(L_1) classifier.

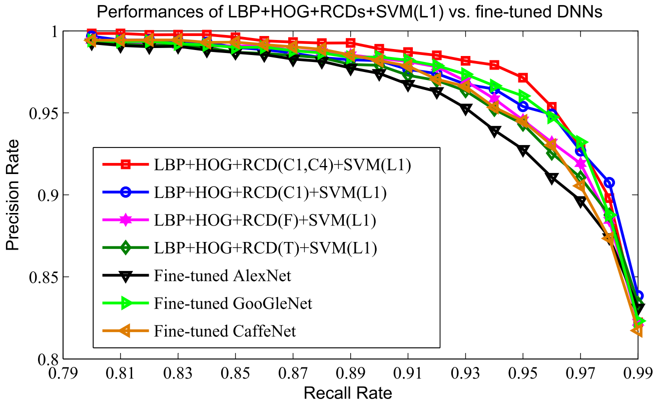

Fig. 5. Performances of fixed features + SVM versus fine-tuned DCNNs.

always outperforms RBF kernel, and the γ kernel parameter has a relative small influence upon the classifier AC.

Fig. 3 shows that SVM (L_1) outperforms SVM (RBF), and the later outperforms BP and Adaboost.

Fig. 4 shows that RCD (C2, C3) outperforms RCD (C3), RCD (F), RCD (T), LBP + HOG, LBP, and HOG.

Fig. 5 shows that LBP + HOG + RCD(C1, C4) + SVM (L_1) outperforms fine-tuned AlexNet, GoogLeNet, and CaffeNet obviously. As a contrast, the GPU (Tesla K20, 2496 kernels) training time of the DCNNs needs at least 4–30 h, while the CPU (Intel core i5, four kernels, 2.6 GHz) training time of the fixed features + nonlinear SVM never exceeds 370 s.

VI. Conclusion

In order to improve the AC of vehicle detection, we study the problem of expanding RCD feature, proposed two novel formulas to construct RCD based on one or two constant convolutional masks, building ten novel RCDs: four RCD(Ci) and six RCD(Ci, Cj) ($1 \leq i, j \leq 4$). They achieved a much higher performance than the popular features (HOG, LBP, LBP + HOG, and previous RCDs), and some of them even exceed four known DCNNs (AlexNet, GoogLeNet, CaffeNet, and LeNet) when using SVM (L_1 kernel) classifier. Among all RCDs, RCD(C1, C2) is specially recommended for its better robustness. Compared with DCNNs, classifier based on hand-crafted descriptors has the obvious advantage of swiftness, simplicity, and convenience. Such descriptors have infinite possibilities to be improved by human wisdom, and being suitable for a wide range of recognition tasks.

References

[1] O. Tuzel, F. Porikli, and P. Meer, “Region covariance: A fast descriptor for detection and classification,” in *Proc. Eur. Conf. Comput. Vis.*, Lecture Notes in Computer Science, vol. 3952. 2006, pp. 589–600.

[2] O. Tuzel, F. Porikli, and P. Meer, “Pedestrian detection via classification on Riemannian manifolds,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1713–1727, Oct. 2008.

[3] C. Huang, S. Zhu, and K. Yu. (2012). “Large scale strongly supervised ensemble metric learning, with applications to face verification and retrieval.” [Online]. Available: https://arxiv.org/abs/1212.6094

[4] J. Yao and J. M. Odobez, “Fast human detection from videos using covariance features,” in *Proc. 8th ECCV Workshop Vis. Surveill.*, Oct. 2008, pp. 1–8.

[5] C. Undurraga and D. Mery, “Improving tracking algorithms using saliency,” in *Proc. 16th Iberoamerican Congr. Pattern Recognit.*, vol. 7042. 2011, pp. 141–148.

[6] W. Ayedi, H. Snoussi, and M. Abid, “A fast multi-scale covariance descriptor for object re-identification,” *Pattern Recognit. Lett.*, vol. 33, no. 14, pp. 1902–1907, 2012.

[7] E. Erdem and A. Erdem, “Visual saliency estimation by nonlinearly integrating features using region covariances,” *J. Vis.*, vol. 13, no. 4, pp. 1–20, Mar. 2013.

[8] L. Qin, H. Snoussi, and F. Abdallah, “Adaptive covariance matrix for object region representation,” *Proc. SPIE*, vol. 8878, art no. 887848, Jul. 2013.

[9] P. C. Cargill, C. U. Rius, D. M. Quiroz, and A. Soto, “Performance evaluation of the covariance descriptor for target detection,” in *Proc. Int. Conf. Chilean Comput. Sci. Soc.*, Nov. 2009, pp. 133–141.

[10] H. Faulkner *et al.*, “A study of the region covariance descriptor: Impact of feature selection and image transformations,” in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, Nov. 2015, pp. 1–8, doi: 10.1109/DICTA.2015.7371222.

[11] A. Kembhavi, D. Harwood, and L. S. Davis, “Vehicle detection using partial least squares,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1250–1265, Jun. 2011.

[12] H. Grabner, T. T. Nguyen, B. Gruber, and H. Bischof, “On-line boosting-based car detection from aerial images,” *J. Photogramm. Remote Sens.*, vol. 63, no. 3, pp. 382–396, 2008.

[13] K. Ali, F. Fleuret, D. Hasler, and P. Fua, “A real-time deformable detector,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 225–239, Feb. 2012.

[14] P. Liang, G. Teodoro, H. Ling, E. Blasch, G. Chen, and L. Bai, “Multiple kernel learning for vehicle detection in wide area motion imagery,” in *Proc. 15th Int. Conf. Inf. Fusion*, Jul. 2012, pp. 1629–1636.

[15] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, “Vehicle detection in satellite images by hybrid deep convolutional neural networks,” *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1797–1801, Oct. 2014.

[16] H. Sun, X. Sun, H. Wang, Y. Li, and X. Li, “Automatic target detection in high-resolution remote sensing images using spatial sparse coding bag-of-words model,” *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 1, pp. 109–113, Jan. 2012.

[17] Z. Chen *et al.*, “Vehicle detection in high-resolution aerial images via sparse representation and superpixels,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 103–116, Jan. 2016.

[18] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2005, pp. 886–893.

[19] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. NIPS*, 2012, pp. 1097–1105.

[21] C. Szegedy *et al.* “Going deeper with convolutions.” Unpublished paper, 2014. [Online]. Available: https://arxiv.org/abs/1409.4842

[22] Y. Jia *et al.* “Caffe: Convolutional architecture for fast feature embedding.” Unpublished paper, 2013. [Online]. Available: https://arxiv.org/abs/1408.5093

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.