Vehicle Detection Using Local Size-Specific Classifiers

SeungJong NOH(¹), Member and Moongu JEON(²), Nonmember

SUMMARY As the number of surveillance cameras keeps increasing, the demand for automated traffic-monitoring systems is growing. In this paper, we propose a practical vehicle detection method for such systems. In the last decade, vehicle detection mainly has been performed by employing an image scan strategy based on sliding windows whereby a pre-trained appearance model is applied to all image areas. In this approach, because the appearance models are built from vehicle sample images, the normalization of the scales and aspect ratios of samples can significantly influence the performance of vehicle detection. Thus, to successfully apply sliding window schemes to detection, it is crucial to select the normalization sizes very carefully in a wise manner. To address this, we present a novel vehicle detection technique. In contrast to conventional methods that determine the normalization sizes without considering given scene conditions, our technique first learns local region-specific size models based on scene-contextual clues, and then utilizes the obtained size models to normalize samples to construct more elaborate appearance models, namely local size-specific classifiers (LSCs). LSCs can provide advantages in terms of both accuracy and operational speed because they ignore unnecessary information on vehicles that are observable in faraway areas from each sliding window position. We conduct experiments on real highway traffic videos, and demonstrate that the proposed method achieves a 16% increased detection accuracy with at least 3 times faster operational speed compared with the state-of-the-art technique.

key words: intelligent transportation systems, vehicle detection, local size-specific classifier

1. Introduction

Vehicle detection is one of the most important components of automated traffic monitoring systems because it can provide useful information on regions of interest (ROIs) for higher video analysis techniques. To accurately detect vehicles, we should effectively address three major challenges: location ambiguity, size ambiguity, and appearance variation of vehicles [1], [2]. These challenges have been overcome by sliding window based detection methods [2]–[5] because of their superior performance [6], [7] compared with other approaches employing branch-and-bound search [8], [9] and coarse-to-fine search [10]–[12]. Recently, more sophisticated techniques have also been proposed based on convolutional neural networks [13], [14] and 3D model structures [12], [15], [16]. However, these techniques are not appropriate for practical applications because they require high resolution images and computational costs.

Although considerable progress has been made [2], [5], sliding window schemes still encounter many problems when dealing with appearance variation of vehicles. To handle severe appearance variation, conventional approaches use an image classifier learned from a large-scale dataset. Specifically, for an input frame, sliding-window schemes first define a set of bounding-boxes by deforming the search window shape at each image position according to particular vehicle size models, and then produce the final detection results by locally optimizing classification responses between a pretrained image classifier and subimages captured from each bounding-box area. Learning an appropriate classifier for this scheme requires the selection of a normalization size to regularize the scales and aspect ratios of samples because this significantly impacts the performance of the vehicle detector. For instance, in a scene with large vehicle scales, detection quality can be degraded considerably when applying a classifier trained using samples normalized to a small size (Sect. 2). In existing methods [2]–[6], normalization sizes are set manually based on expert’s experiences, and therefore detection accuracy varies unstably according to specific scene conditions.

To overcome this drawback, we propose an effective vehicle detection strategy that automatically selects normalization sizes depending on scene-contextual information rather than user input. The fundamental principle of the proposed approach is that vehicle location, size and appearance clues are closely correlated, and they should therefore be jointly modeled for accurate detection. To accomplish this, we train an image classifier for each vehicle size pattern observable at each local area. For a given scene, we first generate semantic region models (SRMs) in Fig. 1 (a) by analyzing the movement patterns of vehicles [2] and then register a set of size models for every SRM. Next, using training samples whose sizes are normalized based on the obtained size models, we construct more tailored image-classifiers, namely, local size-specific classifiers (LSCs). Figure 1 (b) shows several examples of LSCs learned for a fixed camera scene. The main difference between LSCs and conventional image classifiers [3], [4], [16]–[19] is that LSCs can provide not only appearance information but also location and size information. Combined with the sliding window technique, this property can offer several important advantages for practical applications.

Most importantly, the additional information obtained
Fig. 1 Depiction of LSCs: (a) SRMs (colored ellipses) for vehicle detection; (b) several examples of LSCs registered at $R_1$ and $R_2$. Each LSC represents appearance models for vehicles with different sizes at local regions $R_1$ and $R_2$.

by LSCs allows systems to achieve more accurate vehicle detection. As shown in Fig. 1 (b), each LSC provides well separated appearance models for vehicles that are observed at faraway locations or whose sizes are very dissimilar. Thus, by applying LSCs to sliding window schemes, we can obtain more reliable vehicle detection performance. In addition, LSCs help systems to operate detectors with lower computational costs. In conventional techniques [2]–[6], because appearance and size models are built independently, redundant combinations between them (e.g., a sedan appearance and a bus size) are also exhaustively analyzed. In contrast, our method avoids such unnecessary computations by jointly considering appearance and size information using LSCs. From experiments on real traffic videos, we will compare the proposed method with a state-of-the-art sliding window technique in [2]. Also, to investigate the relationship between normalization sizes and vehicle scales, we measure the detection accuracy at three representative regions considering the scales of the vehicles for the CCTV camera as shown in Fig. 2. More detailed simulation results can be found in Sect. 3.

It may be suggested in Fig. 2 (d) that smaller normalization sizes lead to a higher detection accuracy for small scale vehicles. However, the results for 48×48 and 64×64 indicate that overly small normalization sizes result in poor detection accuracy for large scales vehicles. In contrast, when samples are normalized to sizes larger than 80×80, the detectors achieve better accuracy for large vehicles in the scene. However, excessively large normalization sizes can significantly deteriorate detection performance. In Fig. 2 (d), the results of the 96×96 size achieve a lower accuracy than those of the 80×80 size across all vehicle scales. This is because, although large normalization sizes contribute to create more informative classifiers, they also increase the minimum size of detectable vehicles.

From these analysis results, we conclude that sample normalization sizes must be determined very carefully based on local region-specific scene conditions. The proposed LSCs are designed to solve this problem. In the following sections, we provide a more detailed explanation of each component of our approach.

2. Proposed Method

To verify the significance of LSCs, we first analyze the effects of sample normalization sizes on vehicle detection. For our experiments, we resize the training samples to four normalization sizes: 48×48, 64×64, 80×80 and 96×96 pixels. Then, for each normalization size, we train support vector machine (SVM) classifiers with histogram of oriented gradient (HOG) [4] features using the size-regularized samples. We implement the detectors based on the adaptive sliding window technique in [2]. Also, to investigate the relationship between normalization sizes and vehicle scales, we measure the detection accuracy at three representative regions considering the scales of the vehicles for the CCTV camera as shown in Fig. 2. More detailed simulation results can be found in Sect. 3.

The remainder of this paper is organized as follows. In Sect. 2, we explain the proposed vehicle detection strategy in detail. Next, we show several simulation results in Sect. 3, and then provide conclusions and future work in Sect. 4.

2.1 Learning LSCs

The first step for training LSCs entails collecting vehicle sample images from a scene (Sect. 2.1.1). Next, we register a set of size-models at each SRM using the obtained sample images (Sect. 2.1.2), and then build LSCs for these size-models (Sect. 2.1.3). We perform vehicle detection by jointly analyzing the appearance and size information of vehicles using the constructed LSCs (Sect. 2.2).

2.1.1 Constructing Training-Sets for Each Local Region

In general, the appearance variation of vehicles arises from pose changes, intra-class variation (e.g., sedan1, sedan2 and
where each sample is extracted. Let how much SRMs are spatially closely related with positions of SRMs using: the probabilities that samples are extracted from each area assign the gathered samples to SRMs (Fig. 3), we estimate \( \phi \) puts of intervention by using background subtraction [20]. Next, to ff density function (pdf) of Under the assumption that the underlying probability den-

Fig. 3 Assigning samples to SRMs: (a) the violet-box represents a region at which a sample image is extracted based on background subtraction [5] and whose center coordinate is \( s \). The sample for the violet-box is assigned to SRM \( R_1 \) in Fig. 1 because it gives a small Eq. (3) value for \( R_1 \); (b) examples of samples assigned to \( R_1 \).

sedan3), and inter-class variation (e.g., sedan, truck and bus). To appropriately address the appearance variation caused by pose changes, we first construct SRMs for a given scene (Fig. 1 (a)) [2]. In a traffic monitoring video recorded from a fixed CCTV camera, the vehicle poses are determined by their moving directions. Thus, we can regard a SRM as a region that is consistent with respect to vehicle poses because its area is defined by clustering moving direction and location features of vehicles. Based on this fact, we constrain the pose changes of vehicles while learning appearance models by creating LSCs.

After the SRMs are built, we use a semi-automated algorithm for dataset generation [5] whereby training samples are effectively collected from a scene with minimal human intervention by using background subtraction [20]. Next, to assign the gathered samples to SRMs (Fig. 3), we estimate the probabilities that samples are extracted from each area of SRMs using:

\[
P(SRM|\text{sample}) = \frac{P(SRM) \cdot P(\text{sample}|SRM)}{P(\text{sample})}. \tag{1}
\]

Under the assumption that the underlying probability density function (pdf) of \( \text{sample} \) is uniformly distributed, Eq. (1) can be converted into:

\[
\varphi(\text{sample}, \text{SRM}) = P(\text{SRM}) \cdot P(\text{sample}|\text{SRM}). \tag{2}
\]

In general scenes, it is hard to directly compute the outputs of \( \varphi(\cdot) \). Therefore, to estimate its values, we investigate how much SRMs are spatially closely related with positions where each sample is extracted. Let \( s = (x, y)^T \) denote an image coordinate in a video at which a sample is extracted (Fig. 3 (a)). Then, we can approximately compute \( \varphi(\cdot) \) by the following equation:

\[
\phi(s, \text{SRM}) = w_{\text{SRM}} \cdot \mathcal{N}(s; \mu_{\text{SRM}}, \Sigma_{\text{SRM}}), \tag{3}
\]

where \( \mathcal{N}(\cdot) \) indicates a bivariate normal density function, and \( w_{\text{SRM}}, \mu_{\text{SRM}} \) and \( \Sigma_{\text{SRM}} \) denote a weight, a mean vector and a covariance matrix for a SRM, respectively. These statistics can be automatically leaned by analyzing position and movement patterns of vehicles in a scene, and each of them determines sizes, locations and orientations of SRMs, respectively [2].

Because vehicles show more similar appearance and size characteristics in closer areas, we can consider that Eq. (3) quantifies the degree of spatial similarity between the samples and SRMs. In this work, we calculate outputs of Eq. (3) for all possible pairs of each sample and each SRM. To learn LSCs, a sample is assigned to the \( N_S \) number of SRMs that give the highest \( \phi(\cdot) \) scores (Fig. 3 (b)). Here, \( N_S \) is the trade-off parameter balancing classifier accuracy with learning efficiency. More specifically, a large value of \( N_S \) can decrease the accuracy of classifiers because it increases the probability that sample images extracted at faraway locations are used to train the same classifiers. On the other hands, a small value of \( N_S \) requires the collection of a lot of training samples because it reduces the probability that samples are assigned to each SRM. From experiments on real traffic videos, we found that the best performance is obtained when \( N_S \) is set to 2–4 for general scenes where 30–40 SRMs are generated (Fig. 6).

2.1.2 Learning Size Models for Each Local Region

To create size models for vehicles in road areas, we analyze the scale and aspect ratio patterns of the vehicle sample images which are assigned to each SRM (Fig. 3 (b)). In a fixed scene, the location and pose of a vehicle influences its scale and aspect ratio, respectively, and vehicle types are directly related to both scale and aspect ratio. However, in a sufficiently small SRM (e.g., 50×50 pixels), because vehicles are shown at very similar locations in limited poses, we can ignore size changes caused by unconstrained vehicle locations and poses [2]. Therefore, in this work, we train SRM-specific size models to incorporate the assumption that vehicle sizes are affected only by diverse types of vehicles.

Let \( S_{SRM} = \{s_{\text{samples}}^{k=1,...,N_S}\} \) denote a set of training samples assigned to a SRM. To produce size models, we first compute the following feature vectors for each \( \text{samples}_k \):

\[
r_k = (s_{c_k}, a_{r_k})^T, \tag{4}
\]

where \( s_{c_k} \) and \( a_{r_k} \) represent the scale and aspect ratio of \( \text{samples}_k \), respectively, and are calculated from the widths and heights of samples as:

\[
s_{c_k} = \text{height}_k / \text{width}_k, \tag{5}
\]

\[
a_{r_k} = \text{height}_k / \text{width}_k. \tag{6}
\]

Next, we cluster the computed size features \( \{r_{k=1,...,N_S}\} \) using the basic sequential clustering algorithm [21]. We denote a set of cluster mean-vectors after the \( (k-1) \)-th feature vector \( r_{k-1} \) is processed as:

\[
C_{k-1} = \{\bar{r}_m = (\bar{s}_{c_m}, \bar{a}_{r_m})^T\}. \tag{7}
\]

In our sequential clustering, each \( r_k \) is evaluated based on Eq. (8) and Eq. (9) to investigate matching responses to already created clusters:
Here, \( \tau \) indicates the best matching size model for \( \text{sample}_k \), and \( D_M(\cdot) \) denotes the Mahalanobis distance from size vector \( \text{vector}_k \) (Eq. (4)) to a cluster centroid \( \overline{\text{vector}}_m \) with covariance \( \Sigma_m \):

\[
D_M(\text{vector}_k; \overline{\text{vector}}_m, \Sigma_m) = \sqrt{(\text{vector}_k - \overline{\text{vector}}_m)^T \Sigma_m^{-1} (\text{vector}_k - \overline{\text{vector}}_m)}.
\]

(11)

For each size model \( \overline{\text{vector}}_m \), we use the redistributed samples as positive images to learn a LSC, and we additionally extract 1–2 negative images per redistributed sample from the background. Next, we normalize each positive/negative image so that its size is to be identical to the size model corresponding to the related redistributed sample. Then, using all of the obtained training images, we produce a LSC for \( \overline{\text{vector}}_m \) by training a HOG-SVM classifier [4] (Fig. 4). Although we use the HOG-SVM classifier in this work, appearance models can also be built using other image classifiers such as cascades of boosted classifiers [22]–[24], 2D shape exemplars [17], [25], deformable part models [18] and convolutional neural networks [19]. However, we do not employ such classifiers because they require much longer training time and more complex implementation details.

As specified in Sect. 2.1.2, in a sufficiently small SRM, size models are built depending only on the diverse types of vehicles. Therefore, LSCs can offer well-separated appearance models for various sub-classes of vehicles with very different sizes. In Figs. 4 (b) and (d), we present several examples of LSCs. They demonstrate that LSCs effectively address not only inter-class variation caused by different vehicle types but also intra-class variation among similar vehicle types.

2.2 Vehicle Detection Using LSCs

In practical CCTV-based applications, object detection has been mainly performed using sliding window techniques. The classical approach can only handle scale ambiguity of objects [4], [6], [22] and thus it has been used to detect object classes such as humans and faces whose aspect ratio changes are not severe in a scene. However, this approach is not appropriate for traffic monitoring systems because both scales and aspect ratios vary significantly for vehicles. To overcome this drawback, adaptive sliding window methods in which vehicle sizes are investigated based on particular size models have been proposed [2], [5]. However, this scheme also fails in achieving accurate detection results because it ignores the relationship between the appearance and...
size of vehicles. For instance, it can cause many false alarms when appearance models for large vehicles and size-models for small vehicles are used together, or when the converse situations occur.

On the other hand, the proposed detection method jointly investigates appearance and size patterns of vehicles based on LSCs. This strategy can provide much higher detection accuracy and operation speed because it prevents systems from analyzing redundant combinations on vehicle appearances and sizes at each search window position.

### 2.2.1 Initial Detection

To improve computational efficiency, we first extract ROIs in a frame using background subtraction [20] and then apply a search window to each ROI. More explicitly, for a given ROI $\mathcal{F}$, we first find the most spatially closely related SRM by:

$SRM_{\mathcal{F}} = \arg \min_{SRM \in \xi} \phi(s_{\mathcal{F}}, SRM), \tag{12}$

where $\xi$ indicates a set of SRMs constructed in a scene and $\phi(\cdot)$ is the function defined in Eq. (3). $s_{\mathcal{F}}$ denotes a center coordinate of $\mathcal{F}$. In the subsequent procedure, we obtain the necessary data for detection from LSCs registered at $SRM_{\mathcal{F}}$ under the assumption that $\mathcal{F}$ has the same properties as $SRM_{\mathcal{F}}$.

Once $SRM_{\mathcal{F}}$ is selected, we conduct the sliding window scheme specified in Algorithm 1 to compute initial detection responses near the foreground region $\mathcal{F}$. In local areas in a scene, when an appearance model for small vehicles (e.g., the LSC in Fig. 4 (d)) is applied to detect large vehicles (e.g., vehicles in Fig. 4 (a)), numerous false positive detections can be generated inside the regions of the large vehicles because the appearance model can be matched to some parts of the large vehicles (Fig. 5 (d)). To rectify this, we apply the LSCs of $SRM_{\mathcal{F}}$ to the ROI $\mathcal{F}$ in a descending order of the scales stored in the related size models.

Let $\mathcal{L}$ denote a set of LSCs built for $SRM_{\mathcal{F}}$ and sorted in a descending order of scales, and $\tau_O$ denote a user adjustable parameter ranging from 0 to 1 (line 1 in Algorithm 1). As the first step to obtain detection responses for a LSC in $\mathcal{L}$, we deform the shape of a search window based on its size model (line 4), and define candidate detections $d' \in \mathcal{D}$ by locating the search window at each position in $\mathcal{F}$ (line 7–9). Then, to avoid the detection responses generated by small scale LSCs inside regions of large vehicles, we remove the candidate detections whose bounding box areas are too overlapped with those of already generated initial detections (line 11–15). Next, after the classification scores are evaluated for the remaining candidate detections (line 16–18), we obtain the initial detection responses for a LSC (line 20).

In Algorithm 1, the parameter $\tau_O$ controls the degree of overlap between detection responses for different scales of vehicles. By setting $\tau_O$ to small values, we can reduce the chances of false positive detections corresponding to some parts of large vehicles. However, a value that is too small can cause the algorithm to miss small vehicles that are located close to large vehicles (Fig. 5 (a)). On the other hands, if $\tau_O$ becomes large, although detection sensitivity for small vehicles can be improved, overall detection quality can be degraded due to many false positive alarms (Fig. 5 (c)). We set $\tau_O$ to approximately 0.4–0.5 for a scene containing $20\times20$–$200\times200$ pixels of vehicles (Fig. 5 (b)).

### 2.2.2 Non-Maximum Suppression

In [4], Dalal developed an effective non-maximum suppression framework that optimizes responses of initial detections based on two requirements: True initial detections should
give high classification scores and be observed more densely in a frame. To obtain the final detection results, we employ the fundamental idea of this non-maximum suppression technique. First, we represent a pdf of initial detections through a multivariate kernel density estimator with Gaussian kernels [26] as

\[ f(d) = \frac{1}{|D_{\text{init}}| \cdot (2\pi)^{D/2}} \sum_{d_i \in D_{\text{init}}} \frac{s_{c_d_i}}{|H_i|^{D/2}} \exp \left( -\frac{D_m^2(d; d_i, H_i)}{2} \right) \]

(13)

where an initial detection \( d_i \) and the corresponding classification score \( s_{c_d_i} \) is a data point and its weight, respectively. \( H_i \) denotes a \( 4 \times 4 \) bandwidth matrix defined manually for each \( d_i \), and fixed to \( \text{diag}(20, 20, 20, 20) \) in this work. The function \( D_m() \) indicates the Mahalanobis distance in Eq. (11).

In the feature space, modes of \( f(d) \) are found where high \( s_{c_d_i} \) values are produced, and a higher number of data points \( d_i \in D_{\text{init}} \) are located. Therefore, we can regard determining final detections from initial detections as finding modes of \( f(d) \) from given data points. Let \( d_i' \) denote a state-vector for \( d_i \) at the \( t \)-th mean-shift iteration [4], [27]. Then, we can calculate modes of \( f(d) \) by iteratively computing the following equations for all data points \( d_i \in D_{\text{init}} \) until the mean-shift vector \( (\mathbf{d}_i')^{t+1} - \mathbf{d}_i' \) converges to \( \mathbf{0} \):

\[ \mathbf{d}_i^{t+1} = H(d_i') \cdot \left[ \sum_{d_i \in D_{\text{init}}} w_j(d_i') H_j^{-1} d_j \right] \]

(14)

where

\[ H_j^{-1}(d_i') = \sum_{d_j \in D_{\text{init}}} w_j(d_i') H_j^{-1} \]

(15)

\[ w_j(d_i') = \frac{s_{c_d_i} |H_j|^{-1/2} \exp \left( -\frac{D_m^2(d_i'; d_j, H_j)/2}{2} \right)}{\sum_{d_i \in D_{\text{init}}} s_{c_d_i} |H_j|^{-1/2} \exp \left( -\frac{D_m^2(d_i'; d_j, H_j)/2}{2} \right)} \]

(16)

We show several examples of the found modes, i.e., final vehicle detection results for diverse initial detections, in Fig. 5.

3. Experimental Evaluation

We conducted experiments for five real highway traffic scenes involving unconstrained types, sizes and locations of vehicles (Fig. 8). For each scene, 12,000 training and 4,000 test image sequences were captured at a 640x480 pixel resolution from a fixed CCTV camera, and 200 ground truth detections were generated randomly in the test images. We implemented a simulator using the Visual Studio 2010 compiler supporting the parallel pattern library, and ran it on a workstation with an Intel Xeon CPU E5-2670.

For comparison, we additionally implemented two vehicle detection strategies employing the classical (i.e., CS W) [4], [22] and the newest adaptive (i.e., AS W) [2] sliding window techniques. We trained the appearance models for each detection method using the same dataset. More specifically, for each scene, we built the LSCs of the proposed detector and a HOG-SVM [4] classifier of CSW and ASW based on the 3,000–6000 positive and 10,000–12,000 negative samples that were semi-automatically collected from training sequences [5]. For the classifiers of CSW and ASW, we chose sample normalization sizes for each scene after examining a 32x32–96x96 range of candidate regions (Sect. 2).

Figure 6 shows the detailed results of LSC learning. In our work, since LSCs are produced for size models created depending only on vehicle types observable in local areas, ‘LSC#’ indicates the diversity of the vehicles appearing in a given scene. In this experiment, the least and most types of vehicles were observed in Scene1 and Scene5, respectively. We also provide several examples of actually constructed LSCs to show that each LSC can jointly provide appearance and size information of vehicles, and can offer well-separated appearance models for vehicles with significantly different sizes. From the subfigures (a) and (b) in Fig. 6, we can confirm that while appearance data on small vehicles such as sedans and SUVs are incorporated into LSCs in (a), the data on large vehicles such as buses and trucks are incorporated into LSCs in (b).

Next, we verify the effectiveness of our proposed detection scheme compared with the existing techniques CSW and ASW. To quantitatively measure the detection accuracy, we calculate the harmonic mean of the precision and recall for each method as:

\[ \text{accuracy} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \]

(17)

The evaluation results in Fig. 7 show that ASW obtains slightly better accuracy than our scheme in a scene with...
Fig. 7  Quantitative comparison of vehicle detection performance.

Fig. 8  Qualitative vehicle detection performance. The violet and yellow boxes show the results of background subtraction and vehicle detection, respectively.

Table 1  Comparison of operation speed (in fps)

| Scene  | Scene1 | Scene2 | Scene3 | Scene4 | Scene5 | Ave. |
|--------|--------|--------|--------|--------|--------|------|
| CSW    | 1.8    | 14.6   | 6.7    | 8.1    | 16.7   | 9.6  |
| ASW    | 3.1    | 21.2   | 9.1    | 15.5   | 20.6   | 13.9 |
| Proposed | 16.7  | 40.9   | 29.7   | 97.4   | 50.0   | 46.9 |

4. Conclusion

For vehicle detection, selection of normalization sizes to regularize the scales and aspect ratios of samples is essential because it can significantly influence the detection accuracy. To appropriately address this, we present an effective detection technique whereby vehicles are found based on novel image classifiers called LSCs. In contrast to conventional image classifiers [3], [4], [16]–[19], LSCs are constructed using normalization sizes specifically determined for local regions depending on scene-contextual information rather than ASW somewhat improves the operation speed by using automatically trained particular size models [2], it still requires a considerable computational cost because it ignores the relationship between the appearance and size of vehicles. Our method achieves a 16% increased detection accuracy (Fig. 7) with operation speeds that are at least three times faster (Table 1) by considering both size and appearance patterns of vehicles using systematically trained LSCs (Sect. 2.2.1).
than human intervention. Also, because LSCs jointly provide not only appearance clues but also size and location clues for observable vehicles, they can help systems operate in a simper and more reliable manner. From the experiments on real traffic data, we verified that our LSC-based scheme achieves higher accuracy with lower computational costs compared with existing sliding window-based methods [2], [4], [22]. However, there are still some challenges to overcome in accomplishing more practical vehicle detection. For instance, diverse weather conditions must be addressed and the concept of the detector adaptation [28]–[30] can be applied to continuously update LSCs. In future work, we will focus on developing detection techniques for further improvements by incorporating these issues.

Acknowledgments

This work was supported by institute for the information & communications technology promotion (IITP) grant funded by the Korean government (MSIP) (B0101-16-0525, Development of global multi-target tracking and event prediction techniques based on real-time large-scale video analysis).

References

[1] S. Pankanti, L. Brown, J. Connell, A. Datta, Q. Fan, R.S. Feris, N. Haas, Y. Li, N. Ratha, and H. Trinh, “Practical computer vision: Example techniques and challenges,” IBM Journal of Research and Development, vol.55, no.5, pp.3:1–3:12, 2011.
[2] S. Noh, D. Shim, and M. Jeon, “Adaptive sliding-window strategy for vehicle detection in highway environments,” IEEE Trans. Intell. Trans. Systems, vol.17, no.2, pp.323–335, 2016.
[3] C. Papageorgiou and T. Poggio, “A trainable system for object detection,” Int. Journal of Comput. Vis., vol.38, no.1, pp.15–33, 2000.
[4] N. Dalal, Finding people in images and videos, PhD thesis, Institute National Polytechnique de Grenoble, 2006.
[5] R.S. Feris, B. Siddiquie, J. Pettersson, Y. Zhai, A. Datta, L.M. Brown, and S. Pankanti, “Large-scale vehicle detection, indexing, and search in urban surveillance videos,” IEEE Trans. Multimedia, vol.14, no.1, pp.28–42, 2012.
[6] P. Dollar, S. Belongie, and P. Perona, “The fastest pedestrian detector in the west,” Proc. BMVA, pp.68.1–68.11, 2010.
[7] PASCAL visual object classes challenges, 2007.
[8] M.B. Blaschko and C.H. Lampert, “Learning to localize objects with structured output regression,” Proc. ECCV, pp.2–15, 2008.
[9] C.H. Lampert, M.B. Blaschko, and T. Hofmann, “Efficient sub-window search: A branch and bound framework for object localization,” IEEE Trans. Pattern Anal. Mach. Intell., vol.31, no.12, pp.2129–2142, 2009.
[10] M. Pedersoli, A. Vedaldi, and J. Gonzalez, “A coarse-to-fine approach for fast deformable object detection,” Proc. CVPR, pp.1353–1360, 2011.
[11] G. Cheng, J. Han, L. Guo, and T. Liu, “Learning coarse-to-fine sparselets for efficient object detection and scene classification,” Proc. CVPR, pp.1173–1181, 2015.
[12] R. Mottaghi, Y. Xiang, and S. Savarese, “A coarse-to-fine model for 3D pose estimation and sub-category recognition,” Proc. CVPR, pp.418–426, 2015.
[13] Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee, “Improving object detection with deep convolutional networks via Bayesian optimization and structured prediction,” Proc. CVPR, pp.249–258, 2015.
[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol.38, no.1, pp.142–158, 2016.
[15] H. Su, M. Sun, L. Fei-Fei, and S. Savarese, “Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories,” Proc. ICCV, pp.213–220, 2009.
[16] J. Liebelt and C. Schmid, “Multi-view object class detection with a 3D geometric model,” Proc. CVPR, pp.1688–1695, 2010.
[17] D.M. Gavrila, “A Bayesian, exemplar-based approach to hierarchical shape matching,” IEEE Trans. Pattern Anal. Mach. Intell., vol.29, no.8, pp.1408–1421, 2007.
[18] P.F. Felzenszwalb, R.B. Girshick, and D. McAllester, “Cascade object detection with deformable part models,” Proc. CVPR, pp.2241–2248, 2010.
[19] A. Krizhevsky, I. Sutskever, and G.E. Hinton, “ImageNet classification with deep convolutional neural networks,” Proc. NIPS, pp.1–9, 2012.
[20] S. Noh and M. Jeon, “A new framework for background subtraction using multiple cues,” Proc. ACCV, vol.7726, pp.493–506, 2013.
[21] S. Theodoridis and K. Koutroumbas, “Sequential clustering algorithms,” Pattern Recognition, 4th ed., pp.633–643, Elsevier, 2008.
[22] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” Proc. CVPR, pp.511–518, 2001.
[23] K. Mikolajczyk, C. Schmid, and A. Zisserman, “Human detection based on a probabilistic assembly of robust part detectors,” Proc. ECCV, vol.3021, pp.69–82, 2004.
[24] S.C. Brubaker, J. Wu, J. Sun, M.D. Multin, and J.M. Rehg, “On the design of cascades of boosted ensembles for face detection,” Int. Journal of Comput. Vis., vol.77, no.1-3, pp.65–86, 2008.
[25] O. Chum and A. Zisserman, “An exemplar model for learning object classes,” Proc. CVPR, pp.1–8, 2007.
[26] M.P. Wand and M.C. Jones, Kernel Smoothing 1st ed., Chapman & Hall, 1995.
[27] Y. Cheng, “Mean shift, mode seeking, and clustering,” IEEE Trans. Pattern Anal. Mach. Intell., vol.17, no.8, pp.790–799, 1995.
[28] X. Wang, G. Hua, and T.X. Han, “Detection by detections: non-parametric detector adaptation for a video,” Proc. CVPR, pp.350–357, 2012.
[29] P. Sharma, C. Huang, and R. Nevatia, “Unsupervised incremental learning for improved object detection in a video,” Proc. CVPR, pp.3298–3305, 2012.
[30] P. Sharma and R. Nevatia, “Efficient detector adaptation for object detection in a video,” Proc. CVPR, pp.3254–3261, 2013.

SeungJong Noh received a B.E. degree in information and telecommunication engineering from the Korea Aerospace University, Goyang, Korea, in 2009 and a M.S degree in information and communication engineering from the Gwangju Institute of Science and Technology, Gwangju, Korea in 2011. He is currently working toward a Ph.D. degree for machine learning and pattern recognition at the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju, Korea.
Moongu Jeon received the B.S. degree in architectural engineering from the Korea University, Seoul, Korea, in 1988 and the M.S. and Ph.D. degrees in computer science and scientific computation from the University of Minnesota, Minneapolis, MN, USA, in 1999 and 2001, respectively. As a Postgraduate Researcher, he worked on optimal control problems at the University of California at Santa Barbara, Santa Barbara, CA, USA, in 2001–2003, and then moved to the National Research Council of Canada, where he worked on the sparse representation of high-dimensional data and the level set methods for image processing until July 2005. In 2005, he joined the Gwangju Institute of Science and Technology, Gwangju, Korea, where he is currently a full professor in the School of Electrical Engineering and Computer Science. His current research interests are in machine learning, computer vision, and ITS.