A Cascaded Ensemble of Sparse-and-Dense Dictionaries for Vehicle Detection

Zihao Rong *, Shaofan Wang, Dehui Kong and Baocai Yin

Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; rzh@emails.bjut.edu.cn (Z.R.); kdh@bjut.edu.cn (D.K.); ybc@bjut.edu.cn (B.Y.)
* Correspondence: wangshaofan@bjut.edu.cn

Featured Application: The vehicle detection algorithm proposed in this work could be used in autonomous driving systems to understand the environment, or could be applied in surveillance systems to extract useful transportation information through a camera.

Abstract: Vehicle detection as a special case of object detection has practical meaning but faces challenges, such as the difficulty of detecting vehicles of various orientations, the serious influence from occlusion, the clutter of background, etc. In addition, existing effective approaches, like deep-learning-based ones, demand a large amount of training time and data, which causes trouble for their application. In this work, we propose a dictionary-learning-based vehicle detection approach which explicitly addresses these problems. Specifically, an ensemble of sparse-and-dense dictionaries (ESDD) are learned through supervised low-rank decomposition; each pair of sparse-and-dense dictionaries (SDD) in the ensemble is trained to represent either a subcategory of vehicle (corresponding to a certain orientation range or occlusion level) or a subcategory of background (corresponding to a cluster of background patterns) and only gives good reconstructions to samples of the corresponding subcategory, making the ESDD capable of classifying vehicles from background even though they exhibit various appearances. We further organize ESDD into a two-level cascade (CESDD) to perform coarse-to-fine two-stage classification for better performance and computation reduction. The CESDD is then coupled with a downstream AdaBoost process to generate robust classifications. The proposed CESDD model is used as a window classifier in a sliding-window scan process over image pyramids to produce multi-scale detections, and an adapted mean-shift-like non-maximum suppression process is adopted to remove duplicate detections. Our CESDD vehicle detection approach is evaluated on KITTI dataset and compared with other strong counterparts; the experimental results exhibit the effectiveness of CESDD-based classification and detection, and the training of CESDD only demands small amount of time and data.

Keywords: object detection; vehicle detection; dictionary learning; ensemble learning

1. Introduction

Vehicle detection as a special case of object detection is a computer vision technique having practical meaning. For example, it plays an important role in autonomous systems based on camera input. However, vehicle detection faces challenges, like the problem of recognizing vehicles exhibiting various orientations, the serious influence from occlusions, the clutter of background content, to list a few.

Existing vehicle detection approaches, e.g., [1–6], could be divided into two categories: non-deep-learning approaches and deep-learning-based approaches. Generally, non-deep-learning detection approaches rely on a traditional machine learning classification model as window or proposal classifier with limited discriminative ability [1,7], or further aided by 3D geometrical models while requiring more detailed labeling [6]. Existing deep-learning-based detection approaches [8,9] exhibit strong discriminative ability, but their...
training-data-intensive and training-time-intensive characteristics cause troubles for their real-world application. In contrast, dictionary learning models have been successfully applied to recognition tasks through learning of dictionaries for discriminative coding, attaining good recognition ability with low demanding of training data and time [10,11]. In the case of vehicle detection, vehicles and background exhibit various appearances which are hard to represent with a single dictionary, but are well represented with an ensemble of multiple dictionaries [12,13]; furthermore, dictionaries could easily cooperate with additional machine learning models to attain strong recognition abilities, like [13,14].

This paper presents a dictionary-learning-based sliding-window-styled vehicle detection approach. In general, the proposed approach learns an ensemble of dictionaries to represent both vehicle appearances and background patterns. Specifically, vehicle and background categories are further divided into several subcategories, each of which is easier to represent with a dictionary model; each subcategory of vehicle corresponds to certain orientation range (a sub-range in $0-2\pi$) and certain occlusion level (fully visible/partly occluded/largely occluded); and each subcategory of background corresponds to a cluster of background patterns (the background pattern clusters are obtained by clustering algorithms like K-means). For each subcategory, we learn a pair of sparse-and-dense dictionaries (SDDs) to represent corresponding sample appearances; SDD is capable of capturing subcategory-specific intrinsic features with its sparse dictionary and absorbing non-subcategory-specific patterns (like noises and shadows) with its dense dictionary, so it is able to give good reconstructions to samples of the corresponding subcategory only but not for others; the desired SDD could be efficiently learned through the supervised low-rank (SLR) decomposition [11]. Thus, an ensemble of such SDDs (ESDD) for all subcategories of vehicle and background are capable of well representing both of them even though their appearances are various due to the aforementioned challenges, and each constituting SDD’s subcategory-specific reconstruction ability enables the ESDD to classify vehicle from background. We further organize the ESDD as a two-level cascade of ESDD (CESDD) to perform two-stage coarse-to-fine classifications which is more robust and reduces computations; input samples are firstly classified by the first level of ESDD corresponding to easy-to-distinguish subcategories and then, based on the first-level classification results, selectively passed to SDDs within the second level (corresponding to hard-to-distinguish subcategories) for re-classification. To attain further accuracy enhancement, the CESDD is further coupled with a downstream AdaBoost process whose weak classifiers take corresponding SDDs’ reconstruction residues as input features and are aggregated to produce final classifications. The CESDD model is used as a window classifier in a simple-to-implement sliding-window scanning process over input image pyramids to generate multi-scale detections. These detections usually contain duplicate ones, so we adopt non-maximum suppression (NMS) process then; specifically, instead of the often-used greedy NMS [2] which brutally trade-off between accuracy and recall rate through inter-detection IoU thresholding, we adopt the mean-shift-like NMS [7] which could effectively remove duplicate detections while properly handling the detections of very near vehicles, and some adaptations are introduced for bounding box refinement and false positive reduction. To summarize, the main contributions of this work are:

- an SDD-ensemble-based vehicle/background classification approach (ESDD);
- an effective organization of SDD ensemble into a two-level cascade for sliding-window classification robustness and computation saving (CESDD);
- a cooperation mechanism of CESDD with a downstream AdaBoost process for accurate sliding-window classification;
- an adapted mean-shift-like NMS method for duplicate detection removal.

The remainder of the paper is organized as follows: Section 2 reviews related works; Section 3 introduces the training of CESDD; Section 4 introduces vehicle detection with CESDD; Section 5 presents experiments on CESDD; Section 6 gives a conclusion of this work and further improvement directions of CESDD.
2. Related Works

In this section, we review previous works on vehicle detection and object detection of different categories of methodologies, and works on application approaches of detection on autonomous systems.

2.1. Non-Deep-Learning Object Detection

As for the domain of non-deep-learning sliding-window-styled object detection, there are several effective schemes with various complexity. The renowned Viola-Jones detector [15] achieved good human face detection performance, employing a set of simple features describing local image intensity patterns and an AdaBoost classifier. Dalal and Triggs [1] designed the effective HOG feature to be used with a linear-SVM classifier, achieving big improvement in pedestrian detection. To adapt to deformable objects, Felzenszwalb et al. [2] proposed a root window filter in connection with several part filters whose positions are adjustable, in combination with a Latent-SVM classifier; this proposal gained considerable enhancement in multi-class object detection. Girshick et al. [3] further generalized this part-based model to grammar models which allow more flexible subdivision of parts and embodied ability to represent partial occlusions. Based on the idea of organizing multiple window filters with graph structure as in the case of grammar models, Wu et al. [16] proposed the And-Or Graph model with more complex graph structure to represent variations of vehicle appearances, and even the appearances of multiple vehicles altogether, being able to handle more various occlusions.

Some other non-deep-learning object detection methods employ 3D geometry models to assist detection. Xiang et al. [17] adopted 3D voxel models to represent several frequently appearing vehicle appearances, being able to perform accurate 2D detections as well as 3D detections. Zia et al. [6] established fine-grained 3D wireframe models for 3D vehicle detection, and thus enabled inference of occlusions to the grain level of wireframe vertex.

2.2. Deep-Learning-Based Object Detection

Deep-learning-based object detection methods use deep neural networks to undertake all or a great portion of the whole process of detection. Girshick et al. [4] proposed the region-based convolutional neural network model (R-CNN) for object detection task, which adopted selective-search-based region proposal generation, CNN-based feature extraction and SVM-based classification in sequence. This R-CNN model was further modified into Fast R-CNN [5] by absorbing classification and bounding box regression stages into one CNN, achieving better detection performance. Redmon et al. [18] proposed the YOLO object detection model which executes the complete detection task all in one deep neural network, regressing from the input image all the way to bounding boxes in real-time. Liu et al. [8] proposed the single-shot multibox detector (SSD) model which uses a deep CNN to explore simultaneously at many positions on an input image plane with multiple types of anchor boxes of various sizes and aspect ratios, and adjusts these anchor boxes to fit on target objects. Murtza et al. [19] noticed the effectiveness of both manually designed features and neural-network-learned features for object detection, and thus proposed concatenating HOG pyramid with CNN features and learning an ensemble classifier upon the concatenated features; this method was proved to be effective on pedestrian detection. Zhang et al. [20] used a pre-trained CNN model for image classification and performed transaction mining to obtain frequently occurring activation patterns as cues for target object detection in previously unseen images. Rahman et al. [21] performed object detection on RGB-D images and proposed a region proposal generation network that is able to fuse multi-modal information. Wang et al. [22] noticed the inconvenience of manually setting hyperparameters for region proposal network (RPN) which is an essential component in many two-stage object detection methods, and thus proposed to utilize collective intelligence algorithms, like particle swarm optimization, to optimize these hyperparameters, enhancing the accuracy of RPN-based methods. Xu et al. [23] noticed the inaccuracy of bounding box regression in R-CNN series methods, and proposed a kind of
particle-searching-based algorithm instead; the particles search for local features pertinent to the target object in the feature map generated by the network, and were then clustered and grouped to induce the possible bounding boxes.

2.3. Application Approaches of Detection on Autonomous Systems

As for the special case of application on autonomous systems, various detection approaches have been proposed with special designs to ensure the efficiency and reliability of inference. Fremont et al. [24] proposed to use fish-eye camera input to perform human detection around heavy machines like forklifts; to deal with the distortion in fish-eye images, they use deformable part-based models (DPMs) [2] with different part anchor settings for different places in the camera view; they further introduced LiDAR input on which initial regions of interest are generated and perform finer detections within these regions, resulting in enhanced time performance and reliability. Tsiktsiris et al. [25] proposed a hybrid model of convolutional neural networks and recurrent neural networks to perform abnormal event detection in autonomous shuttle mobility infrastructures; they adopted a special auto-encoder which is comprised of both convolution layers and ConvLSTM [26] layers to extract spatiotemporal features for video sequences within shuttle infrastructures, and fed the video features to subsequent stacked LSTMs [27] to produce “normal/abnormal” classification results. Xu et al. [28] proposed to deal with the car rotation problem in detection on unmanned aerial vehicles’ imaging by adjusting the images according to road direction which is estimated through straight line detection; they also proposed a switching algorithm to choose between two candidate detectors based on their time performances under different scenes, in order to maintain detection rate.

Table 1 summarizes the strengths and weaknesses of detection approaches of the aforementioned methodologies and our CESDD approach. Compared with these approaches, our CESDD could be trained with a small amount of training samples and training time while still achieving good classification and detection performance.

| Methodology      | Strengths                                      | Weaknesses                                      |
|------------------|-----------------------------------------------|-------------------------------------------------|
| deep-learning-based | powerful feature learning and end-to-end training | training data intensive; training time intensive |
| non-deep-learning | require less training time and training data    | limited learning ability                         |
| our CESDD        | require less training time and training data; relatively good learning ability | computation intensive at inference               |

3. Training of CESDD

The essence of CESDD training is to learn the ensemble of descriptive sparse-and-dense dictionaries, upon which downstream AdaBoost classifier is learned for reinforced vehicle/background classification, from a set of training samples which should be properly processed into features fit for detection. A workflow of training of CESDD is illustrated in Figure 1a, and a complete algorithm description is summarized in Algorithm 1.
Algorithm 1: Algorithm for training CESDD

input : Training samples $VEH \cup BKG$ (vehicle and background image samples) of unified size, vehicle subcategory number $N_{VEH}$ and background subcategory number $N_{BKG}$

output : Trained vehicle detector (learned SDD ensemble and learned downstream Adaboost classifier)

// Sample grouping
1 Group all vehicle samples into $N_{VEH}$ subcategories as $VEH = \bigcup_{i=1}^{N_{VEH}} VEH_i$ according to their orientations and occlusion levels;
2 Group all background samples into $N_{BKG}$ subcategories as $BKG = \bigcup_{i=1}^{N_{BKG}} BKG_i$ using K-means clustering;

// Feature extraction
3 Extract concatenated F-HOG, CS-LBP, Color Names features for each training sample;

// SDD ensemble learning
4 Obtain subcategory-specific dictionary pairs $\{(A_{veh}^{i}, B_{veh}^{i})\}_{i=1}^{N_{VEH}}$, $\{(A_{bkg}^{i}, B_{bkg}^{i})\}_{i=1}^{N_{BKG}}$ by solving (2) using Algorithm A1;
5 Code over SDD ensemble for all training samples to obtain reconstruction residues from all SDDs as $\{R_{veh}^{i}\}_{i=1}^{N_{VEH}}$ and $\{R_{bkg}^{i}\}_{i=1}^{N_{BKG}}$;

// Learning downstream Adaboost classifier (ensemble of weak classifiers $WC_{C1}$, $WC_{C2}$, $WC_{C3}$)
6 Organize SDD ensemble into two levels: Level 1 (vehicle: $FV$ and $PO$ / background: $GB$), and Level 2 (vehicle: $LO$ / background: $VP$);
7 Combine reconstruction residues produced by SDDs of $FV$ and $PO$ vehicle with the ones from $GB$ background into $N_{FV \& PO} \times N_{GB}$ combinations as weak classifiers’ training samples, and perform AdaBoost learning to learn corresponding $N_{FV \& PO} \times N_{GB}$ weak classifiers as $WC_{C1}$;
8 For training samples predicted by Level 1 as positives, combine reconstruction residues produced by SDDs of $FV$ and $PO$ vehicle with the ones from $VP$ background into $N_{FV \& PO} \times N_{VP}$ combinations as weak classifiers’ training samples, and perform AdaBoost learning to learn corresponding $N_{FV \& PO} \times N_{VP}$ weak classifiers as $WC_{C2}$ for Level 2;
9 For training samples predicted by Level 1 as negatives, combine reconstruction residues produced by SDDs of $GB$ background with the ones from $LO$ vehicle into $N_{GB} \times N_{LO}$ combinations as weak classifiers’ training samples, and perform AdaBoost learning to learn corresponding $N_{GB} \times N_{LO}$ weak classifiers as $WC_{C3}$ for Level 2;

Specifically, CESDD training learns a two-level cascade of dictionary ensembles. Level 1 consists of dictionary pairs of subcategories from relatively standard vehicle and background types with clear specificity: fully visible (FV) vehicles and partly occluded (PO) vehicles for vehicle category, and general background (GB) for background category. Level 1 ensemble is responsible for producing initial predictions which are then handed over to Level 2 of the cascade for refinement. Level 2 consists of dictionaries of relatively vague types: largely occluded (LO) vehicles for vehicle category, and vehicle parts (VP) for background category. The usage of Level 2 ensemble differs from Level 1: each of the samples tagged as positive in initial predictions is to be represented over dictionary pairs of type VP, producing corresponding reconstruction residues; the minimal one of these residues are to be compared with the minimal one from vehicle category in Level 1; if the minimal residue from VP is smaller, then the initial prediction should be corrected as negative. This refinement process is identically carried out over samples initially predicted as negative: each of these samples is to be represented over dictionary pairs of LO vehicle subcategories; the minimal reconstruction residue from these subcategories is to be compared with the minimal one from GB subcategories in Level 1; if the minimal residue from LO vehicle is smaller, then the sample should be regarded as positive. This cascade-styled classification scheme exhibited its advantage in reducing false positives during detection, as shown in Figure 2; its classification performance is illustrated in subsequent experiments.
Through levels in the cascade, ESDD-based window classification is performed: a window patch from sliding-window scan process is passed to each SDD in the ensemble for coding, and the resulting reconstruction residues are then transformed into the input to the downstream AdaBoost classifier to give prediction.

![Flowcharts of training CESDD](image)

(a) Training of CESDD. All sparse-and-dense dictionaries (SDDs) of both levels in the cascade are obtained through ESDD learning: training samples are firstly grouped into subcategories; then the SDD ensemble is learned with one pair of dictionaries (A, B) for each subcategory; next, downstream AdaBoost classifier is learned from the residue combinations from the ESDD; (b) Detection with CESDD. Through levels in the cascade, ESDD-based window classification is performed: a window patch from sliding-window scan process is passed to each SDD in the ensemble for coding, and the resulting reconstruction residues are then transformed into the input to the downstream AdaBoost classifier to give prediction.

![Initial detections](image)

(a) Initial detections without cascade; (b) Initial detections with cascade.

### 3.1. Training Sample Categorization

In this work, samples from vehicle category and background category consist of several deliberately specified types, to take care of the variousness of image contents during vehicle detection. Samples in vehicle category consist of three types with different occlusion levels: FV vehicles, PO vehicles, and LO vehicles; samples in background category consist of two types: GB and VP. The constitution is illustrated in Figure 3. These types constitute almost all sorts of image contents in urban street views in KITTI2017 dataset [29] (the 2D object detection subset included in KITTI’s “3D Object Detection Evaluation 2017” channel).
Moreover, to be suitable for subsequent ESDD learning, each type of vehicle samples are further divided into subcategories according to their orientations, as illustrated in Figure 4. Background types are further divided into subcategories through clustering process for which we adopted K-means.

![Training samples consist of vehicles (a) and backgrounds (b). Vehicle category is divided into three types according to extent of occlusion; background category is divided into two types according to whether vehicle part appears.](image)

**Figure 3.** Training samples consist of vehicles (a) and backgrounds (b). Vehicle category is divided into three types according to extent of occlusion; background category is divided into two types according to whether vehicle part appears.

![Vehicle samples of various orientations, thus various aspect ratios. Each group corresponds to a unique orientation range; (a): 0 \sim \frac{\pi}{4}; (b): \frac{\pi}{4} \sim \frac{\pi}{2}; (c): \frac{\pi}{2} \sim \frac{3\pi}{4}; (d): \frac{3\pi}{4} \sim \pi; (e): \frac{3\pi}{4} \sim 2\pi; (f): \frac{3\pi}{2} \sim \frac{5\pi}{4}; (g): \frac{5\pi}{4} \sim \frac{3\pi}{2}; (h): \pi \sim \frac{5\pi}{4}.](image)

**Figure 4.** Vehicle samples of various orientations, thus various aspect ratios. Each group corresponds to a unique orientation range; (a): 0 \sim \frac{\pi}{4}; (b): \frac{\pi}{4} \sim \frac{\pi}{2}; (c): \frac{\pi}{2} \sim \frac{3\pi}{4}; (d): \frac{3\pi}{4} \sim \pi; (e): \frac{3\pi}{4} \sim 2\pi; (f): \frac{3\pi}{2} \sim \frac{5\pi}{4}; (g): \frac{5\pi}{4} \sim \frac{3\pi}{2}; (h): \pi \sim \frac{5\pi}{4}.

### 3.2. Feature Extraction

For in-the-wild vehicle detection, vehicles exhibit various appearances, resulting from geometrical differences, various orientations, tremendous illumination variations, etc. Adoption of descriptive features robust against these influences is critical to effective classification and vehicle detection. Two essential properties of sample appearance should be emphasized here: shape and color. Shape information generally resides in two scales: contours and other geometrical patterns on a macro scale, and textures on a micro scale. In this work, several types of feature extraction schemes are tested and compared regarding their performances in classification and detection, as could be seen in subsequent experiments. Based on the testing results, it is discovered that the combination of features describing all three types of appearance properties (macro shape, micro texture, and color) gives the best classification performance; specifically, we adopted FHOG for description of macro shapes, cell-structured LBP (CSLBP) for micro textures, and color names (CN) for color information.

The concatenated features make up high dimensionality, which is time-consuming when processed and may contain redundant components that are useless in classification. To deal with this issue, \ell_1-based feature selection with the assistance of a linear SVM is utilized and works well with the proposed features, as could be seen from subsequent experiments.

### 3.3. SDD Ensemble Learning

The effectiveness of sliding-window-styled detection relies on the effectiveness of window-wise classification. Through the scanning process, a great number of window patches are densely sampled from many positions in images of multiple scale levels,
consisting usually of far more background patches than well-bounded vehicle patches. Thus, a window-wise classifier is obliged to achieve high enough classification accuracy as well as good recall rate. Classifier’s accuracy and recall rate are factually two views of one essence: modeling quality of target data space.

In this work, the proposed novel classifier, using dictionary ensemble, exploits this very essence directly. Specifically, training samples are categorized as vehicle and background, and are further divided into subcategories. For vehicle category, dividing criterion is set as vehicle orientation and occlusion level, which influence vehicle appearances severely. For background category, sample contents are not related to useful labels (that is, orientations and occlusion levels) as in the case of vehicle category, so we adopt K-means clustering algorithm to directly produce background training sample clusters as subcategories, each containing samples with similarity. With the constitution of subcategories set, dictionaries could be learned. For each subcategory, with limited variance of appearances, dictionary with adequate representation ability could be learned effectively. Specifically, the proposed SDD dictionary learning method is adopted and described in the following subsection.

3.3.1. Basic SDD Learning and Coding

For the task of learning SDD for a single subcategory, the SLR-decomposition-based method [11] is proposed. Given a training sample collection \( D \in \mathbb{R}^{k \times m} \) from a certain subcategory, with \( m \) samples of \( k \)-dimension arranged as column vectors, we decompose it into three parts as:

\[
D = AX + BY + R
\]  

(1)

where the first term \( AX \) preserves subcategory-specific intrinsic patterns existing throughout all samples (\( A \in \mathbb{R}^{k \times n_A}, X \in \mathbb{R}^{n_A \times m} \) denote the intrinsic sparse dictionary and the corresponding sparse coding, respectively), the second term \( BY \) preserves non-intrinsic patterns (\( B \in \mathbb{R}^{k \times n_B}, Y \in \mathbb{R}^{n_B \times m} \) denote the non-intrinsic dense dictionary and the corresponding dense coding, respectively), the third term \( R \in \mathbb{R}^{k \times m} \) represents residues like noises, and \( n_A, n_B \) denote the atom number of \( A, B \), respectively.

Taking into consideration that samples within certain subcategory share similarities to some extent, low-rank property of \( D \) and \( (A, B) \) could be reasonably expected. In addition, with the assumption that \( A \) is responsible for capturing subcategory-intrinsic patterns while \( B \) for non-subcategory-intrinsic patterns, for a sample of the same subcategory, its relatively stable subcategory-intrinsic pattern is expected to be captured by one or a few atoms in \( A \) with each atom being a distinct mode, while the relatively unstable non-subcategory-intrinsic part should be allowed more freedom to choose linear combination coefficients over atoms in \( B \) to better deal with the variances. Thus, the coding \( X \) over \( A \) is expected to be sparse, while the coding \( Y \) over \( B \) is expected to be relatively dense. With all these considerations, the desired objective for the proposed SDD learning could be expressed as:

\[
\min_{A,X,B,Y,R} \|A\|_* + \omega\|X\|_1 + \lambda\|B\|_* + \tau\|Y\|_F^2 + \eta\|R\|_1
\]

s.t. \( D = AX + BY + R \) 

(2)

The low-rank property of \( (A, B) \) could be measured with nuclear norm; the sparsity of codings \( X \) could be measured with \( \ell_1 \) norm; since there is no requirement of sparsity for codings \( Y \), only its magnitude is to be minimized, measured with Frobenius norm; the residues \( R \), possibly noises, should exhibit sparsity and is thus measured with \( \ell_1 \) norm. At the same time, the reconstruction quality is required to be maintained, acting as a restriction term.
Once the dictionary pair \((A, B)\) is learned through optimizing the SDD learning objective proposed above, they could be used to represent incoming samples. Given a single input sample \(d \in \mathbb{R}^k\), its representation with \((A, B)\) could be expressed as:

\[
d = Ax + By + r \tag{3}
\]

where \(x \in \mathbb{R}^n\) is the coding over \(A\), \(y \in \mathbb{R}^n\) is the coding over \(B\), and \(r \in \mathbb{R}^k\) is the residue. The expected properties of the coding result should follow the analysis of the SDD learning objective above. Thus, the objective of coding over \((A, B)\) is directly inferred as below:

\[
\min_{x,y,r} \|x\|_1 + \gamma \|y\|_2^2 + \beta \|r\|_1 \quad \text{s.t. } d = Ax + By + r \tag{4}
\]

The solution to SDD learning objective \((2)\) is left until Appendix A, while the solution to \((4)\) is similar to \((2)\) and hence we skip the details.

### 3.3.2. Constructing SDD Ensemble

With basic SDD learning and coding approaches set up, the strategy of using SDDs to perform vehicle/background classification would be straightforward: we model the data space by representing vehicle and background categories with respective SDDs, and an input sample of certain category should receive distinct reconstruction qualities (in terms of reconstruction residues) from SDDs of different categories. Before coming to the proposed multi-subcategory scheme, a more heuristic scheme is to learn only one SDD for vehicle category and one for background, as illustrated in Figure 5a. However, as simple as it is, this no-subcategory scheme gave unsatisfactory classification performance, as shown in Figure 6a. This behavior could be expected, since both vehicles and background in the real-world dataset (such as KITTI) exhibit various appearances which are hard to represent with only one SDD per category; the great variances of appearances caused trouble for SDD learning to capture patterns intrinsic enough for a whole category. This observation led us to propose the multi-subcategory scheme, which subdivides both vehicle and background categories into multiple subcategories, as described earlier and illustrated in Figure 5b. By introducing more subcategories, the variances of sample appearances within subcategories are within the representation ability of SDDs, and the patterns intrinsic to each subcategory are easier to capture, so the discriminative ability enhancement is expected and proved by experimental results in Figure 6a.

### 3.4. Downstream AdaBoost Classifier Learning

Even though the proposed cascade of SDD ensemble achieved pretty good classification performance, the situation of in-the-wild vehicle detection is still very complex in terms of image appearances. In this case, classification accuracy, especially for vehicle category, deserves more emphasis, since it is critical to the robustness of vehicle detection. The fact of the existence of ensemble of SDDs in the proposed CESDD-based classifier provides a chance to cooperate with ensemble learning models, like AdaBoost. Specifically, every SDD from vehicle category and every other SDD from background category could be coupled to act as a weak vehicle/background classifier (a micro ESDD-based classifier with ensemble size as two). In this way, many such couples of SDDs could be constructed as such kind of weak classifiers, creating the possibility of applying AdaBoost process. These SDD couples are not independent from each other, since there exists couples sharing common SDDs, while AdaBoost algorithm requires independent weak classifiers that are learned incrementally. To bridge this gap, instead of directly using the SDD couples as weak classifiers, additional machine learning models could be adopted as the weak classifiers which take the reconstruction residues from the SDD couples as inputs. With this adaptation, AdaBoost process could be carried out to learn these weak classifiers and corresponding aggregation weights. The steps in CESDD training and CESDD-based detection where this
The downstream AdaBoost mechanism is involved are illustrated as portions of Figure 1a,b; note that each weak classifier is learned from the training samples’ reconstruction residues produced by the corresponding SDD couple only.

In the case of cascaded SDD ensemble, weak classifiers learned from SDD couples purely consisting of Level 1 SDDs are evaluated for all incoming test samples to generate initial classifications, while weak classifiers learned from SDD couples consisting of Level 2 VP type SDDs and Level 1 vehicle category (FV or PO) SDDs are only evaluated for initial positive classifications, and weak classifiers learned from SDD couples consisting of Level 2 LO vehicle type SDDs and Level 1 background category SDDs are to be evaluated only for initial negative classifications. For an input sample, classifications from all evaluated weak classifiers are aggregated to form the final classification; the detailed procedure is illustrated in Figure 7. The effectiveness of this downstream AdaBoost process could be observed in subsequent experimental results, and the enhancement in classification accuracy of vehicle category is noticeable.

Figure 5. Two kinds of subcategory divisions in SDD ensemble and corresponding classification workflows: (a) no subcategory; (b) multiple subcategories in each category.
Figure 6. Quantitative ablation evaluation results of CESDD classifier on KITTI2017. (a) multiple subcategories vs. no subcategory; (b) with cascade vs. without cascade; (c) with AdaBoost vs. without AdaBoost; (d) comparison of features; (e) with vs. without feature selection; (f) CESDD classifier vs. other classifiers.
Figure 7. Illustration of downstream AdaBoost classification process in CESDD. An input window patch is initially represented by Level 1 SDDs and classified by corresponding weak classifiers to generate initial classification, and then, based on the initial classification result, selectively represented by SDDs in Level 2 and accordingly selectively classified by the rest weak classifiers; at last, classifications of activated weak classifiers are aggregated by learned weights to produce final classification.

4. Detection with CESDD

This section describes the process of detection with CESDD, which consists of three stages: sliding-window scan, window-wise classification, and non-maximum suppression. The workflow of the proposed CESDD-based vehicle detection is illustrated in Figure 1b, and a complete algorithm description of vehicle detection with CESDD is summarized in Algorithm 2.

4.1. Sliding-Window Classification

On receiving an input image, an image pyramid is built by up-sampling and down-sampling the image to multiple scale levels. Sliding-window scan process is performed on this pyramid by moving windows of proper aspect ratios (window width over window height) throughout all positions on each image plane in the pyramid. In the case of vehicle detection in driver view on streets, like the case of the KITTI2017 dataset, vehicle bounding boxes’ aspect ratios vary in a limited range depending on the vehicle orientations, from the large aspect ratio of side view to a small aspect ratio of front view, as is shown in Figure 4. This indicates that one single aspect ratio for scan window is not enough to capture vehicles of various orientations. Thus, we adopted two distinct aspect ratios for scan windows: one close to the ratios of side views, and another one close to front views; these two aspect ratio values were censused from the KITTI2017 training set label information. Window patches obtained using these aspect ratios are then resized to a uniform size, for the convenience of subsequent classification; these window patches are further processed into proper features as described in Section 3.2. CESDD-based classification is then carried out for all processed window patches; comprehensive description on classification with CESDD is detailed in Section 3.3, and the cooperation of CESDD with downstream AdaBoost process is described in Lines 5~13 in Algorithm 2. After window classification, initial detections are obtained.
Algorithm 2: Algorithm of vehicle detection with CESDD

```plaintext
Algorithm of vehicle detection with CESDD

input : Learned CESDD-based window classifier; input image img
output : Detections BBoxes (bounding boxes with corresponding confidence scores)
initialize : BBoxes = []
// Pre-processing
1 Build image pyramid Pyr by resizing img into multiple scales;
// Sliding-window scan and classification
for each level L in Pyr do
  for each window position p in image plane of level L do
    Extract features for current window (anchored at p) as d;
    Coding for d over SDDs in Level 1 ensemble and evaluate corresponding reconstruction residues \{r_1\};
    Perform Level 1 AdaBoost classification using WCs_1 with corresponding SDD residues as inputs and generate initial classification cls_init;
    if d is initially classified as positive according to cls_init then
      Coding over Level 2 background category SDDs to obtain corresponding reconstruction residues \{r_2\};
      Perform Level 2 AdaBoost classification using WCs_2 with corresponding SDD residues as inputs and aggregate with classifications of WCs_1 to generate cls_final;
    else
      Coding over Level 2 vehicle category SDDs to obtain corresponding reconstruction residues \{r_3\};
      Perform Level 2 AdaBoost classification using WCs_3 with corresponding SDD residues as inputs and aggregate with classifications of WCs_1 to generate cls_final;
    end
    if d is classified as vehicle according to cls_final then
      Store current detection information (bounding box and corresponding confidence score) into BBoxes;
    end
  end
end
// Post processing
19 Perform mean-shift-based NMS to refine detections BBoxes;
```

4.2. Mean-Shift-Based NMS

Directly applying the proposed CESDD-based classifier to all window patches would likely produce highly overlapped detections around target vehicles, as shown in Figure 6a. To deal with this issue, the mean-shift-based NMS is adapted and applied.

The detections from the previous classification are assigned bounding box information and scores of confidence. Mean-shift-based NMS initially transforms the detections into points in a x-y-scale 3D space, with x and y representing coordinates in the image plane and scale representing the scale level in the pyramid. Then, a kernel density estimation map is built by weighted combination of 3D Gaussian distributions centered at these 3D detection points, with the weights as monotonically increasing functions of the detection confidence scores. Then, from the initial positions of the detections in the 3D space, all detection points are moved in the map according to gradients; this moving process will be iterated several times until convergence where all 3D points have reached modes in the map. Finally, the reached modes will be regarded as desired detections. The detailed process could be found in [7]; this process is identical to a mean-shift process.

Some adaptations are necessary to make the mean-shift-based NMS to fit into the case of vehicle detection in this work. Firstly, the initial detections are not of the same aspect ratio. Thus, for each found mode from the mean-shift-based NMS, the aspect ratios of the group of detection points having reached the same mode are averaged; the average aspect ratio is applied to the output detection bounding box inferred from the mode. Apart from this, modes having attracted very few detection points, like one or two, tend to be false positives and should be dropped, because true vehicle targets always attract more detections around them. This could be observed in Figure 2. Furthermore, a proper threshold should be set to screen out modes with too low kernel density estimation values which correspond to detection confidences.
5. Experiments

In this section, we present the experimental results of our CESDD vehicle detection approach. Specifically, we evaluate CESDD-based classification and CESDD-based detection from multiple aspects. For all evaluations, we set the sparse dictionary A’s atom number as 10, and the dense dictionary B’s atom number as 20, for all subcategories’ SDDs; in total, we set 16 subcategories for vehicle category, and 12 subcategories for background category.

5.1. CESDD Classification

Experiments on classification are conducted over image patch samples obtained from KITTI2017. Vehicle samples, including car type and van type, could be cropped out according to corresponding label information; each vehicle sample is cropped using a window with one of the two pre-set aspect ratios for detection. Background samples, except for VP, are obtained using both two aspect ratios, by randomly sampling in non-vehicle areas. To obtain VP, cropping windows are sampled around vehicle area, partially overlapping with the vehicles with proper extents. In total, we collected 21,556 training samples and 11,710 test samples for CESDD classification evaluation. We evaluate classification performance with four criteria: accuracy for vehicle category (veh acc), recall rate for vehicle category (veh recall), accuracy for background category (bkg acc), recall rate for background category (bkg recall). These criteria are calculated as:

\[
\text{veh acc} = \frac{N_{tp}}{N_{tp} + N_{fp}} \quad \text{veh recall} = \frac{N_{tp}}{N_{veh}}
\]
\[
\text{bkg acc} = \frac{N_{tn}}{N_{tn} + N_{fn}} \quad \text{bkg recall} = \frac{N_{tn}}{N_{bkg}}
\]

where \(N_{tp}, N_{fp}, N_{tn}, N_{fn}\) denote the number of true positives, false positives, true negatives and false negatives in classifier predictions, and \(N_{veh}, N_{bkg}\) denote the number of vehicle and background samples in the test set. Here, we regard classification to vehicle category as “positive” and classification to background category as “negative”, and “true” indicates the classification is correct and “false” indicates incorrect.

Firstly, classification performances of two types of subcategory divisions are evaluated, as is presented in Figure 6a. Clearly, the scheme of an SDD ensemble constituted with multiple subcategories per category works better in classification, showing that the multi-subcategory scheme indeed gives better representation to the various vehicle appearances and background patterns.

The effect of organizing SDD ensemble into cascade is evaluated also. The classification performances with and without cascade are shown in Figure 6b. The test samples here include LO vehicles, thus is more difficult to discriminate. The application of cascade gives similar performance, but with decreased time cost.

Experiments have also been conducted to examine the effect of downstream AdaBoost. The classification performances with and without downstream AdaBoost are shown in Figure 6c. It could be seen that the downstream AdaBoost process improved classification accuracy of vehicle category greatly, which is very important to the robustness of vehicle detection.

The classification performances of several choices of features are evaluated, as shown in Figure 6d. Feature selection’s effect on classification performance is also evaluated, as shown in Figure 6e; it could be observed that the linear-SVM-based feature selection reduced dimensionality significantly while retaining almost the same classification performance as the original high-dimensional features.

The CESDD classifier is compared with other classifiers which are strong counterparts and based on different methodologies. Comparison results with random forest [30], gradient boosted trees (gbtrees) [31], discrete AdaBoost (boost) [32], linear-SVM, and RBF-SVM [33] are presented in Figure 6f. It could be seen that the CESDD classifier gives better performance than most counterparts and attains similar performance with the RBF-
SVM whose hyperparameters are heavily optimized during our experiment, using the same training and testing sample set. We also measured the training time of the counterpart approaches and our CESDD approach, as shown in Table 2. It could be observed that the training of CESDD is efficient, similar with the decision-tree-based counterparts (gradient boosted trees, discrete AdaBoost) and much faster than the SVMs.

Table 2. Training time comparison of CESDD-based classifier and other classifiers.

| Classifier            | Training Time (s) |
|-----------------------|-------------------|
| random forest         | 46.6              |
| gradient boosted trees| 232.1             |
| discrete AdaBoost     | 361.3             |
| linear-SVM            | 960.7             |
| RBF-SVM               | 10,731.3          |
| ours (CESDD)          | 470.0             |

5.2. CESDD Vehicle Detection

In this experiment, the CESDD vehicle detector is trained over 18,195 vehicle samples (including FV, PO, and LO), 33,218 background samples of GB, and 18,402 background samples of VP, all obtained from the KITTI2017 training set. For comparison, the DPM detector is obtained from voc-release5 site [34], which is named CAR_FINAL.MAT and trained from Pascal VOC 2007’s TRAINVAL set.

Qualitative test results from the proposed CESDD vehicle detector with and without mean-shift-based NMS are presented in Figure 8; qualitative comparison with DPM detector is presented in Figure 9. It is observed that the proposed CESDD vehicle detector is effective in these in-the-wild scenarios.

Figure 8. Illustration of the effect of mean-shift-based non-maximum suppression (NMS) on CESDD-based vehicle detection. Purple boxes are initial detections from window classification process; cyan boxes are refined detections through the adapted mean-shift-based NMS. (a,c,e) show initial detections; (b,d,f) show corresponding refined detections.
Figure 9. Qualitative vehicle detection results of our CESDD-based detector (a,c,e,g,i,k) vs. deformable part-based model (DPM) (b,d,f,h,j,l) on KITTI2017.
6. Conclusions

In this work, we have proposed a dictionary-learning-based vehicle detection approach named CESDD. It explicitly represents the various appearances of vehicles and background with an ensemble of SDD (ESDD) dictionaries for all subcategories, which could be efficiently learned using SLR decomposition, and organizes this ESDD as a two-level cascade for two-stage coarse-to-fine classification. Furthermore, CESDD is coupled with a downstream AdaBoost for robust classification. For vehicle detection, this CESDD model is used as a window classifier in sliding-window scan process over image pyramids to generate multi-scale detections, which are then refined using an adapted mean-shift-based NMS. Experimental evaluation results of CESDD-based classification and detection from multiple aspects proved that our CESDD is able to achieve good classification and detection performance with low demanding of training time and data. At present, our CESDD vehicle detection approach is time consuming at inference due to the time efficiency of coding over the dictionary ensemble, which could possibly be addressed by introducing parallelizations in further works.

Author Contributions: Conceptualization, Z.R. and S.W.; data curation, Z.R.; methodology, S.W.; project administration, D.K. and B.Y.; resources, D.K. and B.Y.; software, Z.R.; supervision, D.K.; validation, Z.R.; visualization, Z.R.; writing—original draft, Z.R.; writing—review and editing, S.W. and D.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not available.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Solution to SDD Learning Problem

We solve problem (2) using augmented Lagrange Multiplier method. Specifically, we introduce two auxiliary variables $A', B'$ as proxies for $A, B$, respectively, and transform (2) into the following optimization objective as an augmented Lagrangian function:

$$\min_{A,A',X,B,B',Y,R} \|A'\|_* + \omega \|X\|_1 + A\|B'\|_* + \tau \|Y\|_2^2 + \eta \|R\|_1 + \frac{1}{2} \left( \|D - AX - BY - R + \frac{1}{\mu}Z_1\|_F^2 + \|A - A' + \frac{1}{\mu}Z_2\|_F^2 + \|B - B' + \frac{1}{\mu}Z_3\|_F^2 \right)$$  \hspace{1cm} (A1)

where $Z_1, Z_2, Z_3$ are Lagrange multiplier matrices corresponding to the constraints $D = AX + BY + R, A = A', B = B'$, respectively. We present the complete algorithm for solving (A1) in Algorithm A1, where SVD(·) is the singular value decomposition operation, $K\text{means}(·)$ is the K-means clustering operation over input samples, taking the input-output form as:

$$[\text{centroids}, \text{labels}] = K\text{means}(D),$$ \hspace{1cm} (A2)

where centroids is a $k \times c$ matrix which stores $c$ clusters’ centroids, labels is a $c \times m$ matrix which stores the assigned cluster labels for all samples in $D; S_\lambda(·)$ is the elementwise soft-thresholding operation defined as:

$$S_\lambda(x) = \text{sign}(x) \max(|x| - \lambda, 0).$$ \hspace{1cm} (A3)
Algorithm A1: Algorithm for solving transformed SDD learning problem (A1)

1. **Input**: Training sample matrix \( D \), hyperparameters: \( \omega, \lambda, \tau, \eta, \rho = 1.5, \mu_{\text{max}} = 10^\beta, \epsilon = 10^{-4k} (k \) is feature dimension);
2. **Output**: \( A, A', X, B, B', Y, R; \)
3. **Initialize**: \( [A, X] = \text{Kmeans}([U, \Sigma, V]) \), where \([U, \Sigma, V] = \text{SVD}(D); A' = A; [B, Y] = \text{Kmeans}(D - AX), B' = B; Z_1, Z_2, Z_3 = 0; \mu = 10; \)
4. **While** max\([\|D - AX - BY - R\|_F, \|A - A'\|_F, \|B - B'\|_F] > \epsilon \) do 
5. // Solve for \( B', B \)
6. \[U, \Sigma, V] = \text{SVD}(B + \frac{1}{\rho}Z_3); \]
7. \( B' = US_1 \Sigma V^\top; \)
8. \( B = [(D - AX - R)Y^\top + B' + \frac{1}{\rho}(Z_1Y^\top - Z_2)](YY^\top + I)^{-1}; \)
9. \( R = S_2 \Sigma (D - AX - BY + \frac{1}{\rho}Z_1); \)
10. // Solve for \( Y \)
11. \( Y = (\frac{1}{\rho}I + B^\top B)^{-1}B^\top (D - AX - R + \frac{1}{\rho}Z_1); \)
12. // Solve for \( A', A \)
13. \([U, \Sigma, V] = \text{SVD}(A + \frac{1}{\rho}Z_3); \]
14. \( A' = US_1 \Sigma V^\top; \)
15. \( A = [(D - BY - R)X^\top + A' + \frac{1}{\rho}(Z_1X^\top - Z_3)](XX^\top + I)^{-1}; \)
16. \( R = S_2 \Sigma (D - AX - BY + \frac{1}{\rho}Z_1); \)
17. // Solve for \( X \)
18. \( X = \arg \min_X \frac{1}{2}\|AX - (D - BY - R + \frac{1}{\rho}Z_1)\|^2_2 + \frac{\omega}{2}\|X\|_1; \)
19. \( R = S_2 \Sigma (D - AX - BY + \frac{1}{\rho}Z_1); \)
20. // Update Lagrange multipliers 
21. \( Z_1 = Z_1 + \mu(D - AX - BY - R); \)
22. \( Z_2 = Z_2 + \mu(B - B'); \)
23. \( Z_3 = Z_3 + \mu(A - A'); \)
24. // Update the penalty coefficient \( \mu \)
25. \( \mu = \min(\rho \mu, \mu_{\text{max}}); \)
26. **End**

References

1. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), San Diego, CA, USA, 20–26 June 2005; pp. 886–893. [CrossRef]
2. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.A.; Ramanan, D. Object Detection with Discriminatively Trained Part-Based Models. IEEE Trans. Pattern Anal. Mach. Intell. 2010, 32, 1627–1645. [CrossRef]
3. Girshick, R.B.; Felzenszwalb, P.F.; McAllester, D.A. Object Detection with Grammar Models. In Proceedings of the Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems, Granada, Spain, 12–14 December 2011; pp. 442–450.
4. Girshick, R.B.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
5. Girshick, R.B. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
6. Zia, M.Z.; Stark, M.; Schindler, K. Towards Scene Understanding with Detailed 3D Object Representations. Int. J. Comput. Vis. 2015, 112, 188–203. [CrossRef]
7. Dalal, N. Finding People in Images and Videos. Ph.D. Thesis, Grenoble Institute of Technology, Grenoble, France, 2006.
8. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.E.; Fu, C.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision–ECCV 2016–14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37; [CrossRef]
9. Ren, S.; He, K.; Girshick, R.B.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Cortes, C.; Lawrence, N.D.; Lee, D.D.; Sugiyama, M.; Garnett, R., Eds.; pp. 91–99.
10. Mairal, J.; Bach, F.R.; Ponce, J.; Sapiro, G.; Zisserman, A. Supervised Dictionary Learning. In Proceedings of the Advances in Neural Information Processing Systems 21, the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–11 December 2008; pp. 1033–1040.
11. Jiang, X.; Lai, J. Sparse and Dense Hybrid Representation via Dictionary Decomposition for Face Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 2015, 37, 1067–1079. [CrossRef] [PubMed]
12. Zhu, Z.; Chen, Q.; Zhao, Y. Ensemble dictionary learning for saliency detection. *Image Vision Comput.* 2014, 32, 180–188. [CrossRef]
13. Huang, Y.; Quan, Y.; Liu, T. Supervised Sparse Coding With Decision Forest. *IEEE Signal Process. Lett.* 2019, 26, 327–331. [CrossRef]
14. Quan, Y.; Xu, Y.; Sun, Y.; Huang, Y.; Ji, H. Sparse Coding for Classification via Discrimination Ensemble. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 5839–5847. [CrossRef]
15. Viola, P.A.; Jones, M.J. Rapid Object Detection using a Boosted Cascade of Simple Features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), with CD-ROM, Kauai, HI, USA, 8–14 December 2001; pp. 511–518. [CrossRef]
16. Wu, T.; Li, B.; Zhu, S. Learning And-Or Model to Represent Context and Occlusion for Car Detection and Viewpoint Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* 2016, 38, 1829–1843. [CrossRef] [PubMed]
17. Xiang, Y.; Choi, W.; Lin, Y.; Savarese, S. Data-driven 3D Voxel Patterns for object category recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, 7–12 June 2015; pp. 1903–1911. [CrossRef]
18. Redmon, J.; Divvala, S.K.; Girshick, R.B.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
19. Murtza, I.; Khan, A.; Akhtar, N. Object detection using hybridization of static and dynamic feature spaces and its exploitation by ensemble classification. *Neural Comput. Appl.* 2019, 31, 347–361. [CrossRef]
20. Zhang, R.; Huang, Y.; Pu, M.; Guan, Q.; Zhang, J.; Zou, Q. Mining Objects: Fully Unsupervised Object Discovery and Localization From a Single Image. *arXiv* 2019, arXiv:1902.09968.
21. Rahman, M.M.; Tan, Y.; Xue, J.; Shao, L.; Lu, K. 3D object detection: Learning 3D bounding boxes from scaled down 2D bounding boxes in RGB-D images. *Inf. Sci.* 2019, 476, 147–185. [CrossRef]
22. Wang, G.; Guo, J.; Chen, Y.; Li, Y.; Xu, Q. A PSO and BFO-Based Learning Strategy Applied to Faster R-CNN for Object Detection in Autonomous Driving. *IEEE Access* 2019, 7, 18840–18859. [CrossRef]
23. Xu, G.; Su, X.; Liu, W.; Xiu, C. Target Detection Method Based on Improved Particle Search and Convolution Neural Network. *IEEE Access* 2019, 7, 25972–25979. [CrossRef]
24. Frémont, V.; Bui, M.; Boukerroui, D.; Letort, P. Vision-Based People Detection System for Heavy Machine Applications. *Sensors* 2016, 16, 128. [CrossRef] [PubMed]
25. Tsiktsiris, D.; Dimitriou, N.; Lalas, A.; Dasygenis, M.; Votis, K.; Tzovaras, D. Real-Time Abnormal Event Detection for Enhanced Security in Autonomous Shuttles Mobility Infrastructures. *Sensors* 2020, 20, 4943. [CrossRef] [PubMed]
26. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.; Wong, W.; Woo, W. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In Proceedings of the Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, Montreal, QC, Canada, 7–12 December 2015; Cortes, C.; Lawrence, N.D.; Lee, D.D.; Sugiyama, M.; Garnett, R., Eds.; pp. 802–810.
27. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* 1997, 9, 1735–1780. [CrossRef] [PubMed]
28. Xu, Y.; Yu, G.; Wang, Y.; Wu, X.; Ma, Y. A Hybrid Vehicle Detection Method Based on Viola-Jones and HOG + SVM from UAV Images. *Sensors* 2016, 16, 1325. [CrossRef] [PubMed]
29. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3356–3361.
30. Breiman, L. Random Forests. *Mach. Learn.* 2001, 45, 5–32. [CrossRef]
31. Friedman, J. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* 2000, 29. [CrossRef]
32. Hastie, T.; Rosset, S.; Zhu, J.; Zou, H. Multi-class AdaBoost. *Stat. Interface* 2009, 2, 349–360. [CrossRef]
33. Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* 1995, 20, 273–297. [CrossRef]