Category-consistent deep network learning for accurate vehicle logo recognition

Wanglong Lu[a], Hanli Zhao[a,*], Qi He[a], Hui Huang[a], Xiaogang Jin[b]

[a]College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou, 325035, China
[b]State Key Lab of CAD&CG, Zhejiang University, Hangzhou, 310058, China

Abstract

Vehicle logo recognition (VLR) is essential in intelligent transportation systems. Although many VLR algorithms have been proposed, efficient and accurate VLR remains challenging in machine vision. Many VLR algorithms explicitly detect the coarse region of the vehicle logo either by offsetting the detected location of the license plate or by training on numerous images with manual bounding-box annotations. However, the results of license plate detection can significantly influence the VLR accuracy, whereas bounding-box annotations are considerably labor-intensive. Thus, we propose a novel category-consistent deep network learning framework for accurate VLR. A convolutional-neural-network-based vehicle logo feature extraction model is proposed to extract deep features by considering both high- and low-level features in an image. Moreover, a novel category-consistent mask learning module is proposed to help the framework to focus on category-consistent regions without relying on license plate detection or manual box annotations. The deep network is trained and optimized iteratively with the objective function incorporating classification loss and category-consistency loss. Extensive experimental evaluations and comparisons on the publicly available HFUT, XMU, CompCars, and VLD-45 datasets demonstrate the feasibility and superiority of the proposed algorithm.

Keywords: Vehicle logo recognition, category consistency, convolutional neural network, deep learning

1. Introduction

Image recognition is an interesting topic in the field of computer vision. Owing to the increasing demand for automatic identification of vehicles, computer-aided intelligent technologies play an important role in improving the performance of various transportation systems [1, 2]. The detection of vehicle license plates and the recognition of vehicle manufacturers are crucial components in these systems [3, 4]. In the recognition process, the vehicle logo is certainly the clearest indicator of a vehicle's manufacturer [5], as vehicle logos with the same maker usually have a unique and standard visual design.

Vehicle logo recognition (VLR) has garnered considerable attention over recent decades, and many algorithms have been proposed to address this problem. By taking advantage of convolutional neural networks (CNNs), Huang et al. [6] proposed a CNN-based VLR system that uses an efficient pretraining strategy to reduce the high computational cost of kernel training. Yu et al. [7] presented a novel learning-based multilayer pyramid network for recognizing vehicle logo images. These methods require additional segmentation of vehicle logo regions based on some prior knowledge, such as license plate detection. However, if the location of the license plate is not accurately detected or if a vehicle has no license plate, the accuracy of subsequent recognition decreases significantly.

Certain CNN-based methods that support the recognition of vehicle logos from frontal images of vehicles have recently been studied. Yang et al. [8] recognized vehicle logos by using the YOLOv3 [9] framework. Yu et al. [10] presented a novel two-stage VLR framework with a region proposal network and convolutional capsule network. All these methods are based on the detection of vehicle logos and, thus, rely on a large quantity of manually annotated bounding boxes with image-level category labels. However, bounding-box annotations of vehicle logos are considerably labor-intensive.

In addition to vehicle logos, frontal faces of vehicles with the same vehicle maker share visual correlations. As shown in Fig. 1 (middle-left), the sample vehicle images of the same make (Acura) are captured under different conditions, such as vehicle model, vehicle color, and illumination. Because these vehicle images belong in the same Acura category, their frontal faces exhibit similar visual characteristics. Such an observation allows us to utilize these common characteristics to improve the performance of VLR. This motivates us to design a new VLR learning framework that automatically focuses on common category-consistent regions without explicitly knowing the locations of vehicle logos. Moreover, we are interested in developing a unified network learning framework for accurately recognizing vehicle logos from both vehicle logo images and frontal images of vehicles. Consequently, our algorithm can not only recognize frontal images of vehicles directly but also be used to improve the recognition performance for segmented vehicle logo patches.

To this end, we propose a novel category-consistent deep network learning framework for accurate VLR. Specifically, we propose a novel vehicle logo feature extraction CNN (called VLF-net) to extract hierarchical features automatically, and then, we use the extracted high- and low-level features to recog-

*Corresponding author. E-mail: hanlizhao@wzu.edu.cn

nize a vehicle logo given an input vehicle logo image. VLF-net reuses extracted intermediate features to reduce both computational cost and the number of parameters by taking advantage of the identity shortcut connection [11] and dense connection [12]. To further improve the recognition performance, a novel category-consistent mask learning (CCML) module is developed to learn the category-consistent regions without knowing accurate vehicle logo regions. Instead of using license plate detection techniques or manual bounding-box annotations of vehicle logos, our framework enforces a deep neural network to distinguish the common regions from uncorrelated backgrounds for each matching vehicle make in an image. Thus, our new algorithm can recognize vehicle logos from frontal images of vehicles without the explicit detection of vehicle logo locations. Our proposed algorithm achieves high accuracy on publicly available datasets of vehicle logo images (HFUT-VL [13] and XMU [6]) and vehicle images (CompCars [14] and VLD-45 [15]).

In summary, our study makes the following contributions:

- A novel feature extraction CNN called VLF-net is proposed to robustly extract multiple levels of vehicle logo features.
- A novel CCML module is proposed to help the network to focus on category-consistent regions without relying on license plate locations or manual box annotations.
- Classification loss and category-consistency loss are incorporated, and a new category-consistent deep network learning framework is proposed for accurate VLR.
- Various experimental results show that the proposed unified algorithm can achieve better performances in recognizing vehicle logos from both vehicle logo images and frontal images of vehicles.

The remainder of this paper is organized as follows. A brief review of the related work is provided in Section 2. Then, the details of our algorithm are introduced in Section 3. Section 4 describes the experimental setup. Experimental comparisons and discussions are presented in Section 5. Finally, the last section concludes the paper and provides a scope for future work.

2. Related work

Existing VLR methods can be roughly categorized into shallow learning-based and deep learning-based methods. Shallow learning-based methods usually do not require a large number of samples and efficiently extract visual features, such as edges, grayscale, and shapes, to describe vehicle logos. Deep learning-based methods can automatically learn multiple stages of invariant vehicle logo features by using CNNs.

In shallow learning-based methods, traditional handcrafted low-level features and shallow visual features have been widely used by leveraging a specific classification to solve the VLR problem. Pan et al. [16] first segmented the regions of interest that cover the logo based on the position of the license plate and then used AdaBoost and support vector machine (SVM) classifiers to localize and detect the logo. Peng et al. [17] proposed a novel feature representation strategy named statistical random sparse distribution to treat low-resolution and low-quality images and used multiscale scanning to locate and classify logos. Tafazzoli et al. [18] incorporated logo detection into their system to boost the reliability of vehicle make and model recognition. They trained their SVM classifier to recognize the regions provided by a sliding window technique, but their approach is computationally slow. Zhao et al. [19] applied the modified Hu invariant to extract vehicle logo features and then used the grey wolf optimization algorithm to optimize the SVM to identify logos. Yu et al. [13] employed patterns of oriented edge magnitudes to enhance the feature representation and employed collaborative representation-based classification to achieve satisfactory recognition results. Oriented texture patterns were also used for object matching in [20]. By using the histogram of oriented gradients (HOG) to describe the visual characteristics of an object, Lu et al. [21] presented a hierarchical multi-stage classification method to recognize vehicle models at the brand level for a given logo sub-region. Recently, Yu et al. [7] proposed a multilayer pyramid network based on learning. They mapped pixel difference matrices extracted from input images with different resolutions to binary matrices to obtain codebooks, and then they applied a multi-codebook-based classification method to solve the VLR problem. However, handcrafted features and shallow visual features are influenced by various imaging conditions, such as rotation and translation, poor illumination, viewpoints variation, and degradation by noise.

Because of the powerful expression ability of deep features, many deep learning-based VLR algorithms have achieved excellent performance. Deep learning-based methods can automatically learn multiple stages of invariant vehicle logo features by using CNNs [22]. Thubsaeng et al. [23] applied their CNN to detect candidate regions and recognize vehicle logos based on a pyramid of HOG and SVM. Soon et al. [24] used the stochastic method of particle swarm optimization to automatically search and optimize a CNN model and hyper-parameters. The fine-tuned and trained CNN ensures good convergence and classification performance. Recently, Soon et al. [25] further used a 7-layer CNN model and the whitening transformation technique to remove redundant adjacent image pixels. The extracted features were sufficiently discriminative to improve recognition accuracy. Recently, a joint framework that simultaneously performs image restoration and recognition was proposed by Chen et al. [26], and it was shown to effectively improve the accuracy of VLR. Many general object recognition algorithms [11, 12] achieved competitive results on general object recognition tasks by using very deep networks; however, they were not designed specifically for VLR. These CNN-based methods do not fully utilize the advantages of low- and high-level features, which contain helpful information to identify different categories to enhance the network robustness.

Some existing VLR systems rely on license plate detection to obtain a coarse region of the vehicle logo using experience-based prior knowledge and then apply a vehicle logo classification technique to perform the VLR task. Huang et al. [6]

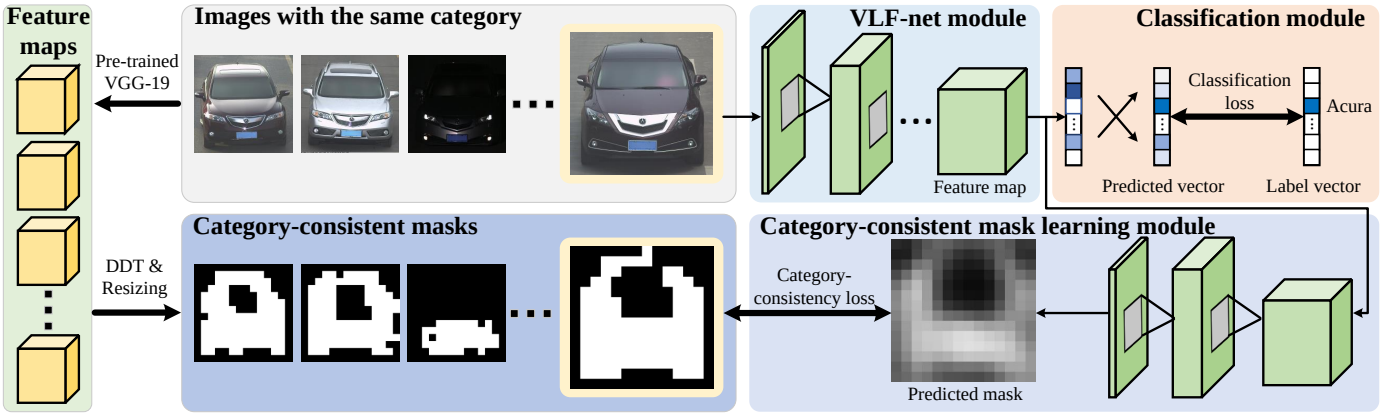


Figure 1: Overall pipeline of the proposed category-consistent deep network learning framework. We take images in the same category to generate category-consistent masks by using a pre-trained VGG-19 model and deep descriptor transformation algorithm. Then, each pair of image and mask is fed into the proposed category-consistent deep network for training. The feature map extracted with the VLF-net module is further fed into the classification module and CCML module. After end-to-end training, these modules can jointly optimize the representation learning of the backbone network.

proposed a new CNN model and used a pretraining strategy based on principal component analysis (PCA [27]) to reduce the computational cost and improve the VLR performance. Li et al. [28] developed a novel distributed system framework and designed a new weight initialization approach to train a MapReduce-based CNN model. The method can not only increase the recognition accuracy but also reduce the computational cost. However, these methods may not work well if there is no license plate in the vehicle logo image or the license plate location is not detected accurately.

Many methods require labor-intensive bounding-box annotations of vehicle logos for training the position and category of vehicle logos based on a general object detection framework [29, 30, 31, 9]. Some algorithms [32, 33, 34] locate the positions of predicted objects by using a region proposal network and then recognize objects based on Faster-RCNN [31]. Other algorithms [35, 36, 8] locate and recognize predicted objects simultaneously based on different versions of the you only look once (YOLO) algorithm [29, 30, 9] and have an advantage over the aforementioned two-stage algorithms in terms of detection speed. Yang et al. [8] designed a modified version of the YOLOv3 model [9] and trained it on their vehicle logo detection dataset. Their YOLOv3-based algorithm achieved good performance on complex scenes. Yu et al. [10] proposed a cascaded deep convolutional network and applied it to their optimized detection framework to detect and recognize vehicle logos precisely. Zhang et al. [37] introduced a novel multi-stage training policy and lightweight network structure with separable convolution for the accurate real-time detection of vehicle logos. Zhou et al. [38] proposed an algorithm to detect vehicle logos under motion blur based on YOLOv3. However, their vehicle logo dataset containing bounding-box annotations required much time to collate. Manual bounding-box annotations of vehicle logos are considerably labor- and cost-intensive. In comparison, our proposed algorithm is a deep learning-based method that can recognize vehicle logos from frontal images of vehicles without the explicit detection of vehicle logo locations.

3. Proposed algorithm

In this section, we introduce the proposed category-consistent deep network learning framework in detail. As illustrated in Fig. 1, our framework consists of three main modules:

- **Vehicle logo feature extraction network module (VLF-net)**: a backbone that extracts high-dimensional deep features from an input image. We use θ_{vlf} to represent the learnable network parameters in VLF-net.
- **Classification module (CM)**: a module that maps the learned feature map produced by VLF-net to a probability distribution vector, $C(I|\theta_{vlf}, \theta_{cls})$, and outputs the vehicle logo category, where θ_{cls} denotes the learnable parameters in the CM.
- **Category-consistent mask learning (CCML) module**: a module that learns a predicted category-consistent region mask, $M'(I|\theta_{vlf}, \theta_{mask})$, based on the feature map produced by VLF-net, where θ_{mask} represents the learnable parameters in the CCML module.

Let $I \in \mathbb{R}^{h\times w\times 3}$ be an input image containing the vehicle logo, where h and w denote the width and height of I, respectively. In the preprocessing step, we employ an unsupervised deep descriptor transformation (DDT) algorithm [39] to generate a category-consistent mask, $M(I) \in \mathbb{R}^{h'\times w'\times 1}$, that coarsely covers the common visual regions for the same category, where h' and w' denote the width and height of $M(I)$, respectively. Then, a deep feature map for I is extracted by using VLF-net. Next, given the number of vehicle logo categories, n, a probability vector, $C(I|\theta_{vlf}, \theta_{cls}) \in \mathbb{R}^{n\times 1}$, is obtained from the feature map by the CM to indicate the predicted probability for each category. The vehicle logo in the input is classified as belonging to the category with the maximum probability. The generated mask, $M(I)$, is fed to the CCML module to enforce VLF-net to pay attention to the common region to learn a better feature representation. The predicted mask, $M'(I|\theta_{vlf}, \theta_{mask}) \in \mathbb{R}^{h'\times w'\times 1}$, indicates the category-consistent region for I.

During training, the CM and CCML module jointly optimize the representation learning of VLF-net with classification loss and category-consistency loss, respectively. During inference, only VLF-net and the CM are activated for the VLR task. Therefore, the DDT algorithm is not required for inference. Note that the proposed algorithm can automatically recognize vehicle logos from frontal images of vehicles without the explicit detection of vehicle logo locations, which requires time-consuming manual annotations.

3.1. Vehicle logo feature extraction network module

Owing to the requirements for robust vehicle logo features used in VLR, we leveraged the advantages of the CNN, which can extract object features, effectively to design VLF-net. In this subsection, we describe our VLF-net in detail by making the best use of high- and low-level features.

We consider that a single image, x_0, is input into a block of a CNN containing L layers. Each layer, l, is defined by a non-linear transformation, $H_l(\cdot)$, composed of multiple consecutive operations. The operations in $H_l(\cdot)$ usually include convolution (Conv), batch normalization (BN), and rectified linear unit (ReLU). An example of a basic network block is shown in Fig. 2 (a).

Let x_l be the output of the l^{th} layer. For a basic convolutional feed-forward network block, the output of the $(l-1)^{th}$ layer, x_{l-1}, is the input to the l^{th} layer. Therefore, the transition of basic connections can be denoted as

$$x_l = H_l(x_{l-1}). \tag{1}$$

The identity shortcut connection was introduced in Res-block [11] for improving the accuracy of image recognition. As shown in Fig. 2 (b), the identity shortcut connection adds an identity function to the output of the non-linear transformation. The output, x_l, of the identity shortcut connection is then defined as

$$x_l = H_l(x_{l-1}) + x_{l-1}. \tag{2}$$

This means that if the identity mapping from x_{l-1} is closer to the optimal function, it should be easier for the remaining transformations (e.g., $H_l(x_{l-1})$) to find the perturbations between the optimal function and identity mapping. Moreover, it can help the gradients and other information to flow easily between the next and previous layers.

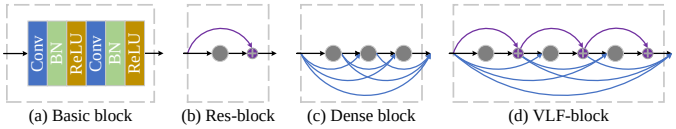
In the dense connection block [12], each layer receives all outputs of preceding layers as input, and its feature maps are used as input to all subsequent layers. The output of the dense connection block can be defined as

$$x_l = H_l([x_0, x_1, \cdots, x_{l-1}]). \tag{3}$$

As shown in Fig. 2 (c), all preceding layers, $x_0, x_1, \cdots, x_{l-1}$, are used as inputs to the l^{th} layer, where $[x_0, x_1, \cdots, x_{l-1}]$ refers to the concatenation operator for $x_0, x_1, \cdots, x_{l-1}$.

In this paper, we propose a new VLF network block by taking advantage of both the identity shortcut connection and dense connection. These connections have been proven in ResNet [11] and DenseNet [12] to efficiently strengthen gradient propagation and alleviate the degradation problem. The proposed VLF-block is defined as

$$x_l = H_l([x_0, x_1, \cdots, x_{l-1}]) + x_{l-1}. \tag{4}$$



(a) Basic block (b) Res-block (c) Dense block (d) VLF-block

Figure 2: Illustration of (a) basic block, (b) Res-block, (c) dense block, and (d) proposed VLF-block. The Res-block, dense block, and VLF-block are composed of multiple basic blocks. The number of basic blocks can be adjusted flexibly.

First, we employ a basic block with a composite function that contains six successive operations, namely, 1×1 Conv, BN, ReLU, 3×3 Conv, BN, and ReLU. Next, the basic block is used to construct a Res-block [11] based on Eq. 2. The basic block and Res-block are then combined to construct the VLF-block. The proposed VLF-block is illustrated in Fig. 2 (d).

The proposed VLF-net is constructed based on the Res-block and VLF-block. The detailed network structure of VLF-net is presented in Table 1. As shown in Fig. 2 (a), each Conv operation is followed by BN and ReLU operations in the basic block. The Res-block and VLF-block are constructed based on the basic block, as shown in Figs. 2 (b and d). As shown in Table 1, we set a dropout probability of 0.5. Layers of "Residual" and "Concat" in Table 1 represent the identity shortcut connection and concatenation operations in Eqs. 2 and 3, respectively. In each VLF-block, we add a 1×1 Conv as a bottleneck layer before each 3×3 Conv, as in ResNet and DenseNet, to reduce the number of input feature maps and improve the computational efficiency. To reduce the dimensionality of the feature maps and improve model compactness, the 1×1 Conv and 2×2 max-pooling (MaxPool) operations are inserted as a transition layer [12] between each VLF-block and Res-block. Table 1 shows that there are 47 convolution layers in VLF-net.

By leveraging the advantages of the shortcut connection and dense connection, VLF-block can prevent the deep network degradation problem and help the network strengthen feature propagation. Therefore, the proposed VLF-net encourages feature reuse in VLR tasks. In the following subsections, we describe how VLF-net is integrated into our framework.

3.2. Classification module

As shown in Fig. 1 (top-right), the proposed CM is based on the feature map extracted by VLF-net. The feature map is taken as input, and the CM outputs the category of the vehicle logo. For example, the vehicle logo of the input image is recognized as the "Acura" category in Fig. 1 (top-right).

The CM adds a global average pooling operator and fully connected (FC) layer to the end of VLF-net. The FC layer acts as a nested linear classifier in this recognition model. In addition, to handle multiple vehicle logo categories, the soft-max function is employed at the end of the CM. The output of the

Table 1: List of detailed operations in the proposed VLF-net. For simplified annotation, each "Conv" layer corresponds to the sequence Conv, BN, and ReLU.

	Layer	No. of filters	Size	Output size	
	Conv	32	3×3	$h \times w$	
	MaxPool	32	2×2 / 2	$h/2 \times w/2$	
VLF-block	Conv	32	1×1		×3
	Conv	32	3×3		
	Residual				
	Concat				
	Conv	64	1×1		
	MaxPool	64	2×2 / 2	$h/2^2 \times w/2^2$	
Res-block	Conv	64	3×3		
	Conv	64	3×3		
	Residual				
VLF-block	Conv	32	1×1		×3
	Conv	64	3×3		
	Residual				
	Concat				
	Conv	128	1×1		
	MaxPool	128	2×2 / 2	$h/2^3 \times w/2^3$	
Res-block	Conv	128	3×3		
	Conv	128	3×3		
	Residual				
VLF-block	Conv	64	1×1		×3
	Conv	128	3×3		
	Residual				
	Concat				
	Conv	256	1×1		
	MaxPool	256	2×2 / 2	$h/2^4 \times w/2^4$	
Res-block	Conv	256	3×3		
	Conv	256	3×3		
	Residual				
VLF-block	Conv	128	1×1		×3
	Conv	256	3×3		
	Residual				
	Concat				
	Conv	512	1×1		
	MaxPool	512	2×2 / 2	$h/2^5 \times w/2^5$	
Res-block	Conv	512	3×3		
	Conv	512	3×3		
	Residual				
VLF-block	Conv	256	1×1		×3
	Conv	512	3×3		
	Residual				
	Concat				
	Conv	1024	1×1		
	MaxPool	1024	2×2 / 2	$h/2^6 \times w/2^6$	
	Conv	512	3×3		
	Conv	1024	3×3		
	Conv	512	3×3	$h/2^6 \times w/2^6$	

soft-max function is a probability feature vector, $C(I|\theta_{vlf}, \theta_{cls})$, containing the probabilities of all vehicle logo categories for the input image. Therefore, the dimension of $C(I|\theta_{vlf}, \theta_{cls})$ is equal to the number of vehicle logo categories, n. The category of the vehicle logo can be recognized by choosing the maximum probability in the predicted probability feature vector, $C(I|\theta_{vlf}, \theta_{cls})$.

The classification loss function, $\mathcal{L}_{cls}$, used in the CM is defined as

$$\mathcal{L}_{cls} = -\sum_{I \in \mathcal{I}} \boldsymbol{l}(I) \cdot \log C(I|\theta_{vlf}, \theta_{cls}), \tag{5}$$

where $\mathcal{I}$ denotes the image set for training, and the label vector, $\boldsymbol{l}(I) \in \mathbb{R}^{n \times 1}$, is the ground-truth one-hot category label for each input image, I.

3.3. Category-consistent mask learning module

Objects in images from the same category usually share some visual commonality, which has been proven to help a learning model to recognize objects [40]. In our work, different images with the same vehicle maker usually have the same visual patterns in the vehicle logo. Therefore, the foreground vehicle logo regions have high correlations among images from the same category of vehicle logo. Here, we enforce our network to emphasize the visually common regions for the same vehicle logo category to further improve the recognition performance. Specifically, a novel CCML module is proposed to help the network to distinguish common regions from the background.

First, we adopt the DDT algorithm [39] to automatically generate category-consistent masks for our deep network learning. The process of this unsupervised image co-localization method is illustrated in Fig. 1 (left). First, we take as input the images with the same vehicle logo categories based on the image-level labels. Then, given a set of images, S, containing m images with the same category, the pre-trained VGG-19 model [41], which was trained on ImageNet, generates a feature map, $F(I|\theta_{vgg}) \in \mathbb{R}^{h'' \times w'' \times d}$, for each image, $I \in S$, in the final CNN layer, where width $h'' = h/2^5$, height $w'' = w/2^5$, depth $d = 512$, and θ_{vgg} are the pre-trained VGG-19 parameters. A large feature set is obtained by gathering these feature maps together. Next, PCA [27] is applied to the feature set along the depth dimension to obtain d eigenvectors and their corresponding eigenvalues. Eigenvector $\xi \in \mathbb{R}^{d \times 1}$ with the largest eigenvalue is applied to each spatial location of F to obtain a heat map, $H \in \mathbb{R}^{h'' \times w'' \times 1}$. Formally, H is expressed as

$$H_{i,j} = \sum_{k=1}^{d} F_{i,j,k}\xi_k, 1 \le i \le h'', 1 \le j \le w''. \tag{6}$$

H is then upsampled by nearest interpolation to the original input size to obtain the upsampled version, $H' \in \mathbb{R}^{h \times w \times 1}$. The nearest interpolation will not change the signs of the numbers because it is a zero-order interpolation method.

Zero thresholding and max connected component analysis [39] are applied on H' to generate a binary category-consistent mask, $M(I) \in \mathbb{R}^{h \times w \times 1}$. As a consequence, with the image-level category labels and pre-trained VGG-19 model, we can automatically generate category-consistent masks for all training

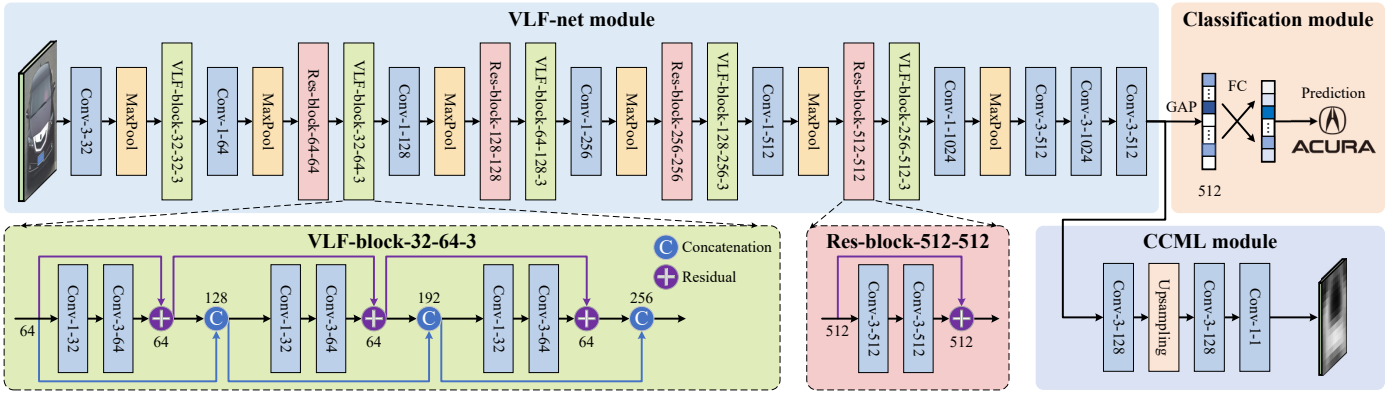


Figure 3: Detailed architecture of the proposed network. The structures of the VLF-blocks and Res-blocks are represented by dashed rectangles. Note that the CCML module is neglected with a simple conditional jump instruction in the forward propagation during inference.

Table 2: List of detailed operations in our CCML module.

Layer	# Filters	Size	Output size
Conv	128	3×3	$h/2^6 \times w/2^6$
Upsampling	128	2×2 / 2	$h/2^5 \times w/2^5$
BN			
ReLU			
Conv	128	3×3	
Conv	1	1×1	
Sigmoid			$h/2^5 \times w/2^5$

images. Note that mask $M(I)$ is used to guide our category-consistent deep network learning, and thus DDT is performed only once as a preprocessing step for the network training.

After generating the category-consistent masks using DDT, we present the novel CCML module. Here, we directly reuse the feature map produced by the VLF-net module as input to the CCML module. As shown in Table 2, the CCML module has 7 successive operations: 3×3 Conv, bilinear upsampling, BN, ReLU, 3×3 Conv, 1×1 Conv, and sigmoid. The CCML generates predicted mask $M'(I|\theta_{vlf}, \theta_{mask}) \in \mathbb{R}^{h' \times w' \times 1}$ with learnable parameters θ_{mask} for the 7 successive operations. Therefore, size $w' \times h'$ of the predicted mask, $M'(I|\theta_{vlf}, \theta_{mask})$, is not equal to that of the category-consistent mask, $M(I)$. Because $M(I)$ is generated from the upsampled heat map, H, with the nearest interpolation, we now employ the nearest dowsampling interpolation scheme on $M(I)$ to obtain the downsampled version, $M_D(I) \in \mathbb{R}^{h' \times w' \times 1}$, without loss of information.

Now that we have the category-consistent mask, $M_D(I)$, and predicted mask, $M'(I|\theta_{vlf}, \theta_{mask})$, and we define a new category-consistency loss function, $\mathcal{L}_{mask}$. Cross-entropy loss has been widely used in many image segmentation applications [42, 43]. Because the category-consistent mask is actually a binary mask, we calculate $\mathcal{L}_{mask}$ with the binary cross-entropy loss. Formally, $\mathcal{L}_{mask}$ is expressed as

$$\begin{aligned}\mathcal{L}_{mask} = -\sum_{I \in \mathcal{I}} \{ & M_D(I) \cdot \log\left(M'(I|\theta_{vlf}, \theta_{mask})\right) \\ & + (1 - M_D(I)) \cdot \log\left(1 - M'(I|\theta_{vlf}, \theta_{mask})\right)\}.\end{aligned} \tag{7}$$

By end-to-end training, our CCML enforces the proposed network to focus on learning the common visual features of images from the same vehicle logo category. As a result, the category-consistent regions are effectively captured with the predicted mask, $M'(I|\theta_{vlf}, \theta_{mask})$. As illustrated in Fig. 1, the learned values in $M'(I|\theta_{vlf}, \theta_{mask})$ naturally distinguish the category-consistent regions from the background. CCML facilitates the learning of discriminative category features and helps the network to learn a better representation to improve the VLR performance.

3.4. Category-consistent deep network learning

The three modules of VLF-net, CM, and CCML in the proposed category-consistent deep network learning framework are trained together in an end-to-end manner.

The final objective function, $\mathcal{L}$, of the proposed framework is now defined as

$$\mathcal{L} = \mathcal{L}_{cls} + \alpha \mathcal{L}_{mask}, \tag{8}$$

where the weight parameter, α, is used for tuning between the classification loss, $\mathcal{L}_{cls}$, and the category-consistency loss, $\mathcal{L}_{mask}$. We set $\alpha = 0.01$ for all experiments reported in this paper.

Fig. 3 shows the detailed architecture of the proposed CNN. During training, the whole network framework is optimized by minimizing the objective function, $\mathcal{L}$. CCML effectively enforces the network to focus on the category-consistent regions while decreasing the influence of irrelevant background. During inference, only VLF-net and the CM are needed for efficient VLR. This is easily implemented using a simple conditional jump instruction in the forward propagation with CCML neglected. Therefore, CCML is only used to learn a better representation and does not introduce additional computational cost at inference time.

4. Experimental setup

4.1. Datasets

We used publicly available datasets containing vehicle logo images (HFUT-VL [13] and XMU [6]) and images of vehicles

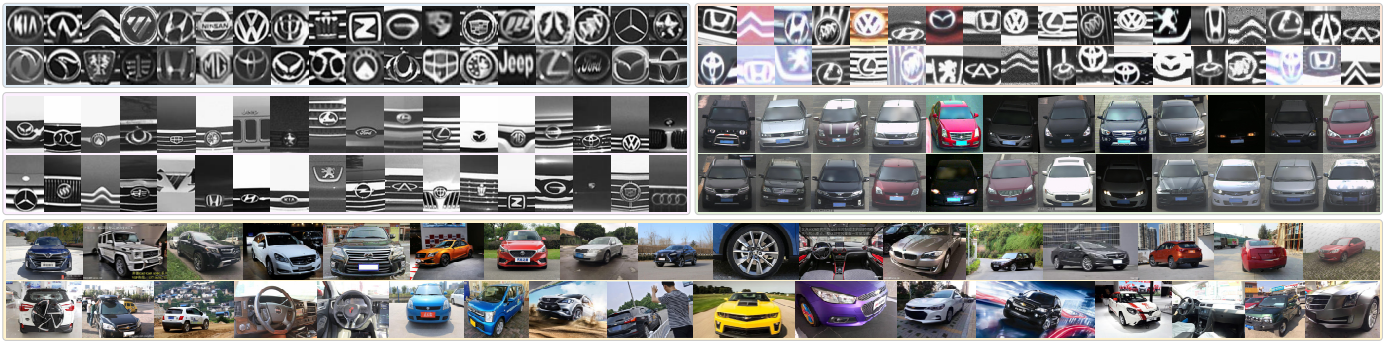


Figure 4: Sample images from the five datasets for experimental evaluation: (top-left) HFUT-VL1 [13], (middle-left) HFUT-VL2 [13], (top-right) XMU [6], (middle-right) CompCars [14], and (bottom) VLD-45 [15].

(CompCars [14] and VLD-45 [15]) to evaluate the performance of our VLF-based VLR algorithms and the superiority of our proposed framework. These datasets are described as follows:

1. The HFUT-VL1 dataset contains 16,000 vehicle logo images with 80 categories of vehicle logos, as shown in Fig. 4 (top-left). Each category consists of 200 images of size 64×64.
2. The HFUT-VL2 dataset has 16,000 images with 80 categories of vehicle logos, as shown in Fig. 4 (middle-left). Each category includes 200 images of size 64×96. Each image contains a vehicle logo and is segmented with a coarse location scheme.
3. The XMU dataset contains 11,500 vehicle logo images with 10 categories of vehicle logos, as shown in Fig. 4 (top-right). Each category includes 1,150 images of size 70×70.
4. The CompCars dataset contains data from two scenarios, one of web nature and another of surveillance nature. In our experiment, the surveillance nature data captured from a front view by surveillance cameras were used as the experimental dataset. The surveillance nature data in CompCars contain 50,000 vehicle images covering 281 vehicle models and 68 categories of vehicle makes, as shown in Fig. 4 (middle-right).
5. The new VLD-45 dataset was built from websites using web crawler technology and Pascal VOC dataset, as shown in Fig. 4 (bottom). The dataset contains 45,000 images and 50,359 vehicle logos with 45 categories. It includes several research challenges, such as small-sized objects, shape deformation, brightness variations, and background noise. The image size varies with a large interval. The largest image size is $7,359 \times 4,422$, whereas the smallest image size is 610×378.

For a fair comparison, for the HFUT-VL1, HFUT-VL2, XMU, and CompCars datasets, we used the same settings as Yu et al. [10], who split each dataset into a training set and testing set without overlap. For each dataset, the training set and testing set accounted for 70% and 30% of images in each category by random selection, respectively.

For the HFUT-VL1, HFUT-VL2, and XMU datasets, during both the training and testing stages, original image sizes were used for HFUT-VL1 and XMU, while images in HFUT-VL2 were resized to 112×112.

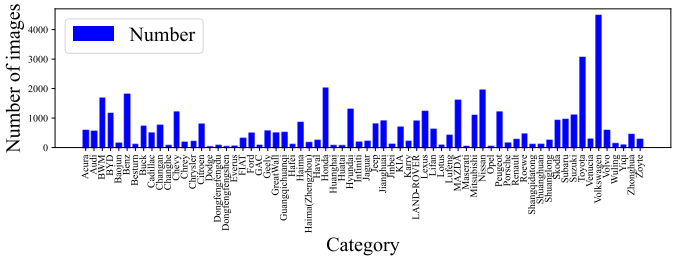


Figure 5: Statistics of the number of images across 68 categories in the surveillance nature data of CompCars.

For the CompCars dataset, we show the statistics of the number of images across 68 categories in the surveillance nature data in Fig. 5. We can see that the distribution of the number of images across categories is not balanced. To reduce the influence of data imbalance, we first apply a random augmentation scheme to categories containing less than 1,000 images in the training set by randomly selecting a random operation, such as vertical and horizontal translation from -0.15 to 0.15, rotation angle from -15° to 15°, horizontal reflection probability of 0.5, HSV saturation from -0.5 to 0.5, and HSV intensity from -0.5 to 0.5. We then sample 1,000 images for each category from the augmented set for training. All images in both the training and testing sets are resized to 256×256. During training, we employ additional random cropping with a size of 224×224.

For the VLD-45 dataset, we follow the settings of Yang et al. [15] for the task of classification. Specifically, we first select 30,000 images with a single object at random. Then, we randomly split images in each category into the training and testing sets with a ratio of 1:1. Therefore, both the training and testing sets contain 15,000 images. All images are resized to 512×512.

During training, we normalize all image data using channel means and standard deviations. An additional random horizontal flip is employed for data augmentation in all datasets.

4.2. *Implementation details*

The proposed VLR model is programed using Python based on the PyTorch framework. In the training optimization stage,

Table 3: Comparison of our algorithm with general CNN recognition models on the HFUT-VL1, HFUT-VL2, XMU, and CompCars datasets. "CM" denotes classification only, while "CM + CCML" denotes classification combined with the proposed CCML module. Top-2 scores are highlighted in bold for each column. An underlined value denotes the greatest performance gain with "CM + CCML" compared to the case with only "CM" for the same network on each test dataset.

Model	Accuracy							
	HFUT-VL1		HFUT-VL2		XMU		CompCars	
	CM	CM + CCML	CM	CM + CCML	CM	CM + CCML	CM	CM + CCML
ResNet-34 [11]	98.52%	98.83%	95.67%	96.17%	**100.0%**	**100.0%**	99.86%	99.86%
ResNet-50 [11]	97.98%	98.44%	94.90%	97.31%	99.91%	99.97%	99.86%	**99.90%**
ResNet-152 [11]	98.29%	98.96%	94.75%	95.40%	94.26%	99.83%	99.86%	99.86%
DenseNet-121 [12]	99.21%	**99.38%**	97.75%	**98.21%**	**100.0%**	**100.0%**	99.86%	99.88%
DenseNet-169 [12]	**99.29%**	99.31%	97.50%	97.92%	**100.0%**	**100.0%**	99.86%	99.88%
DarkNet53 [9]	96.46%	98.54%	91.21%	93.19%	99.16%	99.91%	99.84%	99.86%
Yang [8]	99.17%	99.23%	**97.83%**	98.19%	**100.0%**	**100.0%**	**99.89%**	99.89%
VLF-net	**99.38%**	**99.56%**	**97.79%**	**98.73%**	**100.0%**	**100.0%**	**99.87%**	**99.92%**

Table 4: Comparison of our algorithm with general CNN recognition models on the VLD-45 dataset. "Top-1" and "Top-5" represent the top-1 and top-5 accuracies, respectively. Frames per second (FPS) are averaged over all testing images at inference.

Model	CM					CM + CCML				
	Top-1	Top-5	FPS	Training	# Params	Top-1	Top-5	FPS	Training	# Params
ResNet-34 [11]	76.14%	91.54%	**79.0**	**16.12h**	21.30M	84.10%	95.27%	**80.7**	**17.25h**	21.53M
ResNet-50 [11]	72.35%	90.55%	51.9	28.50h	23.60M	77.86%	93.07%	52.4	29.75h	24.45M
ResNet-152 [11]	76.61%	92.49%	22.9	62.50h	58.23M	84.39%	95.75%	23.1	64.00h	59.08M
DenseNet-121 [12]	**86.17%**	**96.18%**	35.3	35.88h	**7.000M**	**88.61%**	**96.94%**	34.1	36.76h	**7.433M**
DenseNet-169 [12]	78.35%	92.96%	26.3	44.58h	12.56M	88.25%	96.68%	26.5	46.22h	13.25M
DarkNet53 [9]	75.37%	91.65%	49.6	33.20h	40.63M	83.57%	94.96%	49.7	34.12h	41.06M
Yang [8]	65.93%	86.30%	**104.9**	**12.25h**	**4.739M**	72.51%	89.75%	**104.8**	**13.08h**	**4.965M**
VLF-net	**79.27%**	**93.03%**	57.8	27.63h	29.05M	**92.63%**	**98.02%**	57.9	28.50h	29.27M

the back-propagation and stochastic gradient descent algorithms are employed for loss function minimization. The mini-batch size of 128 is used on the HFUT-VL1, HFUT-VL2, and XMU datasets, while a mini-batch size of 32 is used on CompCars. On the VLD-45 dataset, we use a mini-batch size of 8. All datasets and models are trained with 150 epochs. We set the initial learning rate to 0.1 and decay it by a factor of 10 every 30 epochs. Following ResNet [11] and DenseNet [12], we use a weight decay of 10^{-4} and momentum of 0.9.

5. Experimental results and discussion

In this section, we evaluate the performance of the proposed VLR algorithm by comparing it to state-of-the-art algorithms on the aforementioned publicly available HFUT-VL1, HFUT-VL2, XMU, CompCars, and VLD-45 datasets. Several experimental results are presented and extensively discussed.

5.1. Comparison with general CNN recognition models

As shown in Table 3, we compare our algorithm to state-of-the-art general CNN object recognition models on the HFUT-VL1, HFUT-VL2, XMU, and CompCars datasets. Note that all CNN models in Table 3 use the same settings mentioned in Subsection 4.2 and are trained without using pre-trained weights. In addition, we replaced our VLF-net with the compared models in our framework to demonstrate the effectiveness of the proposed CCML. Yang et al.'s algorithm [8] and YOLOv3 [9] recognize objects based on the detection pipeline, and their experimental results are obtained using their backbone networks with our own implementation. Gradients in ResNet [11] and DarkNet [9] flow easily by using identity shortcut connections. DenseNet [12] reuses extracted features to strengthen gradient propagation with dense connections. By combining the advantages of both identity shortcut connections and dense connections, our VLF-based algorithm without CCML achieved recognition accuracies of 99.38%, 97.79%, 100%, and 99.87% on HFUT-VL1, HFUT-VL2, XMU, and CompCars, respectively. By emphasizing the category-consistent common regions, our category-consistent algorithm further improves the VLR accuracies with scores of 99.56%, 98.73%, 100%, and 99.92%, respectively. This experiment also shows that our CCML can benefit most of the compared models. Images in test datasets contain category-irrelevant background information, and thus their corresponding category-consistent masks can provide useful category-consistent indication for recognition. We can see that our proposed CCML improves the performance more obviously on the HFUT-VL1 and HFUT-VL2 datasets than the XMU and CompCars datasets. Because the overall recognition accuracies without CCML for HFUT-VL1 and HFUT-VL2 are lower than those for XMU and CompCars, there are more performance gains with CCML in HFUT-VL1 and HFUT-VL2. Nevertheless, our novel CCML can notably improve the VLR accuracy on all these datasets.

We recorded the frames per second (FPS) for the inference

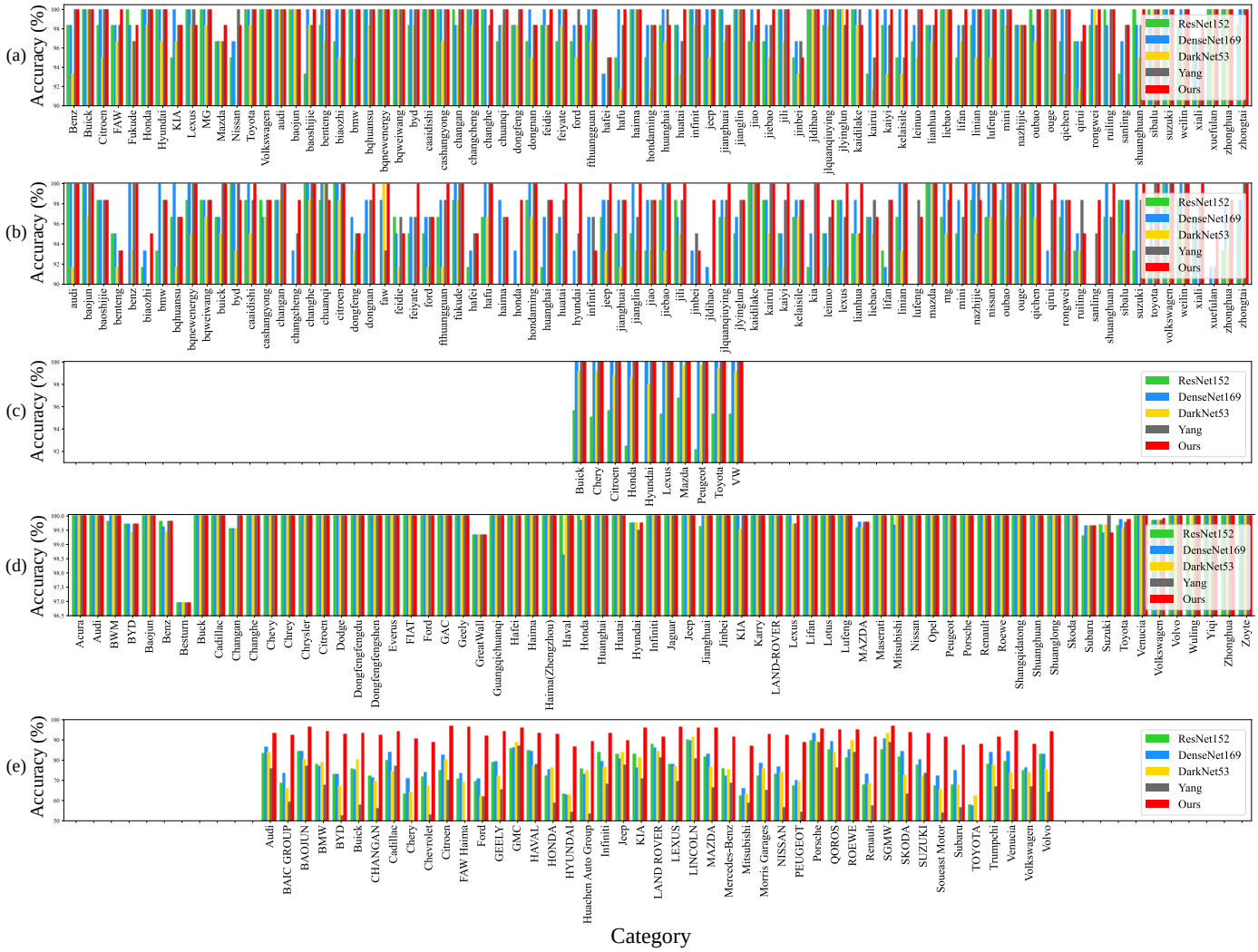


Figure 6: Comparison of recognition accuracy for each category on the (a) HFUT-VL1, (b) HFUT-VL2, (c) XMU, (d) CompCars, and (e) VLD-45 datasets, respectively. Please zoom in for better visualization.

phase on a PC with a 2.80 GHz Intel® Core™ i3-8400 CPU, 16 GB memory, NVIDIA GeForce RTX 2080 GPU, and 64-bit Ubuntu 16.04 operating system. The average performances on HFUT-VL1, HFUT-VL2, XMU, and CompCars were 54.5, 51.3, 53.7, and 34.0 FPS, respectively.

Table 4 shows the comparison of our algorithm to general CNN recognition models on the VLD-45 dataset performed on an NVIDIA Tesla P100 GPU. Because VLD-45 contains a set of images with complex backgrounds, small-scale vehicle logos, shape deformation, and brightness variations, it is difficult to recognize them precisely. We show the top-1 accuracy, top-5 accuracy, FPS at inference, training time, and number of parameters for a comprehensive comparison. Because the degradation problem is alleviated using identity shortcut connections, networks such as ResNet and Darknet achieve good performance for classification. When the networks become deeper, these types of models require more parameters and time to train. By densely reusing shallow features, condensed models, such as DenseNet, can learn more accurate models and contain less trainable parameters but still requiring significant training time. VLF-net takes advantage of both identity shortcut connections and dense connections to achieve top-1 and top-5 accuracies of 79.27% and 93.03%, respectively. With the indication of category-consistent regions provided by CCML, VLF-net can extract more discriminative features to enhance recognition performance with the best top-1 and top-5 accuracies of 92.63% and 98.02%, respectively. The proposed VLF-net uses a combination of DenseNet and ResNet style networks. As shown in Table 4, the number of parameters of VLF-net is slightly larger than that of ResNet-50 and less than that of DarkNet53. VLF-net is also above average in both inference speed and training time. Besides, Table 4 validates that the VLR performance of all compared models is boosted by equipping them with the proposed CCML with only a slight increase in training time (ranging from 0.5 to 1.5 h). Note that our CCML contains only three convolutional layers with 0.22M parameters, which is only approximately 0.76% of the parameters in the whole VLF-net. During inference, only the VLF-net module

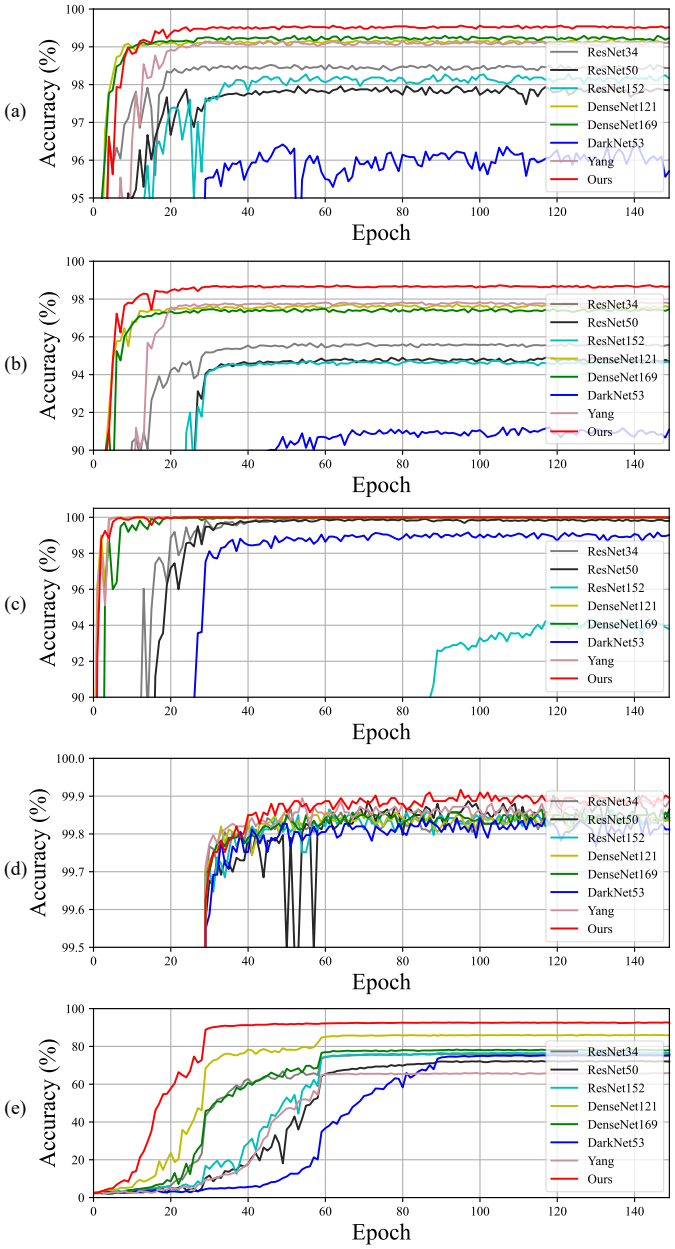


Figure 7: Comparison of validation accuracy for each training epoch on (a) HFUT-VL1, (b) HFUT-VL2, (c) XMU, (d) CompCars, and (e) VLD-45.

and CM are required, and our CCML module can be disabled. Therefore, CCML can effectively improve the recognition performance without introducing extra computation at the inference phase.

We also show comparison of the recognition accuracy for each category on testing datasets in Fig. 6. Overall, our novel category-consistent algorithm outperforms these compared recognition algorithms on these datasets, and our CCML notably improves the model performance in many cases.

To better illustrate the performance of the proposed framework, we compare the validation accuracy for each training epoch on the tested datasets in Fig. 7. We can see that many models fluctuate sharply before the 40^{th} epoch on the HFUT-VL1, HFUT-VL2, and XMU datasets and before the 60^{th} epoch on the CompCars and VLD-45 datasets. Due to the influence of the image background noise, some models do not converge easily during training but are stable at the end, such as ResNet-152 on XMU and DarkNet53 on HFUT-VL2. Although DenseNet-121 and DenseNet-169 converge faster on the HFUT-VL1 and HFUT-VL2 datasets, our algorithm can achieve the highest accuracy after the 40^{th} epoch. In XMU, many models achieve 100% accuracy after convergence, but DenseNet-121, Yang et al.'s [8], and our algorithm can learn faster. On VLD-45, most of the compared models struggle with the extremely complex backgrounds and vehicle logos of small size before the 60^{th} epoch, while our model increases in validation accuracy fast. The highest accuracy on each dataset performed by our algorithm indicates that the feature gradients flow easily with our network, and the learned discriminative category features also help to obtain accuracy gains.

Table 5: Comparison of our algorithm with state-of-the-art VLR algorithms on the HFUT-VL1 and XMU datasets.

Method	Accuracy	
	HFUT-VL1	XMU
Huang [6]	98.90%	99.20%
Peng [17]	97.30%	97.90%
Yu [13]	96.30%	-
Lu [21]	97.80%	98.40%
Yu [7]	98.92%	99.98%
Yang [8]	99.17%	**100.0%**
Yu [10]	99.50%	99.80%
Soon [25]	-	99.13%
Chen [26]	-	**100.0%**
Ours	**99.56%**	**100.0%**

5.2. *Comparison with vehicle logo recognition algorithms*

Table 5 shows the comparison of our algorithm with state-of-the-art VLR algorithms. In this table, the results of Huang et al. [6], Peng et al. [17], Lu et al. [21], and Yu et al. [10] are taken from the paper of Yu et al. [10]. The results of Yu et al. [13], Yu et al. [7], Soon et al. [25], and Chen et al. [26] were obtained directly from the authors' papers.

The algorithms of Peng et al. [17], Yu et al. [13], and Lu et al. [21] are based on a traditional handcrafted feature scheme, and the algorithm of Yu et al. [7] uses non-CNN-based learning of shallow visual features. These methods require a small number of training data. However, their algorithms have difficulty dealing with a wide range of different imaging conditions. The algorithms of Huang et al. [6], Yang et al. [8], Yu et al. [10], Soon et al. [25], and Chen et al. [26] employ CNNs to achieve better performance in general. However, the algorithms of Huang et al. [6], Soon et al. [25], and Chen et al. [26] still have difficulties in extracting high-level logo features with moderate numbers of CNN layers. Although sufficiently deep CNN models are employed by Yang et al. [8] and Yu et al. [10], their algorithms do not take the category-consistent visual commonalities into consideration. By making the best use of high- and low-level features, our VLF-based algorithm can

Table 6: Comparison of model generalizability. All models were trained on VLD-45 and tested on CompCars.

Model	Accuracy	
	CM	CM + CCML
ResNet-34 [11]	42.04%	50.82%
ResNet-50 [11]	34.56%	43.01%
ResNet-152 [11]	44.99%	53.53%
DenseNet-121 [12]	**54.00%**	**54.76%**
DenseNet-169 [12]	45.92%	54.20%
DarkNet53 [9]	47.31%	51.82%
Yang [8]	38.82%	41.34%
VLF-net	**51.01%**	**64.94%**

notably improve the VLR accuracy. Furthermore, with the help of CCML, our VLR algorithm achieves the highest scores of 99.56% and 100% on the HFUT-VL1 and XMU, respectively. In comparison, the methods of Huang et al. [6], Peng et al. [17], and Lu et al. [21] rely on the license plate location to obtain approximate vehicle logo regions for further recognition. The methods of Yu et al. [13], Yu et al. [7], Soon et al. [25], and Chen et al. [26] must train on vehicle logo images segmented by some logo location schemes. The methods of Yang et al. [8] and Yu et al. [10] can recognize vehicle logos from frontal images of vehicles without license plate location, but they require manually annotated bounding boxes for vehicle logos. On the contrary, our novel algorithm does not require the license plate location or manual annotation of bounding boxes. Our algorithm is able to achieve good VLR performance on both vehicle logo images and frontal images of vehicles.

5.3. Model generalizability

Here, we evaluate the model generalizability by training on the VLD-45 dataset while testing on the CompCars dataset. To deal with the inconsistency issue of vehicle logo categories between VLD-45 and CompCars, we set up a category mapping table and eliminated inconsistent categories.

We employed the models trained on VLD-45 to test the performance on CompCars. Table 6 shows the comparison of generalizability of our algorithm with general CNN recognition models. From the table, we can see that the performances of all tested models decrease when testing on CompCars. Note that there are wide disparities between VLD-45 and CompCars in collecting resources, viewpoints, data resolutions, etc., and they belong to different underlying distributions. These models can effectively learn the data distribution of VLD-45 after training, as presented in Table 4. However, the performance reduction may occur because of the inconsistency of distributions between the two datasets. Nevertheless, the proposed VLF-net and DenseNet-121 can still achieve higher performances compared with the other models. Moreover, all tested models can obtain performance benefits from CCML by enforcing intermediate layers to learn more discriminative category-consistent features. As a result, both the proposed VLF-net and CCML have good model generalizability compared with other models.

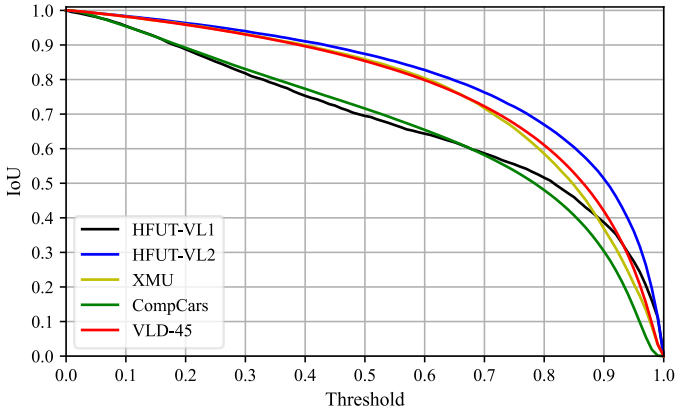


Figure 8: Intersection over union curves with regard to the threshold between DDT-generated category-consistent baseline masks and CCML-predicted masks for the five testing datasets.

5.4. Discussion of CCML

Considering that intersection over union (IoU) is a quantitative measure for evaluating the quality of predicted category-consistent masks, we first generate binary masks by thresholding the soft CCML-predicted masks and then plot IoU curves with regard to the threshold between DDT-generated baseline masks [39] and CCML-predicted masks, as shown in Fig. 8. In this figure, we can see that all IoU scores are greater than 2/3 when the threshold value is 0.5. This means that our predicted masks can effectively cover most of the category-consistent regions for the testing datasets. As a result, CCML can help the backbone network pay attention to discriminative features in category-consistent regions. Nevertheless, as demonstrated in Tables 3 and 4, CCML can be integrated into existing recognition CNNs to improve the VLR performance.

To validate the influence of the weight parameter α and the proposed category-consistency loss, we conducted an ablation study on HFUT-VL1, HFUT-VL2, XMU, and CompCars with different values of α. The corresponding recognition accuracy results are shown in Table 7. Note that when $\alpha = 0$, CCML is actually not used in the network. The VLF-net module without CCML achieves an accuracy of 100% for XMU, and we can see that the introduction of CCML does not lower the recognition performance. For the HFUT-VL1, HFUT-VL2, and CompCars datasets, the recognition accuracies increase with slight fluctuations for $\alpha \in [0.001, 0.01]$, which disappear for $\alpha \in [0.1, 1]$. For VLD-45, the accuracy grows steadily for $\alpha \in [0.001, 0.5]$ and further improves at $\alpha = 1$. There are many images with complex scenes in VLD-45, and CCML can effectively boost the recognition capability of VLF-net by emphasizing category-consistent regions.

Here, we visualize the category-consistent masks generated by DDT and our corresponding predicted masks generated by CCML. Some samples are shown in Fig. 9. To verify the stability of predicted masks in practical applications, where input images may not be fully aligned, we add random rotation and offset to the test images. From the figure, we can see that CCML can automatically generate soft category-consistent masks similar to those generated by DDT. Both DDT and CCML are able

Table 7: Ablation study on the influence of α on recognition accuracy.

Dataset	Weight parameter α							
	0	0.001	0.005	0.01	0.05	0.1	0.5	1
HFUT-VL1	99.38%	99.38%	99.46%	**99.56%**	99.42%	99.21%	98.85%	98.69%
HFUT-VL2	97.79%	98.62%	97.92%	**98.73%**	98.04%	98.25%	98.10%	97.06%
XMU	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**
CompCars	99.87%	**99.92%**	**99.92%**	**99.92%**	**99.92%**	99.90%	99.90%	99.89%
VLD-45	79.27%	89.43%	91.63%	92.63%	92.86%	92.82%	**94.05%**	93.98%

Figure 9: Sample category-consistent masks from experimental datasets colored in green: (top) binary category-consistent baseline masks by DDT, and (bottom) predicted soft masks by CCML.

to coarsely find the category-consistent regions. CCML provides an automatic way to emphasize category-consistent regions, which makes the goal of learning more targeted. CCML helps the network backbone to distinguish the common regions shared by each same vehicle logo category from irrelevant backgrounds to reduce the influence of noise. As a result, category-consistent masks enforce the backbone to focus on the per-category common regions and effectively improve the VLR performance.

5.5. Discussion of recognition under extreme conditions

Now, we discuss VLR performance under extreme conditions. We show some examples of success and failure cases under various extreme conditions collected from the five testing datasets in Fig. 10.

From Fig. 10 (a), we can see that many images are captured under various challenging conditions, such as low illumination, obscuration, shape deformation, and background noise. Our method can still perform well in recognizing these vehicle logo images.

Although our method works well for various vehicle logo images, as demonstrated earlier, it may fail under some extreme conditions. In Fig. 10 (b), we show some examples of failure cases. Note that some misclassified examples are extremely blurry or under very low illumination, and they are difficult to recognize even for human eyes. There are a small number of images containing different kinds of objects instead of vehicles, such as vehicle wheel, motorbike, and steering wheel, and these images may also be misclassified. Some other extreme conditions, such as a side or top view of a car or extremely complex background, would also influence the VLR performance.

6. Conclusions and future work

In this paper, we proposed a novel category-consistent deep network learning framework for enhancing the performance of VLR. By using the characteristics of both the identity shortcut connection and dense connection, our framework can extract hierarchical visual features from a vehicle logo image. Without any experience-based prior knowledge, such as license plate detection or manual bounding-box annotations, our framework can effectively help the backbone to focus on category-consistent regions by localizing discriminative regions in the same category. Consequently, our framework can significantly improve the performance of VLR in both vehicle logo images and frontal images of vehicles. Extensive experimental evaluations and comparisons demonstrated that our novel algorithm can achieve more accurate VLR than existing state-of-the-art algorithms.

In the future, we would like to further improve the VLR accuracy using fine-grained recognition schemes. Moreover, we plan to apply our new VLF-net and the CCML module to other image recognition applications and focus on more challenging recognition tasks with complicated imaging conditions.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments to improve this paper. X. Jin was supported by the National Natural Science Foundation of China (No. 62036010) and the Fundamental Research Funds for the Central Universities. H. Zhao was supported by the Zhejiang Provincial Natural Science Foundation of China (No. LZ21F020001) and the Basic Scientific Research Program of Wenzhou (No. G20200022). Q. He was supported by the Scientific Research Project of Education Department of Zhejiang Province of China (No. Y201941119). H. Huang was supported by the National Natural Science Foundation of China (No. 62072340), Zhejiang Provincial Natural Science Foundation of China (No. LSZ19F020001), and the Key Science and Technology Innovation Project of Wenzhou (No. 2018ZG011).

References

[1] Y. Liu, J. Shen, H. He, Multi-attention deep reinforcement learning and re-ranking for vehicle re-identification, Neurocomputing 414 (2020) 27-35.

[2] S. Shang, J. Shen, J. Wen, P. Kalnis, Deep understanding of big geospatial data for self-driving cars, Neurocomputing 428 (2021) 308-309.

(a) Examples of success cases

(b) Examples of failure cases

Figure 10: Examples of (a) success cases and (b) failure cases under various extreme conditions collected from the five testing datasets.

[3] S. Yu, Y. Wu, W. Li, Z. Song, and W. Zeng, A model for fine-grained vehicle classification based on deep learning, Neurocomputing 257 (2017) 97-103.

[4] Y. Xiang, Y. Fu, and H. Huang, Global relative position space based pooling for fine-grained vehicle recognition, Neurocomputing 367 (2019) 287-298.

[5] L. Lu and H. Huang, Component-based feature extraction and representation schemes for vehicle make and model recognition, Neurocomputing 372 (2020) 92-99.

[6] Y. Huang, R. Wu, Y. Sun, W. Wang, and X. Ding, Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy, IEEE Transactions on Intelligent Transportation Systems 16 (4) (2015) 1951-1960.

[7] Y. Yu, H. Li, J. Wang, H. Min, W. Jia, J. Yu, and C. Chen, A multilayer pyramid network based on learning for vehicle logo recognition, IEEE Transactions on Intelligent Transportation Systems 22 (5) (2021) 3123-3134.

[8] S. Yang, J. Zhang, C. Bo, M. Wang, and L. Chen, Fast vehicle logo detection in complex scenes, Optics & Laser Technology 110 (2019) 196–201.

[9] J. Redmon and A. Farhadi, YOLOv3: An incremental improvement, arXiv preprint (2018) arXiv:1804.02767.

[10] Y. Yu, H. Guan, D. Li, and C. Yu, A cascaded deep convolutional network for vehicle logo recognition from frontal and rear images of vehicles, IEEE Transactions on Intelligent Transportation Systems 22 (2) (2021) 758-771.

[11] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.

[12] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, Densely connected convolutional networks, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 2261-2269.

[13] Y. Yu, J. Wang, J. Lu, Y. Xie, and Z. Nie, Vehicle logo recognition based on overlapping enhanced patterns of oriented edge magnitudes, Computers & Electrical Engineering 71 (2018) 273–283.

[14] L. Yang, P. Luo, C. C. Loy, and X. Tang, A large-scale car dataset for fine-grained categorization and verification, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 3973-3981.

[15] S. Yang, C. Bo, J. Zhang, P. Gao, Y. Li and S. Serikawa, VLD-45: A big dataset for vehicle logo recognition and detection, IEEE Transactions on Intelligent Transportation Systems (2021) doi: 10.1109/TITS.2021.3062113.

[16] H. Pan and B. Zhang, An integrative approach to accurate vehicle logo detection, Journal of Electrical and Computer Engineering (2013) 1-12, article id 391652.

[17] H. Peng, X. Wang, H. Wang, and W. Yang, Recognition of low-resolution Logos in vehicle images based on statistical random sparse distribution, IEEE Transactions on Intelligent Transportation Systems 16 (2) (2015) 681-691.

[18] F. Tafazzoli and H. Frigui, Vehicle make and model recognition using local features and logo detection, in: Proc. IEEE International Symposium on Signal, Image, Video and Communications (ISIVC), IEEE, 2016, pp. 353-358.

[19] J. Zhao and X. Wang, Vehicle-logo recognition based on modified HU invariant moments and SVM, Multimedia Tools and Applications 78 (1) (2017) 75-97.

[20] H. Zhao, H. Guo, X. Jin, J. Shen, X. Mao, and J. Liu, Parallel and efficient approximate nearest patch matching for image editing applications, Neurocomputing 305 (2018) 39–50.

[21] L. Lu and H. Huang, A hierarchical scheme for vehicle make and model recognition from frontal images of vehicles, IEEE Transactions on Intelligent Transportation Systems 20 (5) (2019) 1774-1786.

[22] J. Fang, Y. Zhou, Y. Yu, and S. Du, Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture, IEEE Transactions on Intelligent Transportation Systems 18 (7) (2017) 1782-1792.

[23] W. Thubsaeng, A. Kawewong, and K. Patanukhom, Vehicle logo detection using convolutional neural network and pyramid of histogram of oriented gradients, in: Proc. International Joint Conference on Computer Science and Software Engineering (JCSSE), Chon Buri, 2014, pp. 34-39.

[24] F. C. Soon, H. Y. Khaw, J. H. Chuah, and J. Kanesan, Hyper-parameters optimisation of deep CNN architecture for vehicle logo recognition, IET Intelligent Transport Systems 12 (8) (2018) 939-946.

[25] F. C. Soon, H. Y. Khaw, J. H. Chuah, and J. Kanesan, Vehicle logo recognition using whitening transformation and deep learning, Signal, Image and Video Processing 13 (1) (2019) 111-119.

[26] R. Chen, L. Mihaylova, H. Zhu, and N. Bouaynaya, A deep learning framework for joint image restoration and recognition, Circuits Syst Signal Process 39 (2020) 1561–1580.

[27] K. Pearson, On lines and planes of closest fit to systems of points in space, Philos. Mag. 2 (11) (1901) 559–572.

[28] B. Li and X. Hu, Effective vehicle logo recognition in real-world application using mapreduce based convolutional neural networks with a pretraining strategy, Journal of Intelligent & Fuzzy Systems 34 (3) (2018) 1985–1994.

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

[30] J. Redmon and A. Farhadi, YOLO9000: Better, faster, stronger, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 6517-6525.

[31] S. Ren, K. He, R. Girshick, and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, IEEE Transactions

on Pattern Analysis and Machine Intelligence 39 (6) (2017) 1137-1149.
[32] I. Ansari, Y. Lee, Y. Jeong, and J. Shim, Recognition of car manufacturers using faster R-CNN and perspective transformation, Journal of Korea Multimedia Society 21 (8) (2018) 888-896.
[33] T. Tang, S. Zhou, Z. Deng, H. Zou, and L. Lei, Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining, Sensors (Basel) 17 (2) (2017) article id 336.
[34] Y. Liao, X. Lu, C. Zhang, Y. Wang, and Z. Tang, Mutual enhancement for detection of multiple logos in sports videos, in: Proc. IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 4856-4865.
[35] Q. Peng, W. Luo, G. Hong, M. Feng, Y. Xia, L. Yu, X. Hao, X. Wang, and M. Li, Pedestrian detection for transformer substation based on Gaussian mixture model and YOLO, in: Proc. International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, 2016, pp. 562-565.
[36] T. Tang, Z. Deng, S. Zhou, L. Lei, and H. Zou, Fast vehicle detection in UAV images, in: Proc. International Workshop on Remote Sensing with Intelligent Processing (RSIP), Shanghai, 2017, pp. 1-5.
[37] J. Zhang, S. Yang, C. Bo, and Z. Zhang, Vehicle logo detection based on deep convolutional networks, Computers & Electrical Engineering 90 (2021) 107004.
[38] L. Zhou, W. Min, D. Lin, Q. Han, and R. Liu, Detecting motion blurred vehicle logo in iov using filter-deblurGAN and VL-YOLO, IEEE Transactions on Vehicular Technology 69 (4) 2020 (3604-3614).
[39] X. Wei, C. Zhang, J. Wu, C. Shen, and Z. Zhou, Unsupervised object discovery and co-localization by deep descriptor transformation, Pattern Recognition 88 (2019) 113–126.
[40] M. Zhou, Y. Bai, W. Zhang, T. Zhao, and T. Mei, Look-into-object: Self-supervised structure modeling for object recognition, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11774-11783.
[41] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proc. International Conference on Learning Representations (ICLR), San Diego, CA, USA, 2015.
[42] E. Shelhamer, J. Long, and T. Darrell, Fully convolutional networks for semantic segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (4) (2017) 640-651.
[43] H. Zhao, X. Qiu, W. Lu, H. Huang, and X. Jin, High-quality retinal vessel segmentation using generative adversarial network with a large receptive field, International Journal of Imaging Systems and Technology 30 (3) (2020) 828-842.