Article

Joint Semantic Intelligent Detection of Vehicle Color under Rainy Conditions

Mingdi Hu 1,*,†, Yi Wu 1, Jiulun Fan 1 and Bingyi Jing 2,*,†

1 School of Communications and Information Engineering, Xi’an University of Posts and Telecommunications, Xi’an 710121, China
2 Department of Statistics & Data Science, Southern University of Science and Technology, Shenzhen 518055, China
* Correspondence: humingdi2005@xupt.edu.cn (M.H.); jingby@sustech.edu.cn (B.J.)
† These authors contributed equally to this work.

Abstract: Color is an important feature of vehicles, and it plays a key role in intelligent traffic management and criminal investigation. Existing algorithms for vehicle color recognition are typically trained on data under good weather conditions and have poor robustness for outdoor visual tasks. Fine vehicle color recognition under rainy conditions is still a challenging problem. In this paper, an algorithm for jointly deraining and recognizing vehicle color, (JADAR), is proposed, where three layers of UNet are embedded into RetinaNet-50 to obtain joint semantic fusion information. More precisely, the UNet subnet is used for deraining, and the feature maps of the recovered clean image and the extracted feature maps of the input image are cascaded into the Feature Pyramid Net (FPN) module to achieve joint semantic learning. The joint feature maps are then fed into the class and box subnets to classify and locate objects. The Rain Vehicle Color-24 dataset is used to train the JADAR for vehicle color recognition under rainy conditions, and extensive experiments are conducted. Since the deraining and detecting modules share the feature extraction layers, our algorithm maintains the test time of RetinaNet-50 while improving its robustness. Testing on self-built and public real datasets, the mean average precision (mAP) of vehicle color recognition reaches 72.07%, which beats both sate-of-the-art algorithms for vehicle color recognition and popular target detection algorithms.

Keywords: vehicle color recognition; low–high level joint task; object detection; joint semantic learning; deep neural network; rainy image recovery

1. Introduction

Vehicle information recognition has been applied in the field of intelligent traffic management and criminal investigation. License plate, model, and vehicle color comprise the main vehicle information. Although license plate recognition is a commonly used vehicle information recognition technology [1], it also faces many challenges in criminal investigation and for intelligent traffic law enforcement, as license plates can be easily obscured (partially or fully) or faked/duplicated by criminals. As it can still be identified despite partial occlusion or viewpoint changes, vehicle color recognition is widely applied in video surveillance [2], vehicle detection [3], vehicle tracking [4], automatic driving [5,6], criminal investigation [7], etc. All the above-mentioned tasks inevitably encounter adverse weather conditions, especially rain. This, in turn, adversely affects the performance of object recognition/retrieval, because rain can significantly reduce the contrast of the scene and reduces visibility, compromising image quality. Many scholars have conducted research on how to improve the performance of object detection under rainy conditions.
Vehicle color recognition methods are typically classified as traditional model-driven [8–11] or data-driven deep learning [12–18]. Traditional methods usually use handcrafted feature descriptors to extract visual features and train a classifier to recognize vehicle color. For example, Chen et al. [8] select the region of interest (ROI) of the vehicle to recognize its dominant color and then train a linear support vector machine to classify it. Jeong et al. [9] adopt AdaBoost to classify an HSV histogram of the vehicle’s homogeneity patches into seven color categories.

Deep neural networks have been employed to learn effective feature representations from raw pixels, which has proven to be more powerful than traditional methods. To be more specific, these deep learning methods fall into two groups: general object detection algorithms [19–24] applied to vehicle color recognition [16–18] and algorithms specially designed for color recognition. All these algorithms are trained on datasets with 7–24 colors [8–10,18] obtained under normal weather conditions. Of course, there exists some research to address object recognition under rainy conditions; basically, these proposed methods adopt two-stage instead of end-to-end procedures, which inevitably increases the running time of the entire task.

On the other hand, a number of scholars have paid attention to joint processing of low-level and high-level tasks. Generally, they improve the robustness of object detection [25–28] by embedding domain adaptation, image restoration, style transfer, or other modules into the object detection framework, or by a few-shot transfer learning mechanism [29–32]. These methods have explored the robustness of performance for downstream tasks in many harsh environments except under rainy weather conditions, which motivated our work in the present paper.

In this paper, a Joint Algorithm for Deraining And Recognition (JADAR) is proposed for fine recognition of vehicle color under rainy conditions. The network architecture is shown in Figure 1. To be more specific, we embed the three-layer decoder of UNet-3 into the last three layers of the feature extraction submodule of RetinaNet-50. The main contributions are as follows:

1. JADAR contains far fewer parameters than previous two-stage methods since its subnets, UNET and RetinaNet share the same feature extracting layers. This is of high practical value as the size of outdoor mobile equipment can be substantially reduced.
2. In JADAR, the multi-scale fusion information obtained by cascading feature maps of original rainy images and recovered images are fed into the subsequent class-box subnet. In so doing, multi-scale information across domains is crucially beneficial for fine vehicle color recognition.
3. The joint processing of low-level and high-level tasks can be mutually beneficial. Embedding the image restoration module can help improve the performance of subsequent high-level tasks under severe weather; conversely, the performance of subsequent high-level tasks as evaluation metrics can, in turn, fine-tune the image restoration algorithms.
4. Comprehensive experiments show that our proposed methods outperform basic detection networks and two-stage network and transfer learning methods for the task of color recognition under rainy conditions. Further, our training and testing times are shortened.
Next, related work is introduced in Section 2. **JADAR** is constructed in Section 3. Section 4 shows that our method is superior to the state-of-the-art quantitatively and qualitatively. Section 5 concludes the main content.

### 2. Related Work

There exists some research on vehicle color recognition under normal weather and object detection under adverse weather conditions, which is reviewed below.

#### 2.1. Vehicle Color Recognition under Normal Weather Conditions

Vehicle color recognition methods generally fall into traditional model-based methods [8–11] and data-based deep learning methods [12–18]. Regarding traditional model-based methods, Chen et al. [8] train a linear support vector machine classifier on the region of interest **ROI** in vehicle images based on eight color types; Jeong et al. [9] adopt the multi-class AdaBoost algorithm to classify the color of front-of-vehicle images into seven types; Dule et al. [11] train three classifiers (KNN, ANN, and SVM) for two **ROIs** (smooth hood section and semi-front of vehicle).

Data-based methods have been receiving increasing attention for vehicle color recognition. Hu et al. [12] were the first to apply a convolutional neural network (CNN) with a spatial pyramid strategy to boost the accuracy of vehicle color recognition. Zhang et al. [15] proposed a lightweight CNN for vehicle color recognition. Fu et al. [16] designed **MCFF-CNN** (Multiscale Comprehensive Feature Fusion Convolutional Neural Network) to recognize eight vehicle colors. Hu et al. [18] proposed vehicle color recognition based on a Smooth Modulation Neural Network with Multi-Scale Feature Fusion (**SMNN-MSFF**).

It is worth mentioning that there has been no research on vehicle color detection under bad weather conditions, which is the focus of this paper.

#### 2.2. Object Detection under Adverse Weather Conditions

Bad weather includes rain, snow, haze, etc. The quality of outdoor images or videos collected in these weathers is severely degraded, so target-detection models trained on high-quality images have difficulty handling bad weather. This challenge has been investigated by many scholars with many solutions provided, such as embedding a domain-adaptation module [25–28,31,33,34] into the object detection backbone, such as YOLO, Faster **RCNN**, RetinaNet, etc., two-stages methods consisting of preprocessing and object detection [35–40], or using a few-shot transfer learning mechanism [29–32].

![Figure 1. Framework overview for our JADAR; detailed explanations are in the text.](image-url)
For example, Chen et al. [33] embedded two domain adaptation modules into Faster R-CNN to reduce the domain discrepancy on image level and instance level. Sindagi et al. [31] proposed an unsupervised domain-adapting method to improve generalization of object detection under hazy and rainy conditions. Style transfer is considered in [27], in which the authors construct a cross-domain representation learning method including domain diversification and a multi-domain invariant. Huang et al. [41] combine dual subnet frameworks for object detection under foggy conditions.

Except for the two-stage methods, the above-mentioned methods do not pay special attention to the rainy conditions. However, two-stage methods such as [35–39,42] do pay attention to image deraining instead of object detection. In other words, the joint tasks of deraining and object detection are not taken seriously. Motivated by the above considerations, we propose JADAR for joint semantic intelligent detection of vehicle color in rainy scenes.

3. JADAR Algorithm
3.1. Fusion Network Design

In this paper, a Joint Algorithm for Deraining And Recognition (JADAR) is designed for vehicle color recognition in inclement weather conditions; it is based on RetinaNet-50, as shown in Figure 1. In Figure 1, O is the rainy image input, B is the corresponding clean background image,  is the outputs of the deraining and detecting results. To see results clearly, we zoom in on the recognition results of each car in picture . is the enlarged result of the first car in the picture—the recognition color is silver-gray with a confidence level of 0.91; is the enlarged result of the second car in the picture—the recognition color is black with a confidence level of 0.58; is the enlarged result of the third car in the picture—the recognition color is dark gray with a confidence level of 0.81. The green/blue/purple/orange boxes represent the feature extraction module/UNet-3/information fusion module/class+bbox subnets, respectively. is the regression loss using the smooth L1 loss. is the classification loss using the focal loss. The loss function for deraining is MSE loss. JADAR is trained by the weighted sum of these three losses (see Equation (7)).

The JADAR framework is designed by embedding the three-layer decoder of UNet-3 [43] into the last three sub-blocks of the feature extraction module, as illustrated by the green-tinted box in Figure 1. The whole framework consists of four main modules: image feature extraction module, deraining module, information fusion module, and class+bbox subnets. The rain removal and feature extraction modules share three layers, avoiding extra computational burden. In fact, Section 4.5 shows that JADAR has the same testing time as RetinaNet-50. The last three feature maps and the corresponding recovered feature maps are cascaded together and then fed into their respective class+bbox subnets, which can learn multi-scale joint semantic representations to improve object detection accuracy under rainy conditions. The feature fusion sub-module setting is illustrated in Figure 2.

The overall object function is back-propagated to train the deraining module and to improve rainy image deraining performance recursively. The object detection backbone network uses three-scale class+bbox subnets to leverage multi-scale fusion color feature maps to classify 24 car color types and locate the bounding-box.
Figure 2. Architecture and weights of the proposed network in detail.

3.2. Model Formulation and Model Optimization

Let the physical mechanism of rainy image corruption be

\[ x = y + z \]  \hspace{1cm} (1)\]

where \( x \), \( y \), \( z \) denote rainy image \( O \), recovered clean background image \( B \), and rain layer \( R \), respectively. To tackle the problem of supervised vehicle object detection by color in inclement weather conditions, a joint network is proposed to learn joint semantic representation from an input rainy image \( x \). Let \( y \) denote the corresponding label of rainy image \( x \).

As demonstrated by the green box in Figure 1, the last three feature maps \( f_1(x), f_2(x), f_3(x) \) are taken from the feature extraction sub-blocks of RetinaNet. Then, \( f_1(x) \) is fed into the corresponding last layer of the decoder of \( \text{UNet-3} \), and \( g_1(x) \) is output. Next, \( g_1(x) \) and \( f_2(x) \) are cascaded into the penultimate layer \( \text{UNet-3} \), and \( g_2(x) \) is output; then \( g_2(x) \) and \( f_3(x) \) are cascaded into the last decoder layer of \( \text{UNet-3} \), and \( g_3(x) \) is output. The output of the deraining module \( \hat{y} \) is denoted by \( g_3(x) \). Thus, the mean square error (MSE) loss is used as deraining object function \( L_{\text{der}} \) as follows:

\[ L_{\text{der}} = \frac{1}{n} \sum_{i=1}^{n} \lVert (\hat{y} - y) \rVert^2 \]  \hspace{1cm} (2)\]

where \( n \) is the number of rainy images. Finally, \( f_1(x), g_1(x), f_2(x) \) and \( g_2(x) \) and \( f_3(x) \) and \( g_3(x) \) are cascaded and input into differently scaled \text{class + box} subnets, where joint semantic information is fused, 24 vehicle colors are classified, and box-bounded regressions are achieved; the last cascading output image is denoted \( y' \).

The classification loss function is

\[ L_{\text{cls}}(p_{it}) = - \sum_{i=1}^{C} (a_t(1 - p_{it})^\gamma \log(p_{it}) + (1 - a_t)(p_{it})^\gamma \log(p_{it})) \]  \hspace{1cm} (3)\]

where \( a_t \) is a balancing factor to balance the uneven proportion of positive and negative examples of every vehicle color category, \( C = 24 \) denotes the number of all vehicle color categories, \( \gamma \geq 0 \) is a tunable focusing parameter (we take \( \gamma = 2.0 \) in Section 4 following [24]),
\( t \) is equal to 0 or 1, which denotes the positive or negative sample, \( p_{it} \in [0, 1] \) denotes the prediction probability of the positive sample of the \( i \)-th vehicle color class, and \( 1 - p_{it} \) indicates the prediction probability of negative examples of every vehicle color category \( i \in \{1, 2, \cdots, 24\} \); i.e.,

\[
p_{it} = \begin{cases} rc1_{p_i} & \text{if } t = 1 \\ 1 - p_i & \text{otherwise} \end{cases}
\]

The loss function of the box bounding regression is

\[
L_{\text{reg}} = \frac{1}{n} \sum_{i=1}^{n} L_{\text{reg}}(i),
\]

with

\[
L_{\text{reg}}(i) = \begin{cases} 0.5a^2 & \text{if } |a| < 1 \\ |a| - 0.5 & \text{otherwise} \end{cases}
\]

where \( a = t_i - t_i^* \), and \( t_i = \{t_x, t_y, t_w, t_h\}, t_i^* = \{t_x^*, t_y^*, t_w^*, t_h^*\} \). Here \((x, y)\) denotes the center coordinates of the bounding box, \(w/h\) denotes the width / height, and \(t_i(t_i^*)\) represents the offset of the prediction box (the ground truth box).

Now, \( L_{\text{reg}}(i) \) represents the regression loss for the \( i \)-th image, and \( L_{\text{reg}} \) represents the total regression loss for all images. The total loss function is then given by

\[
L_{\text{tot}} = L_{\text{cls}}(p_{it}) + L_{\text{reg}} + \lambda L_{\text{der}},
\]

where \( \lambda \in [0, 1] \) is a hyperparameter controlling the strength of the image deraining module’s adjustment to the rainy weather target detection performance. In this context, for \( \lambda = 0.5 \), mAP of the proposed network detection is optimal from many ablation experiments. See Section 4.3 for details.

### 4. Experiments

#### 4.1. Experimental Setup

**Implementation Details.** JADAR is trained end-to-end on the *Rain Vehicle Color-24* image set using the *Adam* optimizer [44] to simultaneously learn image deraining, color classification, and object localization on the PyTorch platform. All experiments are implemented on the AutoDL platform with a Tesla P40. The hyper-parameters \( a \) and \( \gamma \) of the classification loss function \( L_{\text{cls}} \) are set to 0.25 and 2, respectively. We divide *Rain Vehicle Color-24* into a training set, a validation set and a testing set at a ratio of 8:1:1. The batch size is 4, the epoch is 100, and the confidence threshold is 0.5. The learning rate is \( 10^{-4} \) for the first 50 epochs, \( 10^{-5} \) for the next 30 epochs, and \( 10^{-6} \) for the last 20 epochs.

**Evaluation Metric.** Generally, object detection uses *IoU* (Intersection over Union) [21], *Precision* (accuracy) [45], *Recall* [45], *AP* (Average Precision) [18], *mAP* (mean Average Precision) [41], or other evaluation metrics; these concepts are well-known, so we list the formulas in brief:

\[
\text{IoU} = \frac{A \cap B}{A \cup B}
\]

where \( A/B \) denotes GT (bounding box of the object) / the prediction bounding box.

Mathematical definitions of *Precision* and *Recall* are as follows:

\[
\text{precision} = \frac{TP}{TP + FP}
\]

\[
\text{recall} = \frac{TP}{TP + FN}
\]

where \( TP \) is true positives (correctly predicted as positive), \( FP \) is false positives (incorrectly predicted as positive), and \( FN \) is false negatives (failed to predict a positive).
\[ AP = \int_{0}^{1} p(r)dr \]  \hspace{1cm} (11)

where \( p \) is Precision, and \( r \) is Recall.

The \( mAP \) (mean Average Precision) is the average of \( AP \), so \( mAP \) is calculated by

\[ mAP = \frac{\sum_{i=1}^{N} AP_{i}}{N} \]  \hspace{1cm} (12)

where \( N \) is the number of categories.

4.2. Datasets

4.2.1. Synthetic Dataset Rain Vehicle Color-24

Few datasets are available for vehicle color recognition under rainy weather conditions. All our experiments are conducted on enhanced Rain Vehicle Color-24 [46], from which some examples are illustrated in Figure 3.

4.2.2. Real Rain Vehicle Datasets: RID and RIS

Li et al. collected two real rainy image vehicle datasets, RID and RIS [38], for testing object detection. RID is rainy images collected from in-vehicle cameras while driving on rainy days, and RIS is surveillance rainy images collected from network traffic surveillance cameras during rainy weather conditions. The two datasets differ in many aspects: rainfall type, image quality, target size and angle, etc. They represent real-world application scenarios where deraining may be required. RID includes 2495 images, and its rainy image effect is closest to “raindrops” on the camera lens. RIS includes 2048 images, and its rainy image effect is closest to “rain and fog” (many cameras have fog condensation when it rains, and lower resolutions also cause more fog effects) [47]. Due to the highly complex scenes of these two rainy image datasets, it is a challenging dataset, and we choose these two datasets for testing to better illustrate the effectiveness of our proposed algorithm. Examples of these two datasets are given in Figure 4.
4.3. Ablation Study

To determine the optimal design of our proposed framework, we train four combinations on the Rain Vehicle Color-24 dataset: RetinaNet, JADAR1, JADAR2, and JADAR. All these models are trained and tested on Rain Vehicle Color-24 using different loss functions: $\lambda = 0, 0.1, 1.0,$ and $0.5$, respectively. Figure 5 shows that the testing $mAP$ values of the JADAR1, JADAR2, and JADAR models are $2.92\%$, $-3.99\%$, and $4.3\%$ higher, respectively, than the RetinaNet model, which clarifies that joint semantic feature extraction is beneficial to improve vehicle color recognition performance under rainy weather conditions. Referring to Table 1, when the hyper-parameter $\lambda$ is $0.1$, the rain removal module provides a weak assisting effect on vehicle color recognition under rainy weather conditions. When $\lambda$ is $1.0$, it plays the opposite effect. When $\lambda$ is $0.5$, JADAR performs best; so we choose this value in our method.

![Figure 5. APs of different models on the Rain Vehicle Color-24 test set. The x-axis represents the average precision, and the y-axis represents the color categories (24 categories in total).](image)
Table 1. The mAPs using different values for the loss function coefficient $\lambda$ of the deraining module in JADAR on the RVC-24 test set.

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>Model</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>JADAR1</td>
<td>70.69%</td>
</tr>
<tr>
<td>0.5</td>
<td>JADAR</td>
<td>72.07%</td>
</tr>
<tr>
<td>1.0</td>
<td>JADAR2</td>
<td>63.78%</td>
</tr>
</tbody>
</table>

4.4. Experiments and Analysis

4.4.1. Results on Synthetic Datasets

In this section, our proposed algorithm, the vehicle color recognition method, the target detection method, the two-stage method combining rain removal with target detection, and the transfer learning method are compared.

To discuss vehicle color recognition performance, JADAR and SMNN-MSFF [18] are compared. Both are trained on Rain Vehicle Color-24 training subset and tested on its test subset. The quantitative results are shown in the second column of Table 2. These quantitative results confirm that the mAP of our method reaches 72.07%, which is 23.49% higher than SMNN-MSFF. The qualitative results are shown in Figure 6. JADAR outperforms SMNN-MSFF under rainy conditions; for example, there are five vehicles recognized by JADAR, while three vehicles are recognized by SMNN-MSFF. A white vehicle is recognized by JADAR with a confidence score of 0.79, while SMNN-MSFF recognizes it with a confidence score of 0.62.

To compare object detection performance, JADAR, RetinaNet [24], Faster R-CNN [19], SSD [20], and YOLO V3 [21] are compared qualitatively and evaluated by mAP quantitatively. In our experiments, the loss function and settings (i.e., scale, anchor or default box, backbone network, classifier, etc.) of each compared method remains unchanged from the original work. Furthermore, all methods are trained on the Rain Vehicle Color-24 dataset and tested on its test set. The qualitative results of JADAR, Faster R-CNN, YOLO V3, SSD, and RetinaNet for vehicle color recognition in rain are shown in Figures 7 and 8. As can be seen from the figures, our proposed JADAR outperforms other models for fine vehicle color recognition. The quantitative results show that the proposed JADAR is 11.42%, 22.19%, 5.74%, and 4.3% better than Faster R-CNN, YOLO V3, SSD and RetinaNet, respectively, from Table 2.

To compare recognition performances of different joint methods, three state-of-the-art rain removal methods, i.e., LPNet [35], PreNet [48], and RCDNet [49], are chosen to first derain the images, and then RetinaNet is leveraged to recognize vehicle colors. These methods are denoted LR, PR, and RR. Figures 9 and 10 give qualitative comparisons of our JADAR and three two-stage methods for vehicle color recognition under rainy weather conditions. JADAR performs better than other models. From Table 3, our JADAR is 15.56%, 20.37%, and 2.06% higher than LR, PR, and RR, respectively.

To compare with transfer learning methods, two domain-adaptation methods, DaFaster [33] and ATF [50], are compared with JADAR. Here, the VC-24 is the source domain, and Rain Vehicle Color-24 is the target domain; they are leveraged to train the above algorithms. From the 5-th and 6-th columns of Table 3, our method is 25.95% and 9.14% better.
than Da-faster and ATF, respectively. The qualitative results in Figures 9 and 10 show that JADAR identifies more vehicles with higher confidence than the other two methods.

Figure 7. Example 1 of test results of JADAR and object detection methods on the Rain Vehicle Color-24 test set.

Figure 8. Example 2 of test results of JADAR and object detection methods and domain adaptation methods on the Rain Vehicle Color-24 test set.

Figure 9. Example 1 of test results of JADAR and two-stage methods on the Rain Vehicle Color-24 test set.
Table 2. Comparison of recognition accuracy of 24 colors for different network classifications: SM, SMNN-MSFF; FR, Faster RCNN; Yolo, Yolo V3; RN, RetinaNet.

<table>
<thead>
<tr>
<th>Category (R, G, B)</th>
<th>SM</th>
<th>FR</th>
<th>Yolo</th>
<th>SSD</th>
<th>RN</th>
<th>JADAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>white (255, 255, 255)</td>
<td>0.60</td>
<td>0.96</td>
<td>0.95</td>
<td>0.95</td>
<td>0.97</td>
<td>0.97</td>
</tr>
<tr>
<td>black (0, 0, 0)</td>
<td>0.69</td>
<td>0.69</td>
<td>0.92</td>
<td>0.93</td>
<td>0.93</td>
<td>0.95</td>
</tr>
<tr>
<td>orange (237, 145, 33)</td>
<td>0.77</td>
<td>0.90</td>
<td>0.93</td>
<td>0.95</td>
<td>0.97</td>
<td>0.94</td>
</tr>
<tr>
<td>silver-gray (128, 138, 135)</td>
<td>0.31</td>
<td>0.81</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td>0.85</td>
</tr>
<tr>
<td>grass-green (0, 255, 0)</td>
<td>0.82</td>
<td>0.93</td>
<td>0.95</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
</tr>
<tr>
<td>dark-gray (128, 128, 105)</td>
<td>0.43</td>
<td>0.69</td>
<td>0.63</td>
<td>0.67</td>
<td>0.64</td>
<td>0.73</td>
</tr>
<tr>
<td>dark-red (156, 102, 31)</td>
<td>0.48</td>
<td>0.79</td>
<td>0.75</td>
<td>0.88</td>
<td>0.91</td>
<td>0.91</td>
</tr>
<tr>
<td>gray (192, 192, 192)</td>
<td>0.31</td>
<td>0.41</td>
<td>0.15</td>
<td>0.31</td>
<td>0.27</td>
<td>0.36</td>
</tr>
<tr>
<td>red (255, 0, 0)</td>
<td>0.44</td>
<td>0.56</td>
<td>0.60</td>
<td>0.76</td>
<td>0.77</td>
<td>0.83</td>
</tr>
<tr>
<td>cyan (0, 255, 255)</td>
<td>0.62</td>
<td>0.82</td>
<td>0.81</td>
<td>0.87</td>
<td>0.87</td>
<td>0.93</td>
</tr>
<tr>
<td>champagne (255, 227, 132)</td>
<td>0.28</td>
<td>0.74</td>
<td>0.58</td>
<td>0.73</td>
<td>0.66</td>
<td>0.69</td>
</tr>
<tr>
<td>dark-blue (25, 25, 112)</td>
<td>0.52</td>
<td>0.79</td>
<td>0.77</td>
<td>0.75</td>
<td>0.86</td>
<td>0.81</td>
</tr>
<tr>
<td>blue (0, 0, 255)</td>
<td>0.56</td>
<td>0.69</td>
<td>0.55</td>
<td>0.69</td>
<td>0.67</td>
<td>0.69</td>
</tr>
<tr>
<td>dark-brown (94, 38, 18)</td>
<td>0.35</td>
<td>0.18</td>
<td>0.40</td>
<td>0.47</td>
<td>0.37</td>
<td>0.50</td>
</tr>
<tr>
<td>brown (128, 128, 42)</td>
<td>0.35</td>
<td>0.10</td>
<td>0.22</td>
<td>0.34</td>
<td>0.30</td>
<td>0.21</td>
</tr>
<tr>
<td>yellow (255, 255, 0)</td>
<td>0.35</td>
<td>0.83</td>
<td>0.74</td>
<td>0.92</td>
<td>0.86</td>
<td>0.81</td>
</tr>
<tr>
<td>lemon-yellow (255, 215, 0)</td>
<td>0.57</td>
<td>0.99</td>
<td>0.73</td>
<td>0.75</td>
<td>0.75</td>
<td>0.94</td>
</tr>
<tr>
<td>dark-orange (210, 105, 30)</td>
<td>0.32</td>
<td>0.28</td>
<td>0.44</td>
<td>0.47</td>
<td>0.90</td>
<td>0.53</td>
</tr>
<tr>
<td>dark-green (48, 128, 20)</td>
<td>0.59</td>
<td>0.18</td>
<td>0.00</td>
<td>0.34</td>
<td>0.44</td>
<td>0.55</td>
</tr>
<tr>
<td>red-orange (255, 97, 0)</td>
<td>0.38</td>
<td>0.07</td>
<td>0.00</td>
<td>0.52</td>
<td>0.00</td>
<td>0.05</td>
</tr>
<tr>
<td>earthy-yellow (184, 134, 11)</td>
<td>0.68</td>
<td>0.45</td>
<td>0.00</td>
<td>0.28</td>
<td>0.93</td>
<td>1.00</td>
</tr>
<tr>
<td>green (0, 255, 0)</td>
<td>0.18</td>
<td>0.85</td>
<td>0.00</td>
<td>0.97</td>
<td>0.60</td>
<td>0.33</td>
</tr>
<tr>
<td>pink (255, 192, 203)</td>
<td>0.84</td>
<td>0.54</td>
<td>0.00</td>
<td>0.52</td>
<td>0.77</td>
<td>0.75</td>
</tr>
<tr>
<td>purple (160, 32, 240)</td>
<td>0.22</td>
<td>0.03</td>
<td>0.00</td>
<td>0.06</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>mAP</td>
<td>48.58%</td>
<td>60.65%</td>
<td>49.88%</td>
<td>66.33%</td>
<td>67.77%</td>
<td>72.07%</td>
</tr>
</tbody>
</table>

Rainy image | JADAR (o0.98) | LR (o0.92) | PR (o0.87) |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RR (o0.89)</td>
<td>Da-faster (o0.91)</td>
<td>ATF (o0.97)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 10. Example 2 of test results of JADAR and two-stage methods and domain-adaptation methods on the Rain Vehicle Color-24 test set.
Table 3. Comparison of recognition accuracy of 24 colors for different network classifications.

<table>
<thead>
<tr>
<th>Category (R, G, B)</th>
<th>LR</th>
<th>PR</th>
<th>RR</th>
<th>Da-Faster</th>
<th>ATF</th>
<th>JADAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>white (255, 255, 255)</td>
<td>0.88</td>
<td>0.86</td>
<td>0.95</td>
<td>0.74</td>
<td>0.93</td>
<td>0.97</td>
</tr>
<tr>
<td>black (0, 0, 0)</td>
<td>0.75</td>
<td>0.75</td>
<td>0.92</td>
<td>0.61</td>
<td>0.87</td>
<td>0.95</td>
</tr>
<tr>
<td>orange (237, 145, 33)</td>
<td>0.91</td>
<td>0.92</td>
<td>0.96</td>
<td>0.73</td>
<td>0.90</td>
<td>0.94</td>
</tr>
<tr>
<td>silver-gray (128, 138, 135)</td>
<td>0.71</td>
<td>0.66</td>
<td>0.82</td>
<td>0.63</td>
<td>0.63</td>
<td>0.85</td>
</tr>
<tr>
<td>grass-green (0, 255, 0)</td>
<td>0.84</td>
<td>0.88</td>
<td>0.95</td>
<td>0.73</td>
<td>0.92</td>
<td>0.96</td>
</tr>
<tr>
<td>dark-gray (128, 128, 105)</td>
<td>0.55</td>
<td>0.44</td>
<td>0.73</td>
<td>0.38</td>
<td>0.65</td>
<td>0.73</td>
</tr>
<tr>
<td>dark-red (156, 102, 31)</td>
<td>0.67</td>
<td>0.75</td>
<td>0.87</td>
<td>0.54</td>
<td>0.66</td>
<td>0.91</td>
</tr>
<tr>
<td>gray (192, 192, 192)</td>
<td>0.27</td>
<td>0.20</td>
<td>0.37</td>
<td>0.19</td>
<td>0.27</td>
<td>0.36</td>
</tr>
<tr>
<td>red (255, 0, 0)</td>
<td>0.74</td>
<td>0.70</td>
<td>0.87</td>
<td>0.46</td>
<td>0.63</td>
<td>0.83</td>
</tr>
<tr>
<td>champagne (255, 227, 132)</td>
<td>0.81</td>
<td>0.78</td>
<td>0.92</td>
<td>0.75</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>dark-blue (25, 25, 112)</td>
<td>0.58</td>
<td>0.30</td>
<td>0.67</td>
<td>0.16</td>
<td>0.89</td>
<td>0.81</td>
</tr>
<tr>
<td>blue (0, 0, 255)</td>
<td>0.33</td>
<td>0.48</td>
<td>0.53</td>
<td>0.44</td>
<td>0.92</td>
<td>0.69</td>
</tr>
<tr>
<td>dark-brown (94, 38, 18)</td>
<td>0.44</td>
<td>0.32</td>
<td>0.56</td>
<td>0.37</td>
<td>0.75</td>
<td>0.50</td>
</tr>
<tr>
<td>brown (128, 128, 42)</td>
<td>0.33</td>
<td>0.25</td>
<td>0.45</td>
<td>0.06</td>
<td>0.00</td>
<td>0.21</td>
</tr>
<tr>
<td>yellow (255, 255, 0)</td>
<td>0.79</td>
<td>0.77</td>
<td>0.89</td>
<td>0.39</td>
<td>1.00</td>
<td>0.81</td>
</tr>
<tr>
<td>lemon-yellow (255, 215, 0)</td>
<td>0.77</td>
<td>0.78</td>
<td>0.97</td>
<td>0.06</td>
<td>0.00</td>
<td>0.94</td>
</tr>
<tr>
<td>dark-orange (210, 105, 30)</td>
<td>0.56</td>
<td>0.18</td>
<td>0.90</td>
<td>1.00</td>
<td>0.00</td>
<td>0.53</td>
</tr>
<tr>
<td>dark-green (48, 128, 20)</td>
<td>0.65</td>
<td>0.50</td>
<td>0.95</td>
<td>0.22</td>
<td>0.07</td>
<td>0.55</td>
</tr>
<tr>
<td>red-orange (255, 97, 0)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.39</td>
<td>1.00</td>
<td>0.05</td>
</tr>
<tr>
<td>earthy-yellow (184, 134, 11)</td>
<td>0.05</td>
<td>0.67</td>
<td>0.25</td>
<td>0.01</td>
<td>0.40</td>
<td>1.00</td>
</tr>
<tr>
<td>green (0, 255, 0)</td>
<td>0.06</td>
<td>0.13</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.33</td>
</tr>
<tr>
<td>pink (255, 192, 203)</td>
<td>0.44</td>
<td>0.67</td>
<td>0.67</td>
<td>0.92</td>
<td>0.93</td>
<td>0.75</td>
</tr>
<tr>
<td>purple (160, 32, 240)</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

mAP 56.51% 51.70% 70.01% 46.12% 62.93% 72.07%

4.4.2. Results on Real Datasets

We train JADAR, RetinaNet, Faster RCNN, SSD, YOLO V3, LR, PR, RR, Da-faster, and ATF on Rain Vehicle Color-24 and test them on real rainy image vehicle datasets, RID and RIS. The test results are shown in Figures 11-14. As can be seen from these figures, the test results of JADAR on the real datasets, RID and RIS, are generally better than those of other methods. As can be seen from Figure 11, the JADAR and SSD algorithms can correctly identify the two cars in the picture; Yolo V3 can also identify the two cars in the picture, but the black color is mistakenly identified as silver-gray; while the other three algorithms can hardly identify any vehicles in the picture. Referring to Figure 12, because the recognition effects of Faster RCNN and SSD are better than others’, we find a limitation of JADAR in recognizing small targets. Referring to Figure 13, all algorithms can identify the color of the vehicle in the image but with different confidence values; specifically, ATF has the highest confidence value for blue vehicle, with 0.98. However, Figure 14 shows that only JADAR and ATF can identify a certain white vehicle.
Rainy image  
JADAR (b0.7)  
SMNN-MSFF (b0)  
Faster RCNN (b0.59)  
Yolo V3 (b0)  
SSD (b1.0)  
RetinaNet (b0)

Figure 11. Test results of JADAR and color recognition and object detection methods on the RID.

Rainy image  
JADAR (r0.58)  
SMNN-MSFF (r0)  
Faster RCNN (r0.97)  
Yolo V3 (r0)  
SSD (r0.57)  
RetinaNet (r0)

Figure 12. Test results of JADAR and color recognition and object detection methods on the RIS.

Rainy image  
JADAR (b0.82)  
LR (b0.89)  
PR (b0.68)  
RR (b0.92)  
Da-faster (b0.78)  
ATF (b0.98)

Figure 13. Test results of JADAR and two-stage and domain-adaptation methods on the RID.

Rainy image  
JADAR (w0.62)  
LR (w0)  
PR (w0)  
RR (w0)  
Da-faster (w0)  
ATF (w0.69)

Figure 14. Test results of JADAR and two-stage and domain-adaptation methods on the RIS.
4.5. Inference Time

In order to compare the test time of all methods, all network models are tested on a testing subset with an input of $1920 \times 1080$ images. The test times are shown in Table 4. JADAR takes 1.7 s per image on a single Tesla P40 GPU, which is the same as for RetinaNet, but JADAR is 21.8, 1.1, 4.4, 0.8, and 0.9 seconds faster than LR, PR and RR, Da-faster, and ATF, respectively. Therefore, although JADAR has one more decoder module than RetinaNet, it still maintains its original high detection speed.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RN</th>
<th>LR</th>
<th>PR</th>
<th>RR</th>
<th>Da-Faster</th>
<th>ATF</th>
<th>JADAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed (s)</td>
<td>1.7</td>
<td>23.5</td>
<td>2.8</td>
<td>6.1</td>
<td>2.5</td>
<td>2.6</td>
<td>1.7</td>
</tr>
</tbody>
</table>

5. Conclusions

In this paper, we study vehicle color recognition under rainy conditions and propose a joint semantics learning method JADAR, which is designed by embedding UNet-3 into RetinaNet. The UNet module achieves rainy image removal and restores the clean background image. The recovered background image and the rainy image are input together into the class + bbox sub-module of RetinaNet network to accurately extract the joint semantic of the vehicle color features maps. JADAR outperforms other methods under rainy as well as normal conditions for fine vehicle color recognition. Extensive experimental results show that the mAP of the proposed method reaches $72.07\%$ in identifying 24 colors under rainy conditions. Because our algorithm is trained on synthetic datasets, its generalization is not guaranteed. In the future, semi-supervised or few-shot learning is planned to further improve the generalization and realizability of the algorithm. As a further research topic, one can consider fusing overlap functions and fuzzy (rough) sets (see [51–55]) to develop the method of this paper.

Author Contributions: Writing—original draft preparation, M.H.; Experiments and editing, Y.W.; review, J.F., writing—review, B.J. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the National Natural Science Foundation of China (no. 62071378), the Shaanxi Province International Science and Technology Cooperation Program (no. 2022KW-04), and the Xi’an Science and Technology Plan Project (no. 21XJZ0072).

Data Availability Statement: The data that support the findings of this study are openly available at hummingdi2005@github.com.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References


