remote sensing

MDPI

Article

Traffic Sign Detection and Recognition Using Multi-Frame Embedding of Video-Log Images

Jian Xu, Yuchun Huang * and Dakan Ying

School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China; jian_xu@whu.edu.cn (J.X.); dakan_ying@whu.edu.cn (D.Y.)
* Correspondence: hycwhu@whu.edu.cn

Abstract: The detection and recognition of traffic signs is an essential component of intelligent vehicle perception systems, which use on-board cameras to sense traffic sign information. Unfortunately, issues such as long-tailed distribution, occlusion, and deformation greatly decrease the detector's performance. In this research, YOLOv5 is used as a single classification detector for traffic sign localization. Afterwards, we propose a hierarchical classification model (HCM) for the specific classification, which significantly reduces the degree of imbalance between classes without changing the sample size. To cope with the shortcomings of a single image, a training-free multi-frame information integration module (MIM) was constructed, which can extract the detection sequence of traffic signs based on the embedding generated by the HCM. The extracted temporal detection information is used for the redefinition of categories and confidence. At last, this research performed detection and recognition of the full class on two publicly available datasets, TT100K and ONCE. Experimental results show that the HCM-improved YOLOv5 has a *mAP* of 79.0 in full classes, which exceeds that of state-of-the-art methods, and achieves an inference speed of 22.7 FPS. In addition, MIM further improves model performance by integrating multi-frame information while only slightly increasing computational resource consumption.

Keywords: traffic sign; intelligent vehicle; long-tailed distribution; anomalies; embedding; information integration

check for updates

Citation: Xu, J.; Huang, Y.; Ying, D. Traffic Sign Detection and Recognition Using Multi-Frame Embedding of Video-Log Images. *Remote Sens.* **2023**, *15*, 2959. https://doi.org/10.3390/rs15122959

Academic Editors: Mennatullah Siam and Xinshuo Weng

Received: 10 April 2023
Revised: 31 May 2023
Accepted: 3 June 2023
Published: 6 June 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Traffic sign detection and recognition is an essential component of automated driving assistance systems, which can provide critical road guidance information. As illustrated in Figure 1a, traffic signs are typically classified into three types: warning, prohibitory, and mandatory. Each of these categories can be further subdivided to provide a more detailed range of guidance information, such as road types, prohibitions, speed limits, and height limits. Traffic sign detection and recognition require the precise location and classification of traffic signs in the vehicle image.

Traffic signs are designed with distinct shapes such as squares, circles, and triangles, as well as distinct red, yellow, and blue colors to highlight the sign. Traditional detection and recognition methods are thus achieved by designing manual feature descriptors for traffic sign detection and recognition. The use of sliding windows to find high-probability regions in an image containing traffic signs is one example [1]. Researchers have also attempted to determine adaptive segmentation thresholds for traffic sign extraction and classification by computing histograms of images [2,3]. Color space has also been used, for example, in segmentation in HSV [4]. Researchers extracted SURF feature points from signs and used corroding images to match them [5]. In some studies, more complex feature descriptors, such as coarse localization of signs based on the Hough transform [6], were used for sign extraction. Traditional digital morphology-based methods typically necessitate clear images with high-resolution signs and no anomalies to interfere, making them difficult to apply in complex real-world scenarios.

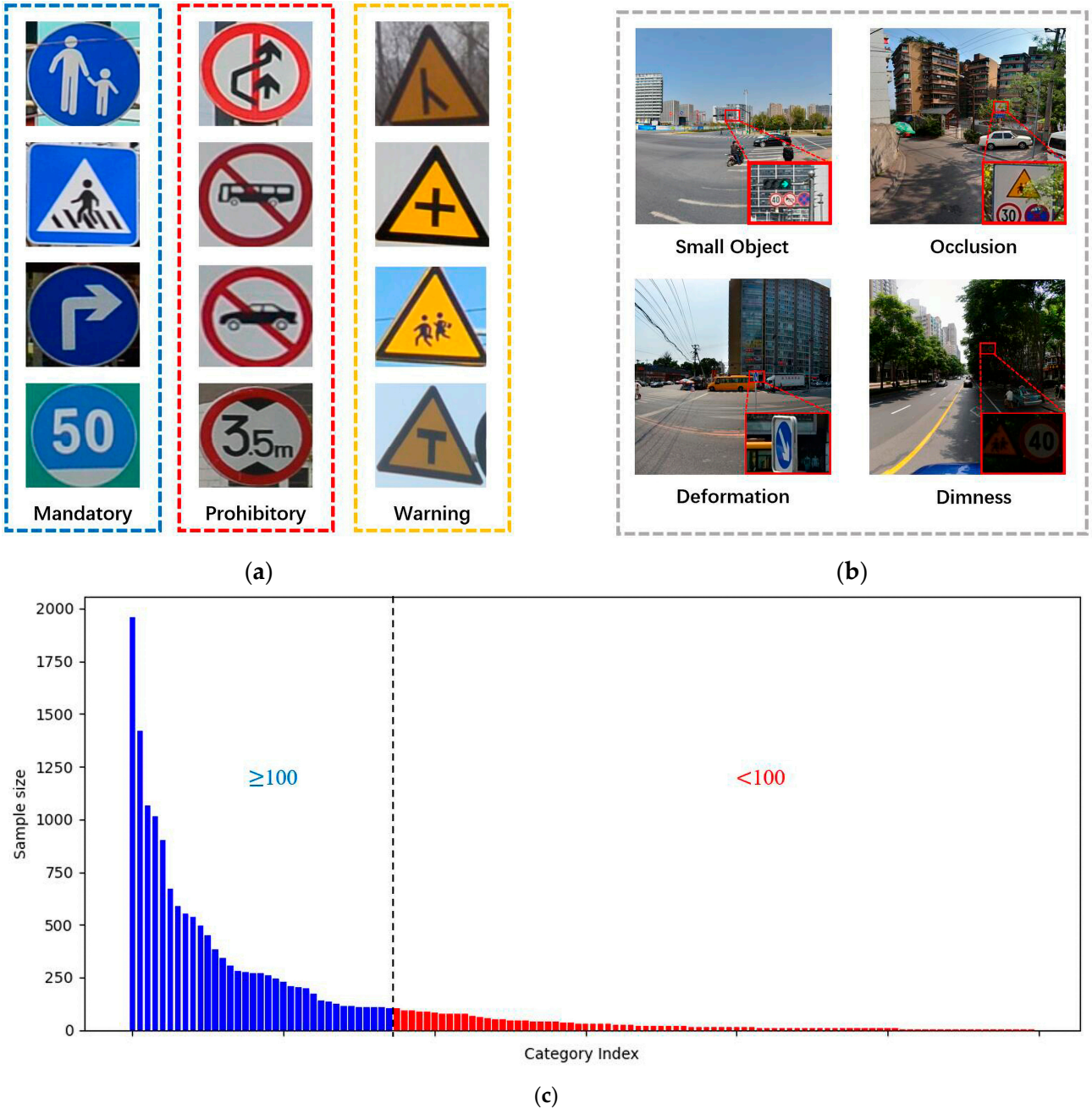


Figure 1. Analysis of traffic signs. (**a**) A common way of classifying traffic signs; (**b**) several situations that have a negative impact on the detector; (**c**) sample distribution of the TT100K dataset.

Researchers added machine learning algorithms for refinement based on digital morphology to improve the model's robustness. Traditional HOG features were combined with SVM for traffic sign detection and classification [7,8]. The researchers attempted to segment the images based on color features and then used SVM to implement classification on the segmented regions [9–11]. Since many traffic signs are circular in shape, the researchers used the circular Hough transform to detect the signs and then classified them using SVM [12,13]. Although the support vector machine improves detection and recognition accuracy, it is still heavily reliant on manual features. As a result, the improved model, while capable of more accurate and detailed classification, struggles to deal with the anomalies depicted in Figure 1b.

Object detection algorithms based on convolutional neural networks have become a better choice for academics due to the extensive use of high-performance computer

systems. However, smaller traffic signs are difficult to detect and classify accurately as the detailed features of small targets are difficult to transfer to the deeper feature maps. Some researchers have used image super-resolution algorithms to enhance the detection of small targets [14,15]. Since attention mechanisms can drive the network to focus more on channel and spatial feature acquisition, adding attention mechanisms to the model can also improve the model's ability to extract semantic features [16–20].

From different viewpoints, traffic signs produce images of varying scales, and changes in scale can also affect detector performance. Researchers attempted to incorporate deformable convolution into the network, which can adjust the perceptual field adaptively [19,21,22]. Because the backbone generates feature maps of varying sizes during layer-by-layer downsampling, some researchers have fused feature information from various scales by constructing feature pyramids in order to improve the model's extraction of multi-scale features [22–30].

Some researchers have improved model performance without changing the detector by using pre-processing techniques, such as image enhancement based on probabilistic models [31], highlighting edge features of traffic signs [32], and enhancing the hue of dark areas of images [33]. In addition, on-board cameras typically take high-resolution images for sensing the vehicle's surroundings but increase the search range for traffic signs. Therefore, an attempt has been made in some studies to construct a coarse-to-fine framework, which is used to reduce computational costs and improve model performance [34–36]. Background information is frequently ignored, but some researchers have improved model accuracy by using background detail features of neighboring signs [37,38]. In addition, spiking neural networks (SNN) are used to improve existing traffic sign detection and recognition algorithms [39–41], which can extract time-related features and have higher computational efficiency on hardware platforms.

To obtain accurate indication information, we must classify traffic signs down to the smallest category, which requires the algorithm to accurately identify up to several hundred sign categories. As shown in Figure 1c, there is a great difference in sample size between the traffic sign classes. This will result in classes with larger sample sizes having better classification accuracy, while classes with sparse samples perform poorly [42]. Existing studies usually only identify traffic signs according to three categories: prohibited, warning, and mandatory, or remove categories with a sample size of less than 100. However, each category of traffic sign is designed to convey important guidance information. Therefore, the detection and recognition of traffic signs need to be implemented in as comprehensive a range of categories as possible.

On-board cameras can continuously capture traffic signs, but most existing studies only use information from a single image. Missed detections or misclassifications due to anomalies such as occlusion and deformation are typically present in only a few frames, but incorrect detection of a single sign can also pose a serious hazard, negatively impacting the environment, infrastructure, and human life. Some researchers have attempted to improve detector performance using image sequences in previous studies [43–46], but this often necessitates an additional training process and consumes more computational resources. Meanwhile, successive detections can result in redundant results.

In general, the main contributions of this paper are summarized as follows:

(1) We propose a hierarchical classification model (HCM) based on the natural distribution characteristics of traffic signs, which is used for the classification of traffic signs in full classes. Meanwhile, the HCM significantly reduces the degree of imbalance between classes without changing the sample size.

(2) To deal with missing or misleading information caused by anomalies, this study designed a multi-frame information aggregation module (MIM) to extract the detection sequence of traffic signs, which is based on the embedding generated by the HCM. The temporal sequence of detection information can deal with the shortcomings of a single image, reducing false detections caused by anomalies.

(3) We validated our method using two open-source datasets, TT100K and ONCE. The HCM-improved YOLOv5 achieves a *mAP* of 79.0 in full classes, exceeding existing state-of-the-art methods. Experiments using ONCE show that MIM further improves the performance of the model by integrating multi-frame information.

2. Methods

The image sequence I captured by the on-board camera is used as input in this work, and we need to detect and recognize traffic signs for each frame. Specifically, we need to detect the bounding box $b^i = \{x^i, y^i, w^i, h^i\}$ of each traffic sign and recognize the specific category c^i. The central coordinates, width, and length of the bounding box are noted as x, y, w, and h, respectively.

$$\left\{ b_t^i, c_t^i \;\middle|\; I \right\},\ c_t^i \in K, t \in T, i \in \{1, 2, \cdots, N_t\} \tag{1}$$

In conclusion, our work can be summarized as Equation (1). K and T in the formula are the set of traffic sign categories and the set of time, respectively. N_t is the number of traffic signs in the image I_t. Figure 2 illustrates the overall framework of our method. While the vehicle moves, the on-board camera catches street scenes, creating a temporal sequence of photos I. The processing steps of our algorithm can be summarized as follows:

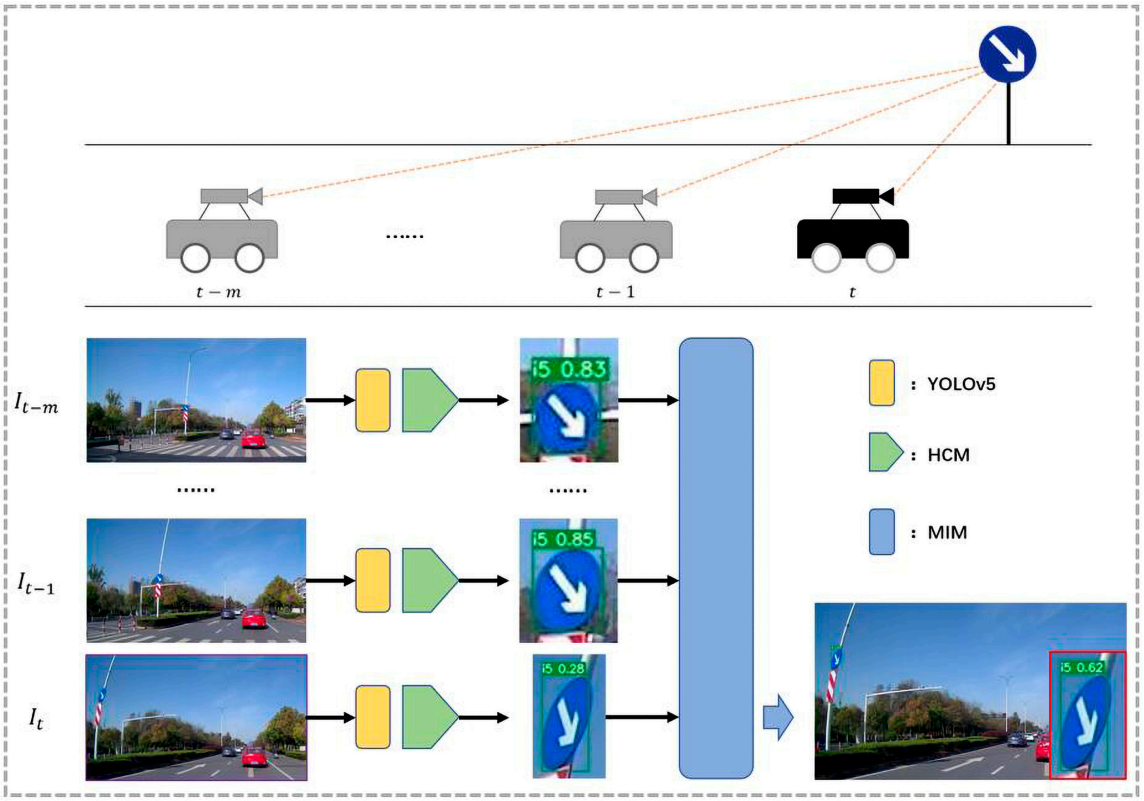


Figure 2. Model overview. Traffic signs are captured several times by the on-board camera while the vehicle is in motion. First, YOLOv5 performs the positioning of the traffic sign. Afterwards, the HCM determines the specific category. Finally, the MIM is used to integrate the information from the multiple detections to redefine the results at time t.

Step 1: We use the previous m frames $\{I_{t-m}, I_{t-m+1}, \ldots, I_{t-1}\}$ as reference frame set for a given image I_t inside I.

Step 2: Based on the image I_t and all reference frames, YOLOv5 generates a number of candidate regions through detection.

Step 3: The hierarchical classification module (HCM) implements the specific classification of candidate areas.

Step 4: Based on the embedding extracted by the HCM, the multi-frame information integration module (MIM) searches for associated boxes in reference frames.

Step 5: MIM analyzes the detection sequences generated by the association operation, which is used for category and confidence redefinition.

2.1. Detector

Object detection is used to locate targets in an image and perform category recognition. Yet, because of small objects, large-scale changes, and long-tailed distributions, it is challenging to accomplish robust detection by merely applying a generic object detection algorithm. At the same time, the algorithm's inference speed is a crucial assessment criterion in order to interpret sign information in a timely manner on rapidly moving cars.

Object detection can be divided into two categories, depending on the framework: one-stage and two-stage. Object detection algorithms with two stages generate regional proposals first, then classify each proposed region. A considerable number of regions of interest are generated via region proposals. For example, R-CNN [47] creates approximately 2000 proposed regions in each input image. Because each proposed region needs independent feature extraction and classification, two-stage object detection necessitates considerable inference space and time costs.

In contrast to the classification and regression of proposed regions, one-stage object detection typically divides the image into a number of grids, each containing a number of a priori boxes. Following that, based on the feature maps provided by the backbone network, the algorithm predicts the position and class of objects within each grid [48]. Compared to the two-stage algorithm, the one-stage object detection uses a more direct global regression and classification. The one-stage framework allows for fast inference as there are not a large number of candidate regions to be computed independently.

In addition to the speed of inference, the accuracy of the model is an important consideration. The scale fluctuations of traffic signs and small objects result in generic object detection methods that are frequently hard to recognize robustly. MS-COCO [49] defines objects with an area of fewer than 32×32 pixels as small objects. The sparse appearance of small objects makes it difficult for the algorithm to distinguish between background and object, and it also places higher challenges on the model's detection accuracy [50]. During the feature extraction process, the backbone network can generate feature maps of different sizes to represent information at various scales. Shallow feature maps contain more detailed spatial features, while deeper feature maps represent more abstract semantic features. To address the performance decrease caused by scale factors, researchers designed the feature pyramid network (FPN) for fusing feature information at multiple scales by concatenating or summing elements between feature maps.

For reasons of inference speed and detection accuracy, the detection and recognition of traffic signs require a one-stage object detection algorithm that adapts to the multi-scale variation. Following comparison, YOLOv5 is selected as the detector in this research. As a member of the yolo series of object detection algorithms, YOLOv5 not only inherits the conventional quick detection capabilities but also applies a number of tactics to mitigate the detrimental impacts of scale variation and small objects. Specifically, benefiting from a unique residual structure and spatial pyramidal pooling, YOLOv5 has excellent multi-scale feature extraction capabilities, which help to extract traffic sign features at different distances. The creation of a bi-directional feature pyramid structure improves information transfer across features at different scales and the model's retention of detailed features in the image. At the same time, traffic signs have distinct shapes and color qualities that set them out from the background, so YOLOv5 can be used to precisely locate traffic signs in an image.

In the real world, traffic signs suffer from a sample imbalance between categories, which has a direct impact on the dataset. Although object detection algorithms contain both localization and classification capabilities, the long-tailed distribution of traffic signs frequently results in significant a reduction in the algorithm's classification performance.

YOLOv5 achieves multi-classification in the head by modifying the feature map size, but the imbalanced distribution of data has a substantial impact on the model's detection and classification performance. Therefore, YOLOv5 is not suitable for both the detection and recognition of traffic signs, but traffic sign localization can be effectively solved by using a single classification that distinguishes the traffic signs from the background.

Given the input image sequence I, we will utilize YOLOv5 to locate the traffic signs in the image. For one of the images I_t, the detector will obtain N_t bounding boxes and the corresponding confidence, denoted as $D_t = \{b_t^i, conf_t^i\}$, $i \in \{1, 2, \cdots, N_t\}$. The bounding box is represented by the term b_t^i in the equation, which contains the length, width, and centroid coordinates of the box, while the other component $conf_t^i$ is used to describe the confidence of the detection, $conf_t^i \in [0, 1]$. By processing the entire image sequence with YOLOv5, we can obtain a detection sequence D.

2.2. Hierarchical Classification Model

Although YOLOv5 as a detector can effectively address the problem of localizing traffic signs in photos, robust classification is difficult to obtain. The difficulty in categorization is due to the fact that there are hundreds of kinds of traffic signs and that the distribution of samples is significantly imbalanced. In general, YOLOv5 needs a new classification module that is able to achieve accurate classification for all classes in the case of long-tailed distributions. Moreover, as there is often more than one traffic sign in an image, the classification model should avoid using complex structural designs, which can lead to long processing times for a single image.

The dataset of traffic signs generally suffers from an uneven distribution of samples, which reflects the distribution in realistic scenarios. The mandatory category of signs, which is typically used to guide lane information, has a large sample size, while the warning category of signs has an extremely limited sample size. Despite the fact that there are hundreds of traffic sign classifications, a few common categories account for the vast majority of samples in the dataset. Based on the sample size, we divided the traffic sign categories in TT100K into three categories: large, medium, and small. Categories with fewer than 10 samples were labeled as small, those with more than 50 samples were labeled as large, and the remaining categories were labeled as medium. Table 1 depicts the percentage of sample size and the percentage of number of categories for these three types of signs. The category with a large sample size accounts for only 20.6% of all categories but has 43.9% of the sample size. In contrast, the low sample size category accounts for more than half of all categories but has a sample size of only 12.2%.

Table 1. Statistical results for the three categories.

Percentage Type	Large	Medium	Small
Sample size	43.9%	43.9%	12.2%
Number of categories	20.6%	24.5%	54.8%

When faced with imbalanced sample sizes, previous research has often used either under-sampling or oversampling to balance the sample size. Both strategies are data-level approaches, but under-sampling yields less data for model training, whereas oversampling lengthens training time and may result in model overfitting [42]. Unlike data augmentation, we employ a grouping strategy to classify traffic sign categories. Traffic signs are classified into three superclasses: warning, prohibitory, and mandatory, with a number of specialized subclasses within each superclass. As illustrated in Figure 3, the superclasses range significantly in color and shape. For example, prohibitory signs have a red circular border, but warning signs have triangular and yellow features. This significant difference in features makes it easy for the classification algorithm to achieve better classification accuracy on the superclasses. The difficulty in classification is to precisely identify subclasses with a long-tailed distribution. However, we discovered that the majority of the mandatory

signs are seen in classes with a large sample size, whereas the warning signs are typically found in classes with a small sample size. While the overall sample distribution exhibits a significant long-tailed distribution, the difference in sample size between subclasses within a superclass is much smaller. Thus, grouping can improve the overall performance of the classification model by reducing the degree of imbalance without changing the sample size.

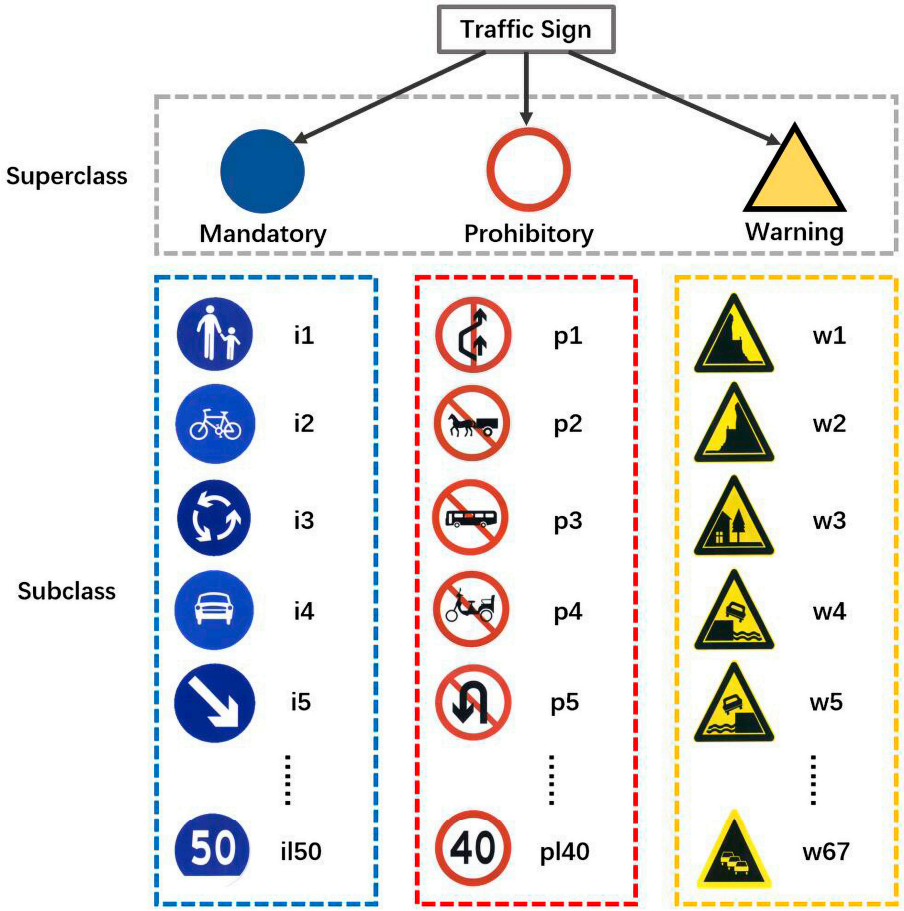


Figure 3. Three superclasses and their corresponding subclasses.

Based on the grouping results, a lightweight hierarchical classification module (HCM) was built and used to identify superclasses and subclasses. Although classifiers tailored to each superclass can be created, this typically slows down inference. Therefore, the HCM is made up of four structurally identical sub-networks, one of which serves as a superclass classifier and the other three for subclass recognition within the superclass. Each subnetwork can be divided into two parts: the backbone and the classification component (CS). The extraction of deep semantic features is required for a number of candidate regions provided by YOLOv5. We built a lightweight backbone h based on MobileNet to accomplish quick feature extraction. The input images were initially rescaled to $224 \times 224 \times 3$. Following that, we extracted features using lightweight convolutional components. To extract features, we then constantly downsample the input image based on the convolutional component. This procedure is as follows: given a candidate region b, the backbone h generates a high-dimensional embedding e through progressive downsampling, $e = h(b)$.

Following traffic sign feature extraction, we combine the convolution and softmax functions to generate a classification component g. Convolution is used for classification rather than fully connected layers because the number of parameters in fully connected layers increases dramatically as the number of classes to be classified increases, increasing space and time consumption. We first determine the convolution kernel θ depending on

the number of categories, then change the feature embedding e to a feature map of the desired size using convolution operations, and lastly use the softmax function to produce the classification result c, $c = g(e, \theta)$ or $c = g(h(b), \theta)$.

As shown in Figure 4, HCM begins prediction by identifying superclasses, following which the relevant subclass classifier is chosen for a specific classification. Simultaneously, there are model training requirements. We adjust the number of classifications by changing the convolution kernel parameter θ in the classification component g. Take the superclass classifier as an example, where the classification component generates a feature map of size 1×3 for the identification of the three superclasses. After completing the structural design of the HCM, we use a cross-entropy loss function for model training. The c_{gt} and c_{pre} in Equation (2) denote the ground truth and predicted value, respectively.

$$L_{log}(c_{gt}, c_{pre}) = -[c_{gt} \log(c_{pre}) + (1 - c_{gt}) \log(1 - c_{pre})] \quad (2)$$

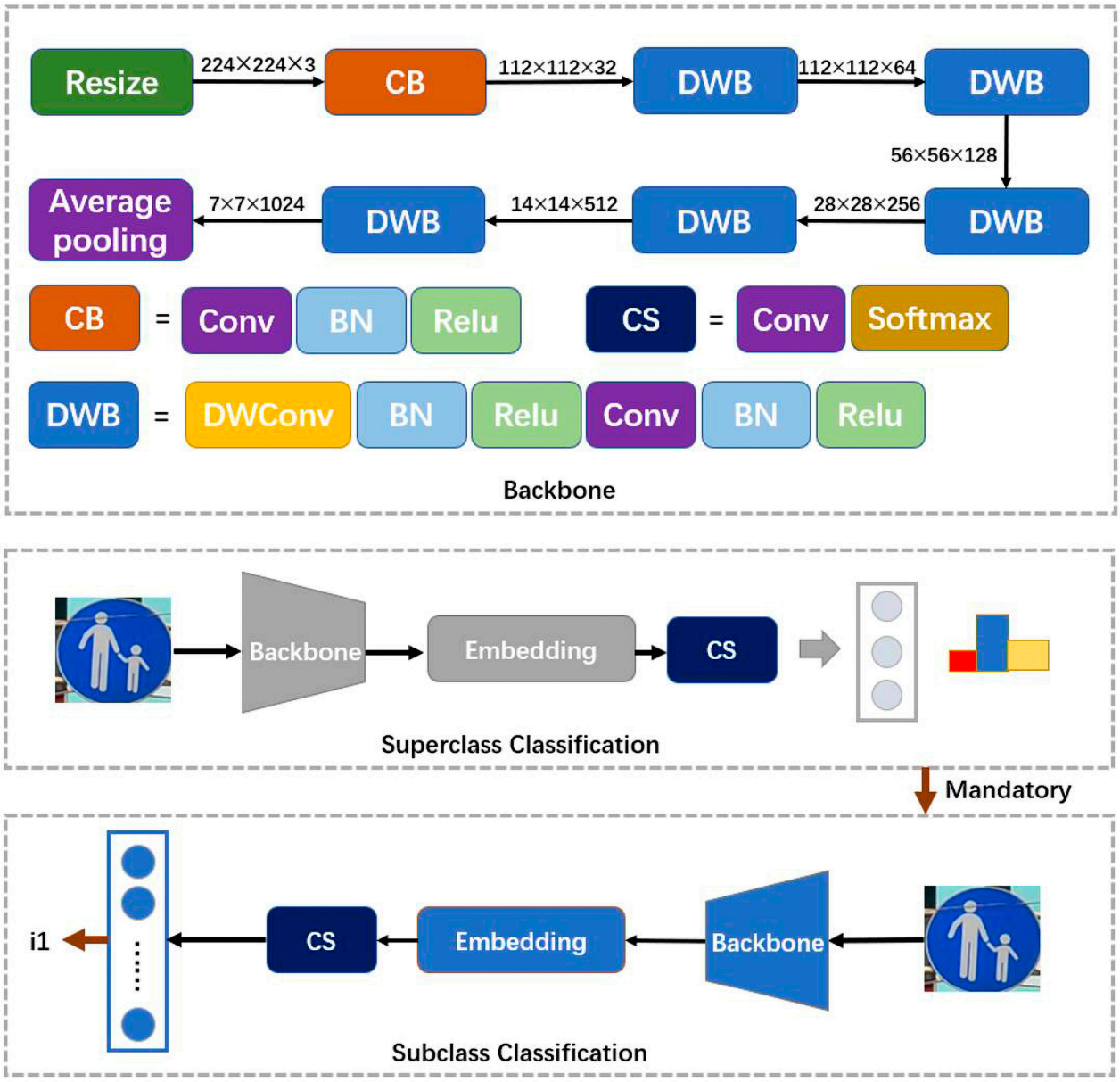


Figure 4. The structure of the HCM. The model first performs the classification of superclasses, and then selects the corresponding subclass classifiers to complete the identification of specific classes.

For one of the frames I_t, we first extract all the candidate regions from the detection result D_t. Afterwards, the HCM identifies each candidate region b_t^i independently. Following the classification results, we supplement D_t with specific categories and embeddings. After processing by HCM, D_t can be expressed by Equation (3).

$$D_t = \left\{ b_t^i, conf_t^i, c_t^i, e_t^i \right\},\ i \in \{1, 2, \cdots, N_t\} \tag{3}$$

2.3. Multi-Frame Information Integration Module

Although the combination of YOLOv5 and HCM can mitigate the detrimental effects of scale variation and long-tailed distribution, occlusion and deformation issues in real-world scenarios can still result in missed detections and misclassification. However, we discovered that traffic signs are typically captured multiple times by the in-vehicle camera, and the majority of the detection and recognition results are correct, while occlusion and deformation are only present in a few frames. To use information from multiple frames, the multi-frame information integration module (MIM) first correlates detection results between frames using the embedding e generated by the HCM and then uses the output from previous frames to improve model performance. The accumulation of detection results from multiple frames can compensate for a single frame's lack of information, reducing the number of missed detections and misclassifications. Besides that, comparing inter-frame detection outputs can be used to eliminate redundant results produced during the continuous detection state.

2.3.1. Correlating Detection Results

Before association, we need to filter the image sequence. The previous m frames $\{I_{t-m}, I_{t-m+1}, \ldots, I_{t-1}\}$ in image sequence I are utilized as a reference frame set for a given picture I_t, where the number of m is connected to the number of times the traffic sign appears in the image sequence. Following that, MIM needs to find associated detections in the reference frame, which necessitates applying high-dimensional embeddings. Although several methods for obtaining embedding have been proposed in previous works [51,52], embeddings generated using HCM can avoid additional computational resource consumption while achieving association.

The HCM generates a high-dimensional embedding of traffic signs, which is useful for distinguishing between different types of traffic signs but difficult to differentiate between instances of the same type. To that purpose, we extract the feature embedding e and the centroid coordinates of the bounding box, which are used for the traffic sign's distinguishing feature set p, $p = \{x, y, e\}$. Following that, we created a function f to determine the similarity between two traffic signs. Specifically, we first designed f_{cos}, a function for quantifying the similarity of embeddings based on cosine similarity. Equation (4) can be used to calculate the corresponding embedding similarity given two distinguishing features, p_1 and p_2.

$$f_{cos}(e_1, e_2) = \frac{e_1 \cdot e_2}{|e_1| \cdot |e_2|} \tag{4}$$

Following that, we created f_{center}, a similarity computation function based on the Euclidean distance. Due to the large range of the Euclidean distance, f_{center} needs to perform a normalizing operation, as indicated in Equation (7).

$$relu(x) = max(0, x) \tag{5}$$

$$ed(x_1, x_2, y_1, y_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{6}$$

$$f_{center}(x_1, x_2, y_1, y_2) = 1 - tanh\left(\frac{relu(ed(x_1, x_2, y_1, y_2) - \alpha)}{\beta} \right) \tag{7}$$

To enable an adjustable normalization calculation of the Euclidean distance, a hyperparameter α is set in f_{center} to reflect the centroid offset of the traffic sign instance between frames. As shown in Figure 5, the computed similarity is greatest when the Euclidean distance is less than α. We also utilize another hyperparameter β to alter the Euclidean distance range of interest. Lastly, Equation (8) depicts the weighted sum integration of similarity information from embedding and Euclidean distance.

$$\begin{gathered} f(p_1, p_2) = \omega_{cos} \times f_{cos}(e_1, e_2) + \omega_{center} \times f_{center}(x_1, x_2, y_1, y_2) \\ s.t.\ \ \omega_{cos} + \omega_{center} = 1,\ \omega_{cos} > \omega_{center} \end{gathered} \tag{8}$$

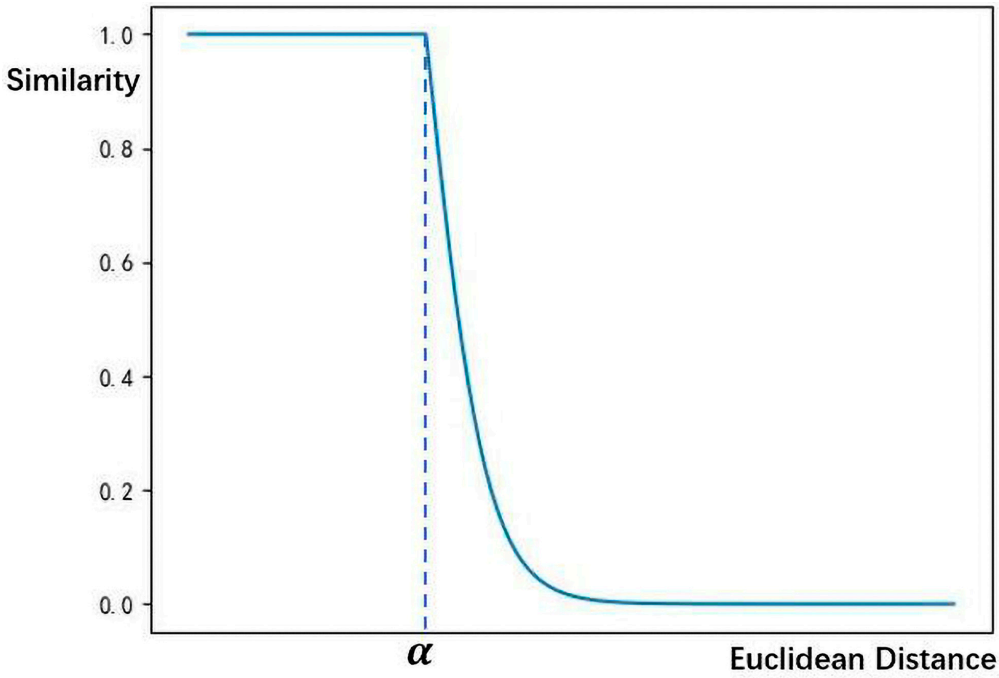


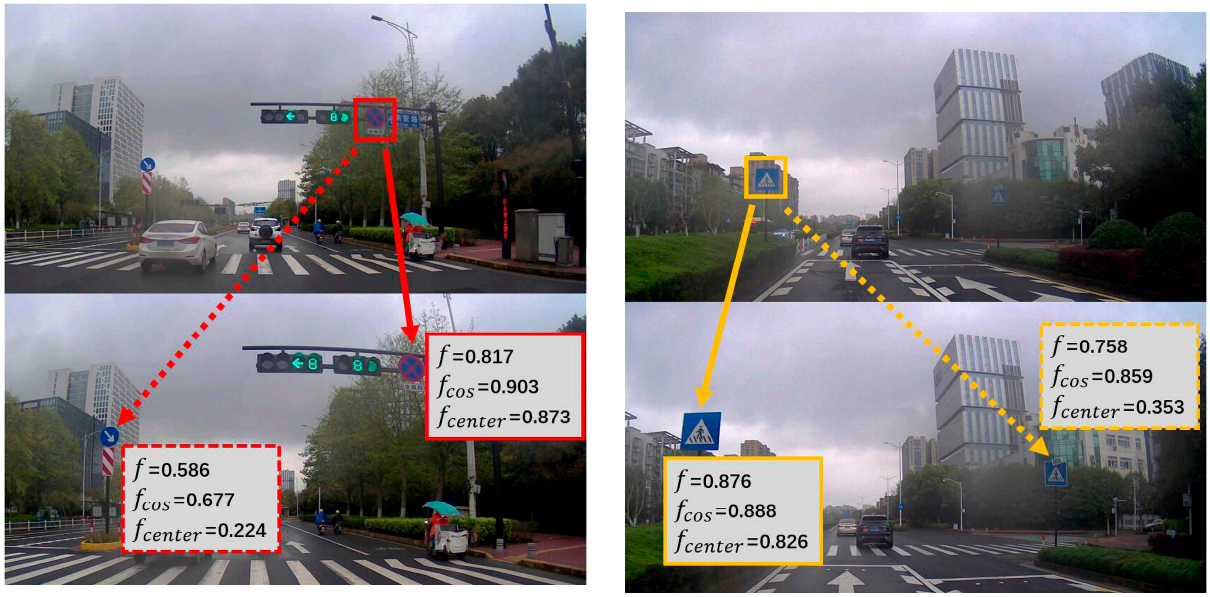
Figure 5. Controlled normalization of Euclidean distances based on f_{center}.

Since the function f achieves result association mostly through embedding, the weight parameters should be set so that ω_{cos} is greater than ω_{center}. Figure 6 shows two examples of how similarity information from both aspects can be used to distinguish among different traffic signs. For a distinguishing feature p_t^j in current frame I_t, $j \in \{1, 2, \cdots, N_t\}$, the similarity is calculated based on the function f for the results in the reference frame $I_{re}, re \in \{t-m, t-m+1, \cdots, t-1\}$. As stated in Equation (9), the procedure will produce a set of similarity scores. Afterwards, we extracted the maximum similarity scores s_{re}^{max} and the corresponding distinguishing feature p_{re}^{max} based on Equations (10) and (11), respectively.

$$\left\{ s_{re}^i \middle| s_{re}^i = f\left(p_t^j, p_{re}^i\right), i \in \{1, 2, \cdots, N_{re}\} \right\} \tag{9}$$

$$s_{re}^{max} = max\left\{ s_{re}^i \right\} \tag{10}$$

$$p_{re}^{max} = argmax\ f\left(p_t^j, p_{re}^i\right),\ s.t.\ \ i \in \{1, 2, \cdots, N_{re}\} \tag{11}$$



(**a**) (**b**)

Figure 6. Two types of association scenarios. (**a**) Distinguishing between different categories of traffic signs; (**b**) distinguishing between different instances of the same category.

If s_{re}^{max} exceeds the similarity threshold ε, the detection result corresponding to p_{re}^{max} becomes the association result in I_{re}, which is denoted as d_{re}^{j}. Similarly, we seek association results that satisfy the requirements in all reference frames. After that, the detection result d_t^j is included to generate the sequence of detection results $u^j = \left\{d_{t-m}^j, d_{t-m+1}^j, \ldots, d_{t-1}^j, d_t^j\right\}$. Lastly, for all traffic signs in the current frame I_t, we can extract the set of detection results $U_t = \left\{u^j, j \in \{1, 2, \cdots, N_t\}\right\}$.

2.3.2. Sequence Analysis

In real-world circumstances, traffic signs have anomalies such as occlusion and deformation, which can lead to false detection or misclassification by the detector. Fundamentally, the anomalies cause the image's information to be absent or deceptive. As anomalies are typically present in only a few frames, the lack of information in a single image can be compensated for by employing several detections, which can enhance the model's performance even further.

Based on the findings of the preceding analysis, MIM redefines categories and confidences based on the sequence of detection results, as illustrated in Figure 7. To count the confidence of category c_{target} in the sequence, we construct a statistical function v, denoted as Equation (12). After that, Equation (13) is used to calculate the category with the highest cumulative confidence in the sequence u^j. Finally, the category with the highest confidence becomes the redefinition category $\overline{c_t^j}$, while the redefinition confidence $\overline{conf_t^j}$ is calculated by Equation (14).

$$v\left(d, c_{target}\right) = \begin{cases} conf, & if\ c = c_{target} \\ 0, & if\ c \neq c_{target} \end{cases} \tag{12}$$

$$\overline{c_t^j} = argmax \sum_{\tau=t-m}^{t} v(d_\tau, c),\ s.t.\ c \in K \tag{13}$$

$$\overline{conf_t^j} = \frac{1}{m+1} \sum_{\tau=t-m}^{t} v\left(d_\tau, \overline{c_t^j}\right) \tag{14}$$

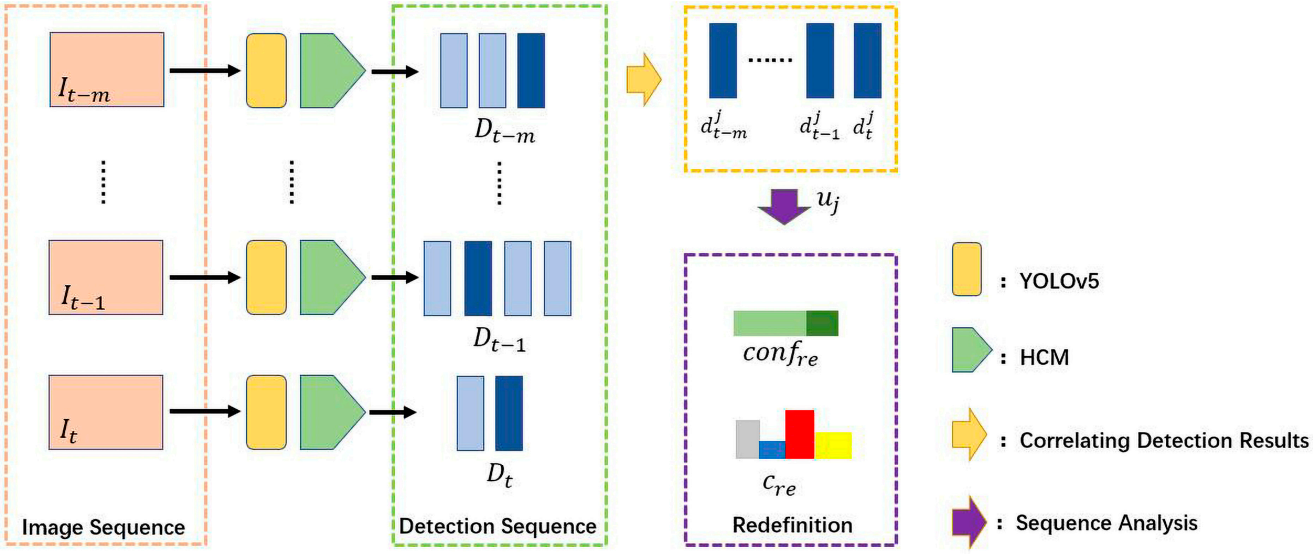


Figure 7. Correlation is used to extract traffic sign detection sequences, and the current results are then redefined by integrating information from multiple detections.

To eliminate detection results with poor confidence, a hyperparameter γ is applied. When the confidence $\overline{conf_t^j}$ is greater than γ, the categories and confidence levels in d_t^j are replaced by the redefined results. The final result can be expressed by Equation (15). Based on the same process, we redefine all detection results in current frame I_t.

$$D_t = \left\{ b_t^j, \overline{conf_t^j}, \overline{c_t^j}, e_t^j \right\},\ j \in \{1, 2, \cdots, N_t\} \tag{15}$$

Overall, by integrating information from multiple detections, sequence analysis mitigates the unfavorable impact of abnormalities. While the majority of the results in the detection sequence are correct, the redefinition results from sequence analysis can correct for a small number of classification errors. Simultaneously, deformation might occur at viewpoints near traffic signs, which usually results in low confidence in the results. When employing multiple detections, the confidence level can be enhanced by leveraging high-confidence information from previous results, resulting in fewer missed detections.

3. Results

3.1. Dataset

We deal with specific kinds of traffic signs in this study; however, a portion of the traffic sign datasets are only labeled with three categories: warning, prohibitory, and mandatory [53,54]. In contrast, TT100K [55] and ONCE [56] are better candidates because their annotation information is more detailed.

TT100K contains 10,000 images with a resolution of 2048 × 2048. At the same time, the data annotation is further refined into 232 specific categories, as shown in Figure 8. Existing studies typically remove categories with a sample size of less than 100 [19,57–60], but we use the full traffic sign category for model training and testing. On the other hand, the ONCE dataset is an autonomous driving dataset with millions of scenes. The images in the dataset were selected from 144 h of on-board camera video, taken under different lighting and weather conditions. In order to use the temporal image data in ONCE, we annotated the ONCE test set similarly to TT100K. After removing the night data, the test set contains 13,268 time-series images with a resolution of 1920 × 1020.

Figure 8. Traffic sign categories in TT100K.

3.2. Metrics

The experiment uses precision, recall, and *F*1 score as metrics to evaluate the overall performance of the detector, as indicated in Equations (16)–(18). *F*1 is the arithmetic mean of precision and recall.

$$Precision = \frac{TP}{TP + FN} \tag{16}$$

$$Recall = \frac{TP}{TP + FP} \tag{17}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{18}$$

where true positives (*TP*) are the number of samples that are actually positive and classified as positive by the classifier; false positives (*FP*) are the number of samples that are actually negative but classified as positive by the classifier; and false negatives (*FN*) are the number of samples that are actually positive but classified as negative by the classifier.

Certain evaluation criteria, such as precision, have the potential to mislead researchers [42]. When a long-tailed distribution exists, high scores may mistakenly represent good performance. As a consequence, we use *mAP* to further evaluate the model's performance. As indicated in Equation (19), *mAP* is the average of each category's mean average precision, i.e., the average precision of all categories divided by the number of categories. This paper uses a fixed intersection-over-union (IoU) value of 0.5 for computing *mAP*.

$$mAP = \frac{\sum Average\ Precision}{N(Class)} \tag{19}$$

3.3. General Detector

Although newer versions of the YOLO model have been proposed, YOLOv5 still has an advantage in the detection of traffic signs. To prove this point, the stable YOLOv7 was chosen for comparison. Based on the TT100K dataset, we calculated the overall metrics for YOLOv5, YOLOv7_l, and YOLOv7_x on the single classification detection task, as shown in Table 2.

Table 2. The overall metrics of YOLOv5 and YOLOv7 when used as a single classification detector.

Method	*Precision* (%)	*Recall* (%)	*F*1 (%)	*AP* (%)
YOLOv5	92.68	95.87	94.24	96.01
YOLOv7_l	95.67	88.71	92.06	96.52
YOLOv7_x	95.57	89.43	92.40	96.72

YOLOv7 outperformed YOLOv5 in terms of AP; however, when inferring, the detection algorithm needed to determine the final output based on a confidence threshold. Based on the same confidence threshold of 0.5, YOLOv7_l outperforms YOLOv5 in terms of precision but is lower than YOLOv5 in terms of recall and *F*1 score. Therefore, YOLOv5 is more suitable for traffic sign detection than YOLOv7.

We tested performance using SSD, Faster RCNN, CenterNet, and YOLOv5 as baselines, which are derived from diverse architectures of target detection algorithms, to indicate that generic object detection algorithms are challenging to apply to the detection and recognition of traffic signs. SSD and YOLOv5 are typical one-stage models; Faster RCNN is a two-stage algorithm; and CenterNet is well known for its unique anchor-free architecture. Since TT100K has detailed category information, all baselines are trained using the training set of TT100K until the model converges.

First, we recorded the overall metrics of the baselines on the TT100K test set in Table 3. The results show that CenterNet and YOLOv5, which have been proposed in recent years, outperformed SSD and Faster RCNN. Figure 9 compares the detection results of the baselines to further analyze the reasons for the difference in performance. The traffic signs in the red boxes are typical small objects in this case, and the signs in the yellow and blue regions show deformations due to the viewpoint. According to the results, SSD and Faster RCNN, which lack the ability to fuse multi-scale information, have a high percentage of missed detections on small objects, whereas CenterNet and YOLOv5, which use a feature pyramid structure, detect much more small-object traffic signs.

Table 3. The overall metrics of baselines evaluated on the TT100K dataset.

Method	*Precision* (%)	*Recall* (%)	*F*1 (%)
SSD	32.26	12.56	18.08
Faster RCNN	33.30	57.02	42.04
CenterNet	54.32	57.12	55.69
YOLOv5	74.37	80.93	77.51

At the same time, this example reflects the negative impact of the long-tailed distribution on the detector. Specifically, the traffic sign in the yellow area has some deformation, but the sample size of the corresponding category is sufficient. In contrast, the corresponding category in the blue region has a much smaller sample size. Although the traffic signs in the yellow and blue areas have similar deformations, the difference in sample size leads to completely different results.

Figure 9. Detection results of different baselines on TT100K dataset.

The overall metrics do not accurately represent the model's performance across categories with varied sample sizes. As a result, we first compute the average precision of the baseline across all categories. After that, we calculated *mAP* in accordance with the difference in sample size, which is shown in Table 4.

Table 4. The *mAP* of baselines evaluated on the TT100K dataset.

Method	mAP_{all}	mAP_{small}	mAP_{medium}	mAP_{large}
SSD	5.87	1.92	8.69	13.00
Faster RCNN	10.30	7.31	11.41	16.93
CenterNet	16.95	4.87	17.11	48.88
YOLOv5	26.64	1.73	36.78	80.79

The results show that there is a significant difference in accuracy between the categories with an adequate sample size and those with fewer samples. YOLOv5 has a higher *mAP* in categories with sufficient sample size. However, for the category with fewer samples, the gap between baselines is substantially lower. As a result, YOLOv5 is not suitable for combining detection and multi-classification tasks for objects having a long-tailed distribution, such as traffic signs.

The photos in TT100K are often taken in bright light, and the majority of the samples are small. ONCE, on the other hand, records photographs in a variety of weather conditions, such as sunny, rainy, and cloudy days, but with a lower proportion of small objects than TT100K. We used the same strategy to compute the metrics for the ONCE baseline method and recorded them in Table 5.

Table 5. The overall metrics of baselines evaluated on the ONCE dataset.

Method	*Precision* (%)	*Recall* (%)	*F*1 (%)
SSD	82.89	5.53	10.36
Faster RCNN	74.32	48.25	58.51
CenterNet	95.43	42.11	58.43
YOLOv5	87.69	71.23	78.61

The test results on ONCE are basically unchanged, with the exception that Faster RCNN performs substantially better than TT100K on ONCE, showing that the small object is the primary cause for Faster RCNN's limited performance. Figure 10 shows detection in three types of weather to illustrate the impact of weather on model performance. The images in the sunny environment are clear, but those in the cloudy and rainy surroundings are substantially dimmer. The effect of environmental elements is also represented in the baseline results; for example, cloudy and rainy conditions result in more missed detections or misclassifications. It was also discovered that, while Faster RCNN performed well overall, it lacked localization precision.

Figure 10. Detection results of different baselines on ONCE dataset.

Based on this, we calculated the *mAP* of the baseline for categories with varying sample sizes, as shown in Table 6. The results of the tests show that the Faster RCNN outperforms SSD and CenterNet in the less-sample category. On cloudy days, the Faster RCNN performs better as a two-stage object detection method.

Table 6. The *mAP* of baselines evaluated on the ONCE dataset.

Weather	Method	mAP_{all}	mAP_{smal}	mAP_{medium}	mAP_{large}
All	SSD	11.23	8.02	7.58	30.88
	Faster RCNN	35.15	28.85	35.71	55.22
	CenterNet	17.52	8.12	11.11	64.83
	YOLOv5	39.59	25.60	43.49	77.82
Sunny	SSD	15.51	11.68	10.30	47.27
	Faster RCNN	41.08	37.78	41.58	55.12
	CenterNet	22.61	11.38	21.04	79.16
	YOLOv5	46.01	31.91	55.42	86.67
Rainy	SSD	25.79	22.96	24.27	61.51
	Faster RCNN	44.76	37.67	60.92	74.29
	CenterNet	37.06	26.67	58.37	87.51
	YOLOv5	53.95	42.66	83.07	90.78
Cloudy	SSD	16.82	10.23	11.73	38.39
	Faster RCNN	47.99	46.44	42.3	57.58
	CenterNet	23.62	0.14	26.51	79.44
	YOLOv5	45.55	23.69	55.12	90.66

3.4. HCM-Based Method

This section evaluates the performance of the HCM-improved model to validate the efficacy of our method. The HCM-improved YOLOv5 is labeled as YOLOv5-HC. In the training process, the input image resolution was adjusted from 2048 × 2048 to 640 × 640 after downsampling. The initial learning rate of the model was set to 0.001, and Adam was used as the optimizer for the network training.

Since HCM is an image classification model, the TT100K training set must be cropped based on the annotations to obtain picture samples of traffic signs. HCM is composed of four separate sub-networks, one of which produces a feature map of size three for superclass recognition and is trained using all of the data in the training set. The remaining subnetworks are used for the three superclasses' subclass identification, and each sub-network is trained using data from the corresponding superclass. The subnetwork for mandatory sign recognition, in particular, generates a feature map of dimension 22, which corresponds to the 22 mandatory traffic sign subclasses. Similarly, to accommodate the specific number of subclasses, the sub-networks used to recognize prohibitory and warning signs will build feature maps of dimensions 119 and 32, respectively.

During the training of each sub-network, the input image was resized to 224 × 224 while Adam was set as the network's optimizer. The network training was divided into two sections. First, we freeze the backbone weights and set the learning rate at 1e-3. Following that, training with 20 epochs is used to change the weights of the convolutional layer in the classification component. The backbone weights are unfrozen in the second stage, and the learning rate is decreased to 1×10^{-4}. To finish fine-tuning the weights, the model is trained for 30 epochs.

After finishing the training, we recorded the overall metrics of YOLOv5-HC on the TT100K test set in Table 7. The results show that YOLOv5-HC outperformed YOLOv5 by 11.16%, 14.36%, and 12.64 points in precision, recall, and *F*1 score, respectively. YOLOv5-HC discovered the most complicated small object i2r in the red region shown in Figure 11. Meanwhile, only YOLOv5-HC correctly classified the few-sample category sign in the blue region.

Table 7. The overall metrics of YOLOv5 and YOLOv5-HC evaluated on the TT100K dataset.

Method	*Precision* (%)	*Recall* (%)	Inference Speed
YOLOv5	74.37	80.93	37.1 FPS
YOLOv5-HC	85.53	95.29	22.7 FPS

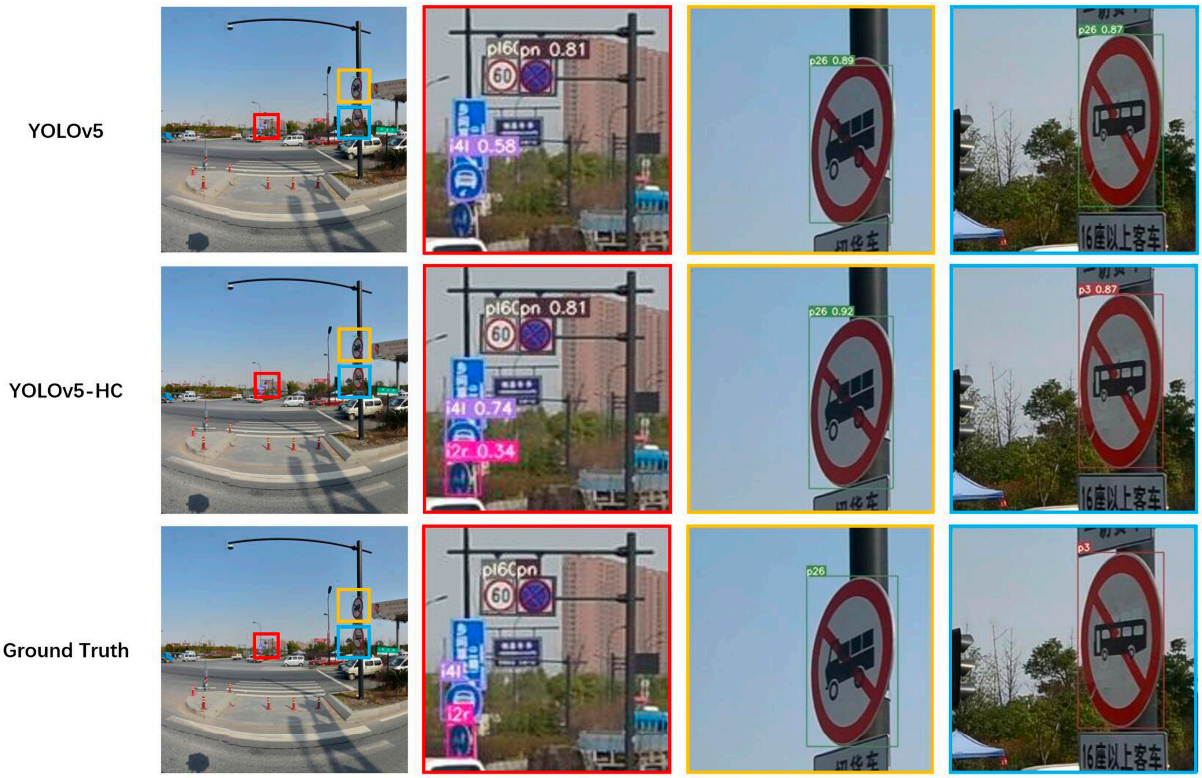


Figure 11. Detection results of YOLOv5 and YOLOv5-HC on TT100K dataset.

We computed *mAP* for YOLOv5-HC on categories with varied sample sizes in order to explicitly analyze the performance of HCM on fewer sample categories, which is recorded in Table 8. YOLOv5-HC improves accuracy across all categories. As the sample size reduces, the degree of performance improvement of HCM on YOLOv5 increases. In addition, we compared YOLOv5-HC with related methods in full classes, as shown in Table 9. Compared to the state-of-the-art model, YOLOv5-HC has a 7.1% improvement in *mAP* across all categories. Furthermore, our method has fewer model parameters.

Table 8. The *mAP* of YOLOv5 and YOLOv5-HC evaluated on the TT100K dataset.

Method	mAP_{all}	mAP_{small}	mAP_{medium}	mAP_{large}
YOLOv5	26.64	1.73	36.78	80.79
YOLOv5-HC	79.04	67.12	92.16	95.10

Table 9. Performance comparison of YOLOv5-HC with existing methods in full classes.

Method	Params	*mAP*
Cao et al. [61]	26.8M	62.3
Wang et al. [24]	8.0M	65.1
Gao et al. [62]	93.6M	71.9
YOLOv5-HC	89.5M	79.0

As traffic sign identification and recognition require rapid perception information, the model's inference speed is also an important metric. We used Nvidia 2080ti to calculate the inference speed of YOLOv5 and YOLOv5-HC, as shown in Table 7. Despite the fact that the use of HCM increased the inference time, YOLOv5-HC still achieves an inference speed of 22.7 FPS. In terms of the balance between model accuracy and inference speed, we want to improve the accuracy of the model as much as possible while satisfying the real-time condition. Taking the autonomous driving dataset ONCE as an example, the data

is sampled at a frequency of 10 FPS, so we consider that the inference speed of 22.7 FPS satisfies the real-time requirement.

3.5. MIM-Based Method

Unlike TT100K, the data in ONCE are organized temporally. While MIM is intended to be a training-free post-processing framework, we still use the model that completed the training in the previous section, but test the detector on the ONCE test set based on multiple frames of images. The MIM-improved YOLOv5-HC is labeled YOLOv5-HM.

The pre-requisite for implementing multi-image processing is to determine the number of reference frames m. Based on the temporal information of the current frame, the MIM selects m images before the current frame as a reference frame set. A temporal sequence that is too long would result in computational redundancy. On the other hand, a temporal sequence that is too short will miss some of the essential sign-detection information. By analyzing the images in ONCE, we set $m = 2$.

The MIM module first needs to correlate the detection results in the image sequence, where a number of hyperparameters need to be set. For ONCE, both α and β, which are used to adjust the mapping effect in f_{center}, are set to 500. These two parameters' values are determined by estimating the centroid offset of the same traffic sign instance in two successive frames. The similarity rating function f uses two weights to integrate similarity information. In the experiments, ω_{cos} and ω_{center} were set to 0.8 and 0.2, respectively. In addition, when the redefinition confidence calculated by the MIM was below a threshold value γ, the corresponding detection was removed. In the ONCE dataset, γ is set to 0.25.

To verify the effectiveness of the MIM, we tested the YOLOv5 and two improved versions on ONCE's test set. According to Table 10, the YOLOv5-HM showed the best performance in terms of overall metrics. Specifically, YOLOv5-HM improved by 0.67%, 1.32%, and 1.05% as compared to YOLOv5-HC in terms of precision, recall, and $F1$ score, respectively.

Table 10. The overall metrics of YOLOv5 and two improved versions evaluated on the ONCE dataset.

Method	*Precision* (%)	*Recall* (%)	*F1* (%)
YOLOv5	87.69	71.23	78.61
YOLOv5-HC	93.18	80.35	86.29
YOLOv5-HM	93.85	81.67	87.34

The on-board camera, as shown in Figure 12, takes continuous photos of the same instance as the vehicle moves. Except for the SSD with poor overall performance, the remaining detectors achieved accurate detection and recognition in the first two frames. However, the majority of the detectors exhibited a missed detection at moment t due to deformation. Although the deformed traffic signs were detected by YOLOv5-HC, the result had a low confidence level. In contrast, because YOLOv5-HM utilizes detection information from multiple frames, the higher confidence in the first two frames is used in the sequence analysis to obtain a higher confidence at moment t.

To investigate the impact of MIM further, we ran *mAP* calculations under various weather conditions and sample sizes, as shown in Table 11. According to the results, YOLOv5-HM achieves an optimal value of 73.86 for the overall *mAP*, an improvement of 1.07 over YOLOv5-HC. YOLOv5-HM outperforms the pre-improvement model in most sample size categories. YOLOv5-HM demonstrated the most substantial performance boost in sunny settings, with an overall *mAP* improvement of 1.37. YOLOv5-HM, on the other hand, demonstrated a relatively smaller performance improvement in cloudy and rainy situations.

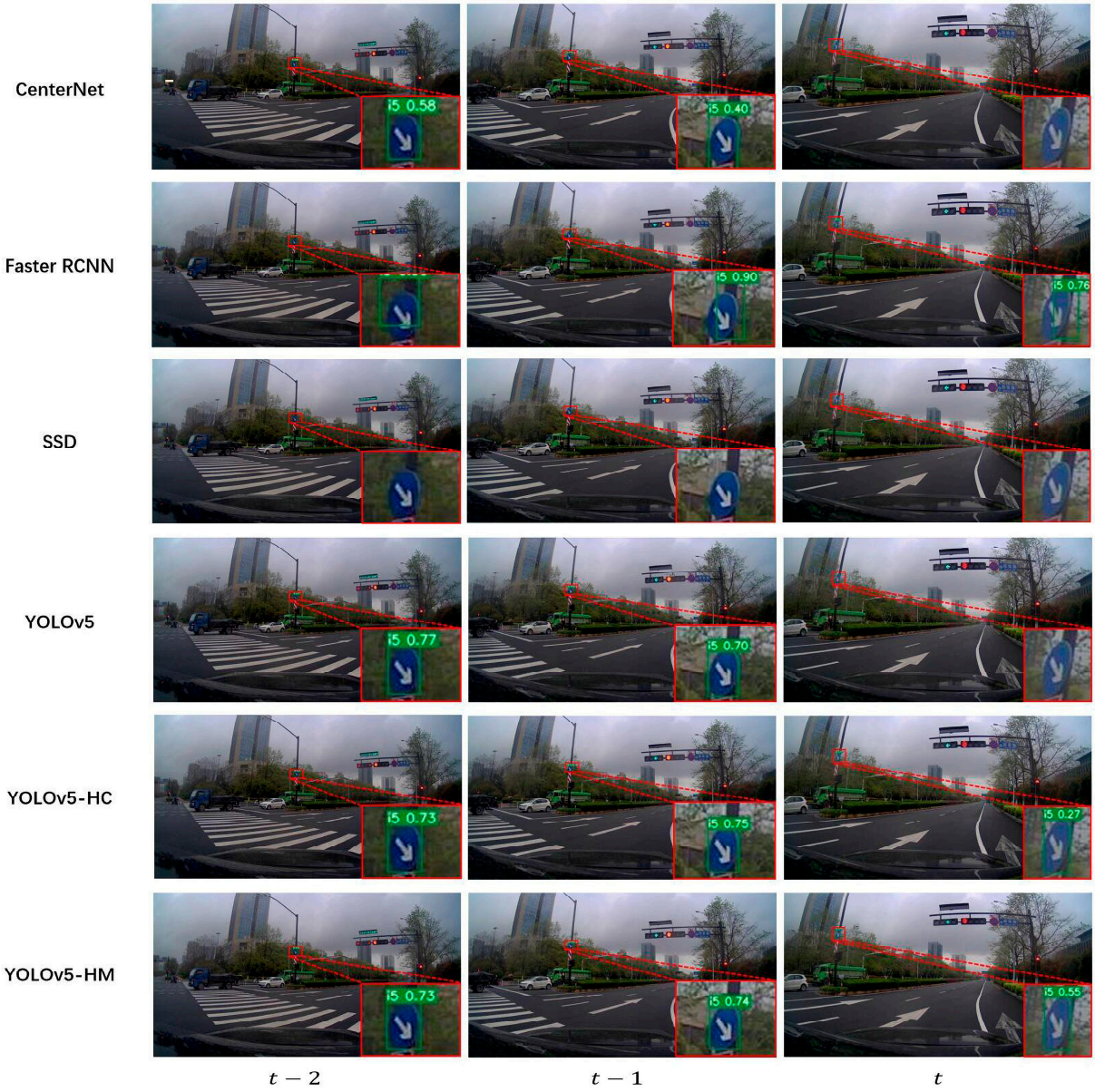


Figure 12. Temporal detection results for different detectors on the ONCE dataset.

Table 11. The *mAP* of YOLOv5 and two improved versions evaluated on the ONCE dataset.

Weather	Method	mAP_{all}	mAP_{small}	mAP_{medium}	mAP_{large}
All	YOLOv5	39.59	25.60	43.49	77.82
	YOLOv5-HC	72.79	65.53	78.66	83.40
	YOLOv5-HM	73.86	66.61	79.89	84.08
Sunny	YOLOv5	46.01	31.91	55.42	86.67
	YOLOv5-HC	83.00	80.74	84.54	89.40
	YOLOv5-HM	84.37	82.63	85.32	89.95
Rainy	YOLOv5	53.95	42.66	83.07	90.78
	YOLOv5-HC	70.67	67.12	77.40	89.51
	YOLOv5-HM	71.13	67.12	79.71	89.48
Cloudy	YOLOv5	45.55	23.69	55.12	90.66
	YOLOv5-HC	73.93	71.53	67.52	86.34
	YOLOv5-HM	73.48	70.45	68.23	86.32

In order to indicate the impact of the improvements, Figure 13 depicts examples of Yolov5 and the two upgraded models under various weather situations. In a rainy situation, YOLOv5-HC exhibits a missed detection due to deformation. After MIM processing, YOLOv5-HM achieves accurate detection and recognition, with a confidence of 0.53. A similar problem arises in sunny situations, where confidence is diminished due to deformation. In addition, the detector sometimes misclassifies. In the cloudy example, YOLOv5-HC misclassified the traffic sign as i2r. Since the classifications in the reference frames were all correct, YOLOv5-HM was able to correct this occasional misclassification using the detection information from multiple frames. In terms of inference speed, since MIM uses the feature embeddings extracted by the HCM, most of the increased computational effort comes from the calculation of the Euclidean distance in the association function, which hardly affects the inference time of the model.

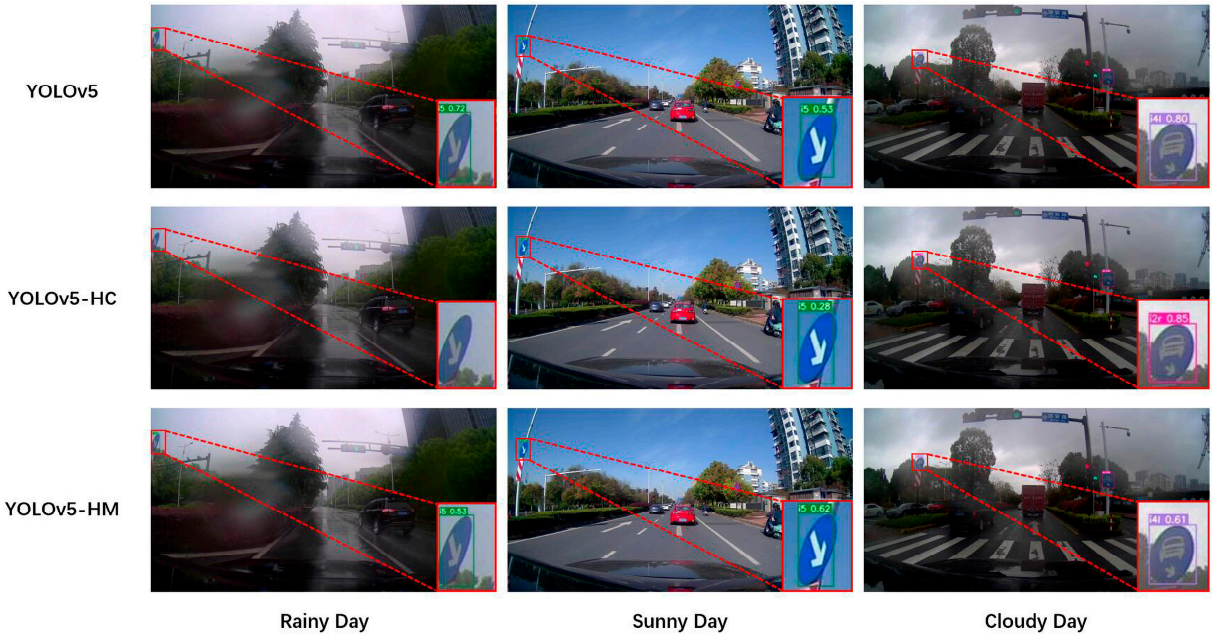


Figure 13. Detection results of YOLOv5 and two improved versions on ONCE dataset.

4. Discussion

The evaluation results based on TT100K and ONCE demonstrate that the feature pyramid structure is critical in dealing with small objects and scale variations. At the same time, the test results reveal a shortcoming in the localization accuracy of the Faster RCNN, which originates from the coarseness of the feature map and the limited information offered by the candidate boxes [48]. Furthermore, the evaluation findings on classes with varying sample sizes show that the long-tailed distribution has a considerable detrimental impact on the detector's performance. For reasons of inference speed and detection performance, YOLOv5 becomes a better choice. The use of YOLOv5 as a single classification detector allows for more efficient localization of traffic signs due to their unique color and shape features. Detecting the most challenging case in Figure 11 also demonstrates increased localization performance.

To cope with detector performance loss caused by unequal distribution, we propose a hierarchical classification model (HCM) that divides traffic signs into three superclasses and corresponding subclasses. This classification makes use of the distribution characteristics of traffic signs. Specifically, mandatory signs are mostly in the category with a large sample size, whereas warning signs are mostly in the category with a small sample size. While the overall sample distribution exhibits a significant long-tailed distribution, the difference in sample size between subclasses within a superclass is much smaller.

Equation (19) can be utilized to quantify the degree of imbalance in sample size between classes, which is calculated as the ratio of the maximum and minimum sample size across all categories [42].

$$\rho = \frac{max_i\{|C_i|\}}{min_i\{|C_i|\}} \tag{20}$$

To clearly highlight the differences after grouping, some of the categories were randomly selected from the large, medium, and small categories. Figure 14 indicates that after grouping, the sample distribution of traffic signs is more balanced, which is especially noticeable for warning traffic signs. Meanwhile, when the sample size falls, the reduction in ρ becomes bigger, resulting in more performance gains for YOLOv5-HC in the fewer sample categories. In addition, owing to its lightweight design, the HCM's hierarchical classification structure does not considerably slow down inference speed. YOLOv5-HC is similar to the two-stage object detection algorithm, but the inference speed is much faster than Faster RCNN.

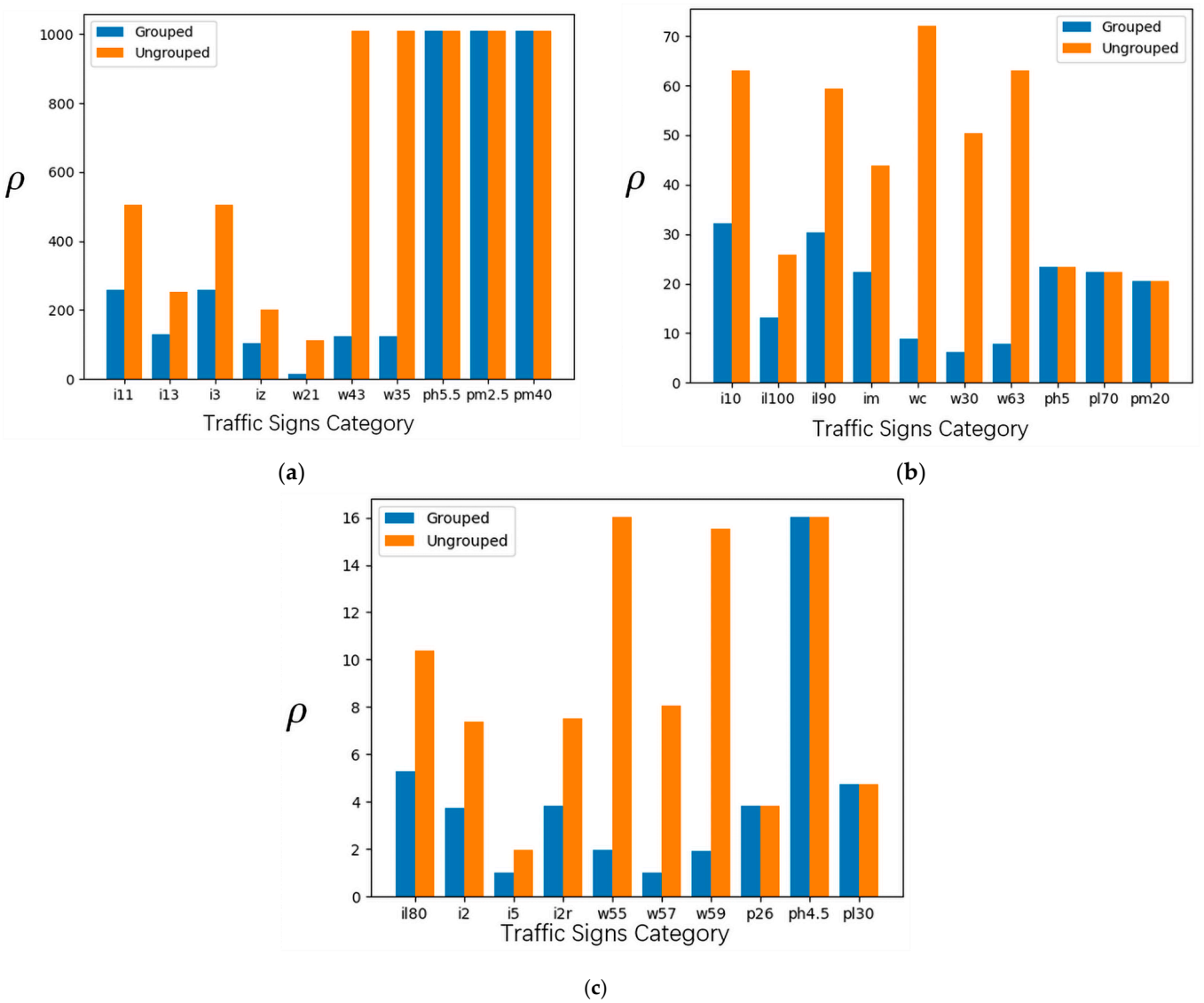


Figure 14. Change in value ρ across categories with different sample sizes after grouping. (**a**) small categories; (**b**) medium categories; (**c**) large categories.

Small objects, scale variations, and long-tailed distributions are the main causes of detector performance degradation. Furthermore, real-world scenes contain anomalies such as deformation and occlusion, which frequently result in missed detections or misclassifica-

tions. Challenging cases generated by anomalies are difficult to solve since the anomalies result in missing or misleading image information. To overcome the limitations of a single image, our proposed multi-frame information integration module (MIM) integrates data from multiple detections to achieve robust detection and recognition. Meanwhile, correlation can be utilized to eliminate the redundant results produced by successive detections.

The evaluation results on the ONCE dataset demonstrate that MIM achieves a performance improvement in most cases. To further analyze the role of MIM, we varied the range of information integration by adjusting the number of reference frames. The data in Table 12 show that as the number of reference frames increases, the performance of YOLOv5-HM gradually improves, indicating the significance of utilizing multi-frame information. However, since traffic signs in ONCE are typically photographed three times, there is no further improvement in model performance when the number of reference frames m in the experiment exceeds two.

Table 12. YOLOv5-HM's overall metrics evaluated on different reference frame numbers.

m	*Precision* (%)	*Recall* (%)	$F1$ (%)	mAP_{all}
0	93.18	80.35	86.29	72.79
1	93.43	81.05	86.80	73.36
2	93.85	81.67	87.34	73.86
3	93.85	81.67	87.34	73.86
4	93.85	81.67	87.34	73.86

5. Conclusions

In this paper, two novel and simple-to-implement modules are proposed to improve the performance of YOLOv5 for traffic sign detection and recognition. YOLOv5 provides outstanding localization performance for small objects and scale variations as a single classification detector. To reduce the negative impact of long-tailed distributions on classification, we propose a hierarchical classification module for the specific classification of traffic signs. Through grouping, HCM divides traffic signs into three superclasses and corresponding subclasses. The grouping takes advantage of traffic sign distributional characteristics, which can greatly reduce sample size discrepancies between classes. However, in the presence of anomalies such as occlusion and deformation, single-image-based algorithms still suffer from missing detection or misclassification. To deal with missing or misleading information caused by anomalies, this study designed a multi-frame information aggregation module to extract the detection sequence of traffic signs, which is based on the embedding generated by the HCM. The temporal sequence of detection information can deal with the shortcomings of a single image, reducing false detections caused by anomalies.

Experimental results based on TT100K show that YOLOv5-HC achieves a mAP of 79.0 in full classes, which exceeds state-of-the-art methods. At the same time, the inference speed of 22.7 FPS satisfies the real-time requirement. Furthermore, YOLOv5-HM using MIM outperformed YOLOv5-HC in terms of overall accuracy, with 0.67% improvement in precision, 1.32% improvement in recall, and 1.05 improvement in $F1$ score, respectively.

YOLOv5 has some shortcomings in traffic sign detection and consumes most of the computational resources. Therefore, we will improve the existing detection module in our research as more advanced object detection algorithms are proposed. Meanwhile, we will also try to improve the inference speed of the existing model using SNN, which can better balance the inference time and model accuracy. In addition, due to the uniqueness of the colors as well as the structure of the traffic signs, we also consider the use of VAE or statistical models to generate distributions that can be used for traffic sign recognition.

Author Contributions: Conceptualization, Y.H. and J.X.; methodology, Y.H. and J.X.; software, J.X.; supervision, Y.H.; validation, D.Y.; visualization, J.X. and D.Y.; writing—original draft, J.X.; writing—review and editing, Y.H. and J.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Wuhan University–Huawei Geoinformatics Innovation Laboratory.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Rehman, Y.; Khan, J.A.; Shin, H. Efficient coarser-to-fine holistic traffic sign detection for occlusion handling. *IET Image Process.* **2018**, *12*, 2229–2237. [CrossRef]
2. Xu, X.; Jin, J.; Zhang, S.; Zhang, L.; Pu, S.; Chen, Z. Smart data driven traffic sign detection method based on adaptive color threshold and shape symmetry. *Future Gener. Comput. Syst.-Int. J. Escience* **2019**, *94*, 381–391. [CrossRef]
3. Yang, Y.; Luo, H.; Xu, H.; Wu, F. Towards Real-Time Traffic Sign Detection and Classification. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2022–2031. [CrossRef]
4. Cao, J.; Song, C.; Peng, S.; Xiao, F.; Song, S. Improved Traffic Sign Detection and Recognition Algorithm for Intelligent Vehicles. *Sensors* **2019**, *19*, 4021. [CrossRef]
5. Guo, S.; Yang, X. Fast recognition algorithm for static traffic sign information. *Open Phys.* **2018**, *16*, 1149–1156. [CrossRef]
6. Yin, S.; Ouyang, P.; Liu, L.; Guo, Y.; Wei, S. Fast Traffic Sign Recognition with a Rotation Invariant Binary Pattern Based Feature. *Sensors* **2015**, *15*, 2161–2180. [CrossRef] [PubMed]
7. Hechri, A.; Mtibaa, A. Two-stage traffic sign detection and recognition based on SVM and convolutional neural networks. *Iet Image Process.* **2020**, *14*, 939–946. [CrossRef]
8. Bouti, A.; Mahraz, M.A.; Riffi, J.; Tairi, H. A robust system for road sign detection and classification using LeNet architecture based on convolutional neural network. *Soft Comput.* **2020**, *24*, 6721–6733. [CrossRef]
9. Madani, A.; Yusof, R. Traffic sign recognition based on color, shape, and pictogram classification using support vector machines. *Neural Comput. Appl.* **2018**, *30*, 2807–2817. [CrossRef]
10. Lillo-Castellano, J.M.; Mora-Jimenez, I.; Figuera-Pozuelo, C.; Rojo-Alvarez, J.L. Traffic sign segmentation and classification using statistical learning methods. *Neurocomputing* **2015**, *153*, 286–299. [CrossRef]
11. Li, H.; Sun, F.; Liu, L.; Wang, L. A novel traffic sign detection method via color segmentation and robust shape matching. *Neurocomputing* **2015**, *169*, 77–88. [CrossRef]
12. Saadna, Y.; Behloul, A.; Mezzoudj, S. Speed limit sign detection and recognition system using SVM and MNIST datasets. *Neural Comput. Appl.* **2019**, *31*, 5005–5015. [CrossRef]
13. Berkaya, S.K.; Gunduz, H.; Ozsen, O.; Akinlar, C.; Gunal, S. On circular traffic sign detection and recognition. *Expert Syst. Appl.* **2016**, *48*, 67–75. [CrossRef]
14. Yu, Y.; Jiang, T.; Li, Y.; Guan, H.; Li, D.; Chen, L.; Yu, C.; Gao, L.; Gao, S.; Li, J. SignHRNet: Street-level traffic signs recognition with an attentive semi-anchoring guided high-resolution network. *Isprs J. Photogramm. Remote Sens.* **2022**, *192*, 142–160. [CrossRef]
15. Wang, Z.-Z.; Xie, K.; Zhang, X.-Y.; Chen, H.-Q.; Wen, C.; He, J.-B. Small-Object Detection Based on YOLO and Dense Block via Image Super-Resolution. *IEEE Access* **2021**, *9*, 56416–56429. [CrossRef]
16. Li, Y.; Li, J.; Meng, P. Attention-YOLOV4: A real-time and high-accurate traffic sign detection algorithm. *Multimed. Tools Appl.* **2023**, *82*, 7567–7582. [CrossRef]
17. Wei, H.; Zhang, Q.; Qian, Y.; Xu, Z.; Han, J. MTSDet: Multi-scale traffic sign detection with attention and path aggregation. *Appl. Intell.* **2023**, *53*, 238–250. [CrossRef]
18. Wang, X.; Guo, J.; Yi, J.; Song, Y.; Xu, J.; Yan, W.; Fu, X. Real-Time and Efficient Multi-Scale Traffic Sign Detection Method for Driverless Cars. *Sensors* **2022**, *22*, 6930. [CrossRef]
19. Hu, J.; Wang, Z.; Chang, M.; Xie, L.; Xu, W.; Chen, N. PSG-Yolov5: A Paradigm for Traffic Sign Detection and Recognition Algorithm Based on Deep Learning. *Symmetry* **2022**, *14*, 2262. [CrossRef]
20. Triki, N.; Karray, M.; Ksantini, M. A Real-Time Traffic Sign Recognition Method Using a New Attention-Based Deep Convolutional Neural Network for Smart Vehicles. *Appl. Sci.* **2023**, *13*, 4793. [CrossRef]
21. Gao, X.; Chen, L.; Wang, K.; Xiong, X.; Wang, H.; Li, Y. Improved Traffic Sign Detection Algorithm Based on Faster R-CNN. *Appl. Sci.* **2022**, *12*, 8948. [CrossRef]
22. Liu, Z.; Shen, C.; Fan, X.; Zeng, G.; Zhao, X. Scale-aware limited deformable convolutional neural networks for traffic sign detection and classification. *IET Intell. Transp. Syst.* **2020**, *14*, 1712–1722. [CrossRef]
23. Zhang, Y.; Lu, Y.; Zhu, W.; Wei, X.; Wei, Z. Traffic sign detection based on multi-scale feature extraction and cascade feature fusion. *J. Supercomput.* **2023**, *79*, 2137–2152. [CrossRef]
24. Wang, J.; Chen, Y.; Dong, Z.; Gao, M. Improved YOLOv5 network for real-time multi-scale traffic sign detection. *Neural Comput. Appl.* **2022**, *35*, 7853–7865. [CrossRef]
25. Wu, J.; Liao, S. Traffic Sign Detection Based on SSD Combined with Receptive Field Module and Path Aggregation Network. *Comput. Intell. Neurosci.* **2022**, *2022*, 4285436. [CrossRef] [PubMed]

26. Yao, Y.; Han, L.; Du, C.; Xu, X.; Jiang, X. Traffic sign detection algorithm based on improved YOLOv4-Tiny. *Signal Process.-Image Commun.* **2022**, *107*, 116783. [CrossRef]
27. Liu, Y.; Peng, J.; Xue, J.-H.; Chen, Y.; Fu, Z.-H. TSingNet: Scale-aware and context-rich feature learning for traffic sign detection and recognition in the wild. *Neurocomputing* **2021**, *447*, 10–22. [CrossRef]
28. Liang, Z.; Shao, J.; Zhang, D.; Gao, L. Traffic sign detection and recognition based on pyramidal convolutional networks. *Neural Comput. Appl.* **2020**, *32*, 6533–6543. [CrossRef]
29. Yuan, Y.; Xiong, Z.; Wang, Q. VSSA-NET: Vertical Spatial Sequence Attention Network for Traffic Sign Detection. *IEEE Trans. Image Process.* **2019**, *28*, 3423–3434. [CrossRef]
30. Ou, Z.; Xiao, F.; Xiong, B.; Shi, S.; Song, M. FAMN: Feature Aggregation Multipath Network for Small Traffic Sign Detection. *IEEE Access* **2019**, *7*, 178798–178810. [CrossRef]
31. Suto, J. An Improved Image Enhancement Method for Traffic Sign Detection. *Electronics* **2022**, *11*, 871. [CrossRef]
32. Khan, J.A.; Chen, Y.; Rehman, Y.; Shin, H. Performance enhancement techniques for traffic sign recognition using a deep neural network. *Multimed. Tools Appl.* **2020**, *79*, 20545–20560. [CrossRef]
33. Khan, J.A.; Yeo, D.; Shin, H. New Dark Area Sensitive Tone Mapping for Deep Learning Based Traffic Sign Recognition. *Sensors* **2018**, *18*, 3776. [CrossRef] [PubMed]
34. Wang, Z.; Wang, J.; Li, Y.; Wang, S. Traffic Sign Recognition with Lightweight Two-Stage Model in Complex Scenes. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 1121–1131. [CrossRef]
35. Liu, L.; Wang, Y.; Li, K.; Li, J. Focus First: Coarse-to-Fine Traffic Sign Detection with Stepwise Learning. *IEEE Access* **2020**, *8*, 171170–171183. [CrossRef]
36. Song, Y.; Fan, R.; Huang, S.; Zhu, Z.; Tong, R. A three-stage real-time detector for traffic signs in large panoramas. *Comput. Vis. Media* **2019**, *5*, 403–416. [CrossRef]
37. Min, W.; Liu, R.; He, D.; Han, Q.; Wei, Q.; Wang, Q. Traffic Sign Recognition Based on Semantic Scene Understanding and Structural Traffic Sign Location. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 15794–15807. [CrossRef]
38. Tian, Y.; Gelernter, J.; Wang, X.; Li, J.; Yu, Y. Traffic Sign Detection Using a Multi-Scale Recurrent Attention Network. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 4466–4475. [CrossRef]
39. Rasteh, A.; Delpech, F.; Aguilar-Melchor, C.; Zimmer, R.; Shouraki, S.B.; Masquelier, T. Encrypted internet traffic classification using a supervised spiking neural network. *Neurocomputing* **2022**, *503*, 272–282. [CrossRef]
40. Zhang, Y.; Xu, H.; Huang, L.; Chen, C. A storage-efficient SNN-CNN hybrid network with RRAM-implemented weights for traffic signs recognition. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106232. [CrossRef]
41. Xie, K.; Zhang, Z.; Li, B.; Kang, J.; Niyato, D.; Xie, S.; Wu, Y. Efficient Federated Learning with Spike Neural Networks for Traffic Sign Recognition. *IEEE Trans. Veh. Technol.* **2022**, *71*, 9980–9992. [CrossRef]
42. Johnson, J.M.; Khoshgoftaar, T.M. Survey on deep learning with class imbalance. *J. Big Data* **2019**, *6*, 27. [CrossRef]
43. Yu, J.; Ye, X.; Tu, Q. Traffic Sign Detection and Recognition in Multiimages Using a Fusion Model with YOLO and VGG Network. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 16632–16642. [CrossRef]
44. Atif, M.; Zoppi, T.; Gharib, M.; Bondavalli, A. Towards Enhancing Traffic Sign Recognition through Sliding Windows. *Sensors* **2022**, *22*, 2683. [CrossRef] [PubMed]
45. Zhang, Y.; Wang, Z.; Song, R.; Yan, C.; Qi, Y. Detection-by-tracking of traffic signs in videos. *Appl. Intell.* **2022**, *52*, 8226–8242. [CrossRef]
46. Song, S.; Li, Y.; Huang, Q.; Li, G. A New Real-Time Detection and Tracking Method in Videos for Small Target Traffic Signs. *Appl. Sci.* **2021**, *11*, 3061. [CrossRef]
47. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J.; IEEE. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
48. Zhao, Z.-Q.; Zheng, P.; Xu, S.-T.; Wu, X. Object detection with deep learning: A review. *IEEE Trans. Neural Netw.* **2019**, *30*, 3212–3232. [CrossRef]
49. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollar, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the 13th European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
50. Tong, K.; Wu, Y.; Zhou, F. Recent advances in small object detection based on deep learning: A review. *Image Vis. Comput.* **2020**, *97*, 103910. [CrossRef]
51. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015.
52. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R.; IEEE. Mask R-CNN. In Proceedings of the 16th IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
53. Houben, S.; Stallkamp, J.; Salmen, J.; Schlipsing, M.; Igel, C.; IEEE. Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 4–9 August 2013.

54. Zhang, J.; Zou, X.; Kuang, L.-D.; Wang, J.; Sherratt, R.S.; Yu, X. CCTSDB 2021: A More Comprehensive Traffic Sign Detection Benchmark. *Hum.-Cent. Comput. Inf. Sci.* **2022**, *12*, 23. [CrossRef]
55. Zhu, Z.; Liang, D.; Zhang, S.; Huang, X.; Li, B.; Hu, S.; IEEE. Traffic-Sign Detection and Classification in the Wild. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 27–30 June 2016; pp. 2110–2118.
56. Mao, J.; Niu, M.; Jiang, C.; Liang, H.; Chen, J.; Liang, X.; Li, Y.; Ye, C.; Zhang, W.; Li, Z.; et al. One million scenes for autonomous driving: ONCE dataset. *arXiv* **2021**, arXiv:2106.11037.
57. Chu, J.; Zhang, C.; Yan, M.; Zhang, H.; Ge, T. TRD-YOLO: A Real-Time, High-Performance Small Traffic Sign Detection Algorithm. *Sensors* **2023**, *23*, 3871. [CrossRef] [PubMed]
58. Sharma, V.; Dhiman, P.; Rout, R.K. Improved traffic sign recognition algorithm based on YOLOv4-tiny. *J. Vis. Commun. Image Represent.* **2023**, *91*, 103774. [CrossRef]
59. Wang, L.; Wang, L.; Zhu, Y.; Chu, A.; Wang, G. CDFF: A fast and highly accurate method for recognizing traffic signs. *Neural Comput. Appl.* **2023**, *35*, 643–662. [CrossRef]
60. Yuan, X.; Kuerban, A.; Chen, Y.; Lin, W. Faster Light Detection Algorithm of Traffic Signs Based on YOLOv5s-A2. *IEEE Access* **2023**, *11*, 19395–19404. [CrossRef]
61. Cao, J.; Zhang, J.; Jin, X. A Traffic-Sign Detection Algorithm Based on Improved Sparse R-CNN. *IEEE Access* **2021**, *9*, 122774–122788. [CrossRef]
62. Gao, E.; Huang, W.; Shi, J.; Wang, X.; Zheng, J.; Du, G.; Tao, Y. Long-Tailed Traffic Sign Detection Using Attentive Fusion and Hierarchical Group Softmax. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 24105–24115. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.