Journal of Network Intelligence ©2026 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **11**, Number **1**, February **2026**

An Adaptive Ensemble YOLOX Model Traffic Sign Detection and Recognition

Rong Hu[1]

[1]Fujian Provincial Key laboratory of big data mining and applications
Fujian University of Technology, Fuzhou 350118, China
896410521@qq.com

Ming Zhang[1]

[1]Fujian Provincial Key Laboratory of Big Data Mining and Applications
Fujian University of Technology, Fuzhou 350118, China
772689671@qq.com

Dequ Chen[1]

[1]Fujian Provincial Key laboratory of big data mining and applications
Fujian University of Technology, Fuzhou 350118, China
dequchen@foxmail.com

Yong Xu[1,*]

[1]Fujian Provincial Key laboratory of big data mining and applications
Fujian University of Technology, Fuzhou 350118, China
xuycn@foxmail.com

*Corresponding author: Yong Xu

Received January 16, 2024, revised June 20, 2024, accepted September 8, 2024.

Abstract. *With the rise of autonomous driving, traffic sign detection and recognition, as its key task, causes great attention and is also full of challenges. The complexity of traffic signs themselves, and the long tail distribution in the real world, make it difficult to improve the accuracy of detection and recognition. In this paper, we present roulette-based re-sampling to highlight particularly difficult categories, trying to eliminate the adverse effects of data distribution. At the same time, an adaptive ensemble YOLOX model is proposed. By learning the category characteristics of different difficulty, heads with different difficulty are integrated, and An adaptive class box fusion module is proposed to fuse specific detection heads, so that the accuracy of the model improves while the efficiency is almost not affected. Through a large number of experiments, we prove that the detection progress of both TT100K and Comprehensive TT100K datasets is improved.*

Keywords: Traffic sign detection and recognition, YOLOX, Adaptive copy-paste, Image augmentation, Ensemble learning

1. **Introduction.** With the ongoing rise in the number of cars, advanced driver assistance systems (ADAS) have garnered considerable attention due to their potential to enhance traffic safety. As one of the key technologies for s ADAS, traffic sign detection and recognition (TSDR) can effectively obtain traffic sign information such as warnings, instructions, and prohibitions in the road, which can effectively improve driving safety and ensure road safety at the same time. However, due to the small size of traffic signs,

partial occlusion, angle variations, fading aging, illumination changes, weather and other complex environmental disturbances, the task is especially challenging.

Traditional TSDR task [1], [2], [3], [4] take advantage of the specific shapes and striking color differences of traffic signs to manually extract features, but these features are not migratory. In recent years, due to the rapid development of deep learning,convolutional neural networks (CNN), many object detection models [5], [6], [7], [8], [9], [10], [11], [12], [13] have been proposed and good results have been achieved. However, these detection models struggle to guarantee a good compromise between efficiency and accuracy. In general, the current TSDR technology suffers from the following difficulties: First, real-world traffic signs usually show a long-tailed distribution (i. e., data in most categories is scarce while other categories are abundant), making it difficult for models to fully learn the feature representation of the tail category. Second, the TSDR task relies heavily on large amounts of annotated data that are often difficult to obtain. In order to study on as sufficient data as possible, most of the current studies focus on four categories in the German Traffic Sign Detection Benchmark (GTSDB) dataset [14] and 45 categories in the Tsinghua-Tencent 100k dataset (TT100k) [15], which is difficult to fully cover the existing traffic signs. Third, most of the traffic signs are more complex and relatively small. For example, many traffic signs account for less than 5% of the whole image's pixel. These would further lead to the learning of a full set of traffic signs more challenging.

To alleviate the above problems, effective methods are proposed in this paper. To alleviate the long-tail distribution present in the dataset, we propose adaptive copy-paste (ACP) method for the data augmentation of tail categories. Compared to other sample augmentation methods that struggle to balance effectiveness and accuracy [16], [17], [18], [19], our ACP method provides a complete data augmentation process applicable to the traffic sign domain and can be transferred to other traffic sign datasets. In order to address the problem of detecting difficult traffic signs, we propose a novel roulette-based re-sampling (RRS) module. Unlike prior methods including re-sampling [20], [21] and loss re-weighting [22], [23], the RRS module can generate new training datasets adaptively and dynamically, focusing on difficult categories during model training. Previous work [24], [25], [26] has shown the effectiveness of ensemble learning in the field of object detection, but it also reduces the efficiency of detection. In order to learn more categories, we propose a simple and effective ensemble model based on YOLOX [12], called adaptive ensemble YOLOX (AE-YOLOX), which introduces a double-level head and adaptive class box fusion (ACBF) module to achieve a better compromise between efficiency and accuracy. Extensive experiments on the TT100K dataset and our constructed Comprehensive TT100K dataset show that, compared to state-of-the-art approaches, the proposed method greatly improves recognition accuracy and achieves better performance on both datasets. To sum up, the contributions of this paper are listed as follows:

1. A large-scale traffic sign dataset, named Comprehensive TT100K (CTT100K), is created. As far as we know, the CTT100K is the largest Chinese traffic sign dataset of its kind;
2. An effective ACP method and the RRS module are proposed to alleviate the long-tailed data and emphasis on data of difficult categories during model training, respectively;
3. The AE-YOLOX model is proposed, which combines different levels of category groups during model training, and assembles the corresponding level-aware head into a double-level head. Then the ACBF module is use to improve the ensemble efficiency. In this way, the performance of detecting difficult traffic signs is improved.

2. **Related work.**

2.1. **Long-Tailed Detection and Recognition.** With the emergence of large-scale datasets and their own long-tail distribution problems, many studies have explored how to alleviate the long-tail distribution, mainly including data re-sampling, loss re-weighting and other methods. Data re-sampling iteratively gathers samples from the original dataset, resulting in a more comprehensive dataset. Loss re-weighting assigns varying weights to individual training samples based on their significance or difficulty, thereby adjusting their impact on the overall loss function. Ando et al. [20] proposed an over-sampling scheme in the deep representation space to avoid computationally intensive analyses. Chawla et al. [21] used a more advanced sampling method that augments artificial examples created by interpolating neighboring data points. Oversampling methods can effectively improve the accuracy of the tail categories, but they can bring overfitting problems and affect the generalization of the model. Lin et al. [22] proposed a focal loss method, which makes the model pay more attention to negative samples, but is sensitive to label assigning. Chen et al. [23] proposed the GradTail method that uses gradients to improve model performance dynamically in the face of long-tailed training data distributions. Through theoretical derivation and a large number of experiments, Yang et al. [27] found that when the number of tail category's instances is very small, data re-sampling and loss re-weighting methods cannot improve performance. Therefore, they proposed a semi-supervised and self-supervised strategy using inherently biased labels to improve learning, but this method is difficult to meet real-time requirement. Recently, Gao et al. [28] proposed a hierarchical group softmax head which improves the performance of the tail classes, but this method requires the construction of the appropriate label tree, and it is difficult to transferred to other datasets.

Different from the previous related works, we propose an ACP method and the RRS module. For the tail categories, we first use ACP method to generate new instances. After obtaining a relatively uniform dataset, we then employ the RRS module dynamically during model training to emphasize on data of the difficult categories. The whole process does not require additional weight parameter optimization.

2.2. **Traffic Sign Detection and Recognition.** Most of the early traffic sign detection models use color, texture and geometric features as input to detect traffic signs based on prior knowledge. Khan et al. [1] detected traffic signs by using color clues and geometric features. Meuter et al. [2] proposed a decision fusion and reasoning module by using track-based Bayesian fusion and decision tree. Greenhalgh et al. [3] used histogram of oriented gradient (HOG) features and a cascade of linear support vector machine (SVM) classifiers to recognize traffic signs. Then, Yang et al. [4] combines CNN with the maximally stable extremal regions (MSERs) detector and color probability model to extract traffic sign features and detect and classify them. Zhang et al. [29] introduced a motion classification and recognition algorithm that combines linear decision-making with SVM techniques, resulting in favorable classification outcomes. Additionally, Zhang et al. [30] proposed a visual information extraction method based on wavelet transform and feature comparison. This method aims to emulate the multi-channel spatial frequency decomposition function, facilitating rapid extraction of significant image distributions and identification of regions of interest (ROI). However, these hand-craft features is difficult to distinguish more similar traffic signs and cannot migrate to other traffic scenes.

In recent years, several studies have combined traditional methodologies with deep neural networks to attain optimization. For instance, Gao et al. [31] introduced an intelligent stage light-based actor identification and positioning system. This system utilizes a tracking algorithm founded on deep convolutional neural networks, enabling the automatic tracking of actors by controlling the lighting system.

With the development CNN, a variety of CNN-based object detection models have emerged. The algorithms can be roughly divided into one-stage [8], [9], [10], [11], [12], [13] and two-stage [5], [6], [7] method depending on whether predicted bounding boxes are generated. Initially, most of the methods focus on the region proposal-based two-stage methods. Lu et al. [32] designed visual attention model based on Faster R-CNN to solve the problem of detecting small objects in large high-resolution images. Tabernik et al. [33] further improved the accuracy of traffic sign detection and recognition based on improved Mask R-CNN. However, due to the existence of region-based proposals generation, these methods are computationally expensive and difficult to achieve real-time requirements. In order to improve the inference speed of the two-stage method, a lot of work has begun to focus on the one-stage method. Zhu et al. [14] provided a new benchmark TT100K dataset and end-to-end CNN-based model which can detect and classify traffic signs at the same time. Yang et al. [34] proposed a multi-scale attention module based on YOLOv3 to better extract features and focus on tiny objects. Li et al. [35] combines the channel attention mechanism with the residual block to improve the ability of YOLOv4 to extract traffic sign features. Chatterjee et al. [36] presented a few-shot learning approach based on YOLOv5 for multimedia quality assessment to assess the quality of multimedia content without any bias and prejudices. Bai et al. [37] combined YOLOv5 with transformer self-attention module to improve the ability of feature extraction. Wang et al. [38] proposed the Ghost-YOLOX that uses GhostNet as the backbone network resulting in fewer number of parameters and computational costs. Although the one-stage TSDR methods meet the real-time requirements, but their detection accuracy has also decreased. In addition, most of the above methods focus on detecting simple categories, which is difficult to meet the reliability requirements of real driving scenarios. In order to detect more categories and alleviate the problem of difficult traffic signs detection, we propose a one-stage method called AE-YOLOX, which designed hierarchical traffic sign classification and adaptive integrated learning, and improved the overall detection accuracy of traffic signs.

2.3. **Ensemble Learning for Object Detection and Recognition.** Ensemble learning has been widely used to improve the performance of a single learner in recent years. Ensemble learning methods can be divided into two types, serial ensemble methods with strong dependency between individual learners and parallel ensemble methods without strong dependency between individual learners. The representative work of the former is Boosting [39] and those of the latter are Bagging [40], and Random forests [41], etc. In recent years, some studies have attempted to combine ensemble learning with object detection. Ren et al. [42] presented boosted local binary (BLB), which combines boosting with object detection. Xia et al. [24] presented a learning-based classification approach on RGB-D data, which combined the histograms of oriented gradients, 2D features and 3D Spin Image features, to represent the traffic-related objects. But it leads to the artificial gap between features, which is not suitable for general target detection. Wang et al. [25] used multiple stage classifiers in coarse-to-fine manner to detect objects. Xu et al. [26] designed non-maximum suppression (NMS) assembling and Feature assembling, which significantly improved the detection performance but the computational overhead and complexity are still unsatisfactory. In order to meet the accuracy and efficiency requirements of TSDR, it is the key to use the appropriate method to integrate the learners. Starting from the model architecture, our method effectively alleviates the problem of efficiency reduction caused by ensemble learning and achieves better performance by sharing the backbone and neck network of model.

3. Proposed method.

In this section, we propose the ACP method for data augmentation of tail categories and construct a new CTT100K dataset. Furthermore, we present our AE-YOLOX model for the TSDR. In this model, we first propose the RRS module to emphasize on difficult traffic signs, then employ the adaptive ensemble method with double-level head to reinforce the training of them, and finally, design the ACBF module to further enhance the inference speed.

3.1. Adaptive Copy-paste Augmentation of Dataset.

3.1.1. *Data Collection.* The TT100K dataset [15], first proposed in 2016, is the largest and popular Chinese TSDR dataset to date (we refer to it as the TT100K-2016 dataset in the following). This dataset contains 10,000 annotated images and 182 traffic sign categories. Among them, 6105 images are the training set, 3071 images are the testing set and 1548 images are other images. The images on TT100K-2016 dataset are acquired from real street scenes with 2048×2048 pixels resolution.

Based on the COCO detection challenge [43], we consider objects as large if they occupy an area more than 96×96 pixels, small if the area is less than 32×32 pixels, and medium if it lies between them. As is shown Figure 1, TT100K-2016 dataset suffers from a long-tailed distribution and serious class imbalance problem, and the size of traffic sign instances is generally concentrated in small or medium sized object. More seriously, 24 of the 182 categories appear only once in the whole dataset, which cannot satisfy the model training demand. Among the 182 categories, most of previous studies focused on only 45 of their top categories, which is shown Figure 2.

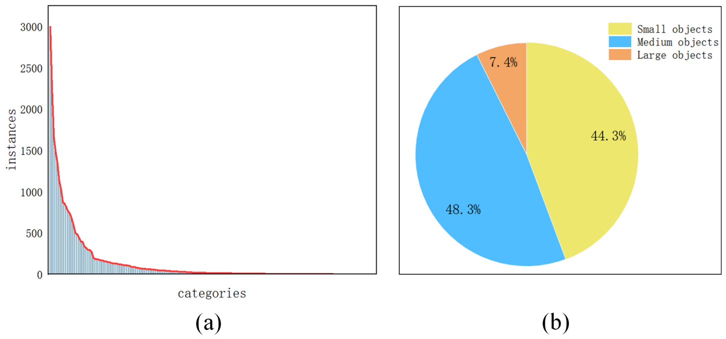


Figure 1. Distribution of TT100K-2016 dataset. (a) The instances of each category in TT100k-2016 dataset. (b) The distribution of instance size. It can be seen that the size of the TT100K-2016 dataset exhibits a long-tailed distribution.

It is clear that TSDR task focusing only on 45 categories can hardly meet the needs of real driving scenarios. Despite a large number of traffic sign datasets, the TSDR tasks for a large number of categories is still a challenging issue.

In response to these issues, TT100K-2016 dataset was further updated in 2021 (hereafter we refer to it as the TT100K-2021) to include more categories, expanded from 182 categories to 201 categories. However, TT100K-2021 dataset has censored some of the important categories (e.g., io, wo, po) in the TT100K-2016 dataset, causing some ambiguity and making it difficult to compare with other advanced methods. In addition, there are still 19 categories in the dataset with only 1 instance.

Therefore, we construct a new comprehensive traffic sign dataset based on the TT100K-2016 and the TT100K-2021 datasets, named the Comprehensive TT100K (CTT100K). We add some important categories to the TT100k-2021 and manually clean the dataset to address the problems of categories ambiguity. As is shown in Figure 3, the CTT100K

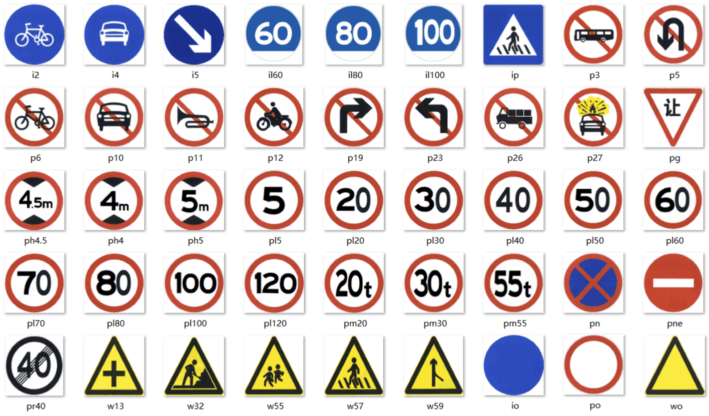


Figure 2. The most used 45 classes in the TT100K-2016dataset.

contains 204 categories (including io, po, wo removed by TT00K-2021) and it is the largest Chinese traffic sign dataset to date.

Data augmentation technique is used to create this dataset as it has been shown to be effective not only in improving the performance of deep neural networks by effectively preventing overfitting, but also in obtaining better generalization even on limited datasets [19]. For the missing instances, we will use data augmentation to expand the tail categories in the CTT100K.

3.1.2. *Adaptive Copy-paste Augmentation.* For some rare categories (categories with very few instances), collecting enough images itself is very difficult. Therefore, adaptive data augmentation of categories is a valuable tool in this case. Copy-paste [19] is a simple and effective data augmentation method. It can generate a large amount of synthetic training data freely by randomly pasting object instances onto the background image to improve the detection and recognition performance of the model, especially for rare categories. However, the object instances used in existing methods are often derived from the instance segmentation dataset itself, which is not provided in the TT100K-2021 dataset. Also, instance segmentation of datasets is usually costly and time-consuming. Therefore, we propose an effective ACP method for traffic signs with fixed contours, which contains shape-based mask generation, random copy-paste for training images generation and shape- based copy-paste for testing and validation images generation. In order to ensure the rationality of data augmentation, the whole ACP process uses only the original images based on the train-test-val split as presented in Section 3.1.1.

1) Shape-based Mask Generation

After pre-experiments, we employ the technique of the currently popular instance segmentation network OCRNet [44] to extract traffic signs, which is an object-contextual representations approach for semantic segmentation, characterizing a pixel by exploiting the representation of the corresponding object class. As is shown in Figure 4, using the pre-trained OCRNet model on the Cityscapes [45] dataset, we can successfully segment the street scene images and extract the traffic sign mask based on the traffic sign categories.

Because the traffic sign mask is coarse and contains invalid background information, refinement of the mask is necessary. The nature of the traffic-sign domain allows us to refine the masks using shape contours, which can be classified as circle, rectangle, triangle, inverted triangle and irregular, as is shown in Figure 5.

To get enough instance masks for different categories in a reproduceable way, we used a random combination of common image augmentation techniques (including perspective

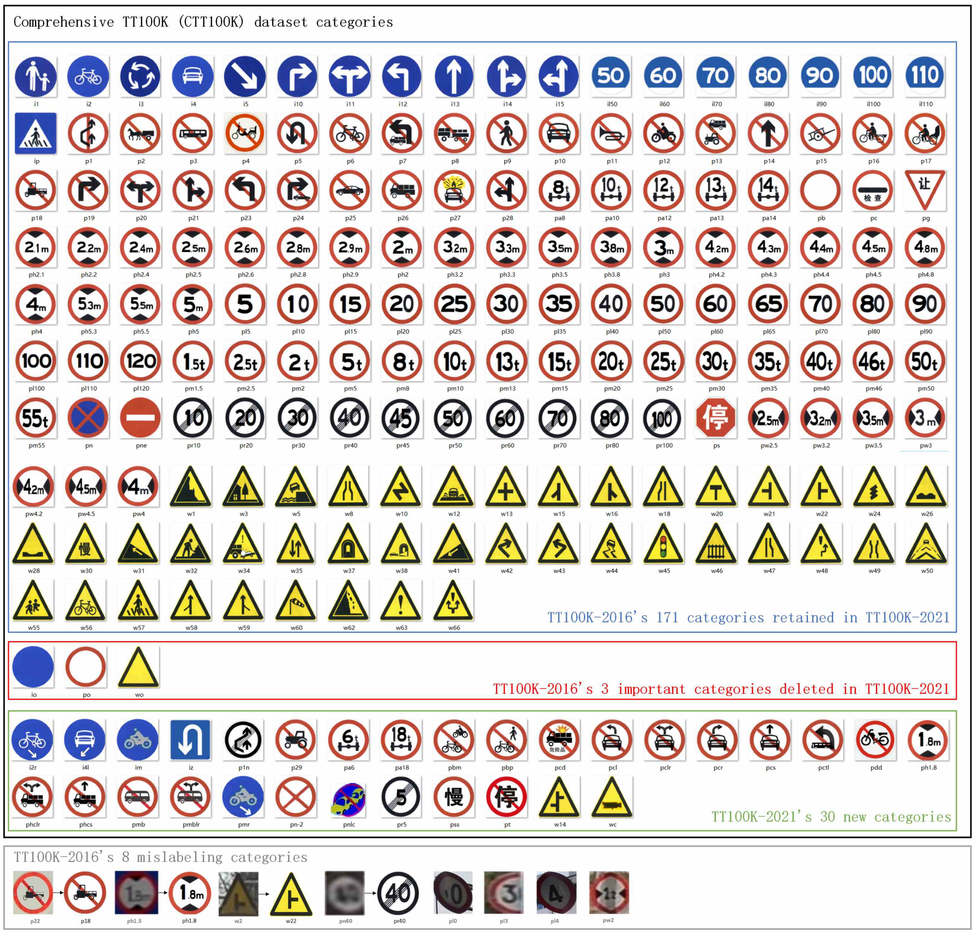

Comprehensive TT100K (CTT100K) dataset categories
TT100K-2016's 171 categories retained in TT100K-2021
TT100K-2016's 3 important categories deleted in TT100K-2021
TT100K-2021's 30 new categories
TT100K-2016's 8 mislabeling categories

Figure 3. The CTT100K dataset categories. Red boxes represent three new traffic sign categories added (e.g., io, wo, po) and the remaining 201 categories are the same as TT100K-2021 dataset.

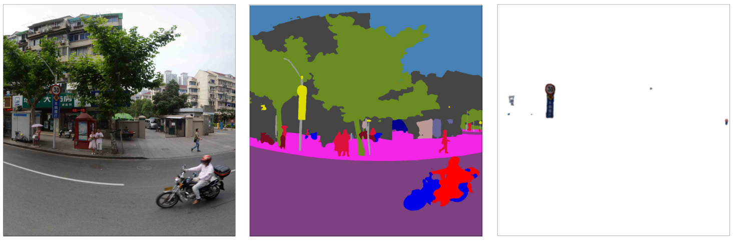


Figure 4. The process of OCRNet. (a) The original street scene images. (b) The result of segmentation using pre-trained OCRNet, where the yellow part is the location of the traffic sign. (c) The extracted traffic signs.

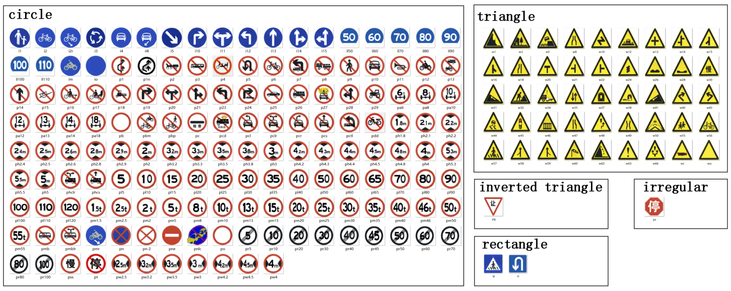


Figure 5. Shape-based mask.

change, changes in scale, geometric grid distortion and variations in brightness and contrast) to generate new masks. The results are shown in Figure 6. After this step, we can have a large number of object instance masks for different categories.

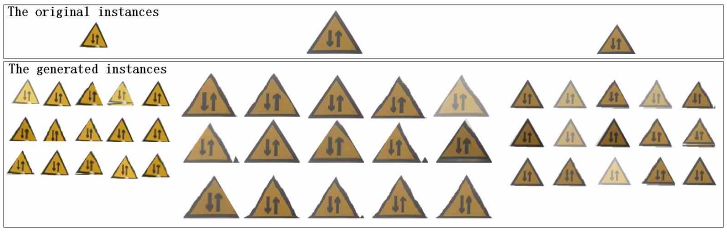
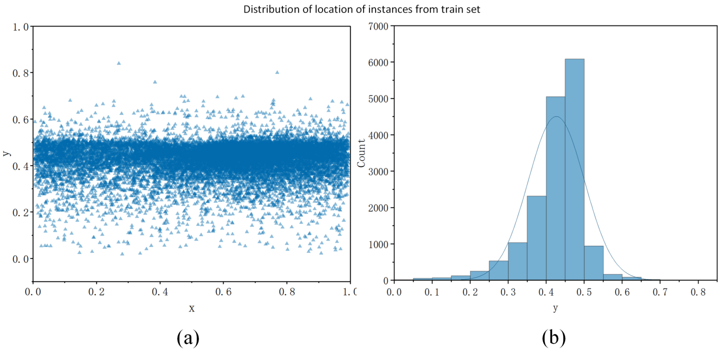


Figure 6. Some instance masks of the w53 class generated during the process of random combination of image augmentation techniques. Top: the original instances and bottom: the generated instances.

2) Random Copy-paste for Training Set

For the training set, in order to construct a large number of additional synthetic samples to meet the model training requirements, the masks generated earlier are adaptively pasted into the real training images using copy-paste for categories with less than 200 instances. In order to generate the most realistic synthetic images possible, we mostly keep the size of the segmented traffic sign instances constant to fit the real-world distribution.

As is shown in Figure 7, we find that traffic sign may be distributed in various locations of the images, but they generally show a uniform distribution on the x-axis and a normal distribution on the y-axis. Therefore, in our proposed method, we presented two different copy-paste methods, uniform distribution-based copy-paste and normal distribution-based copy-paste (we refer to them as UCP and NCP, respectively in the following) for enhancing the training set. Figure 8, UCP generates instances uniformly scattered in all positions of the image and NCP generates instances in a normal distribution, which is uniformly distributed in the x-axis and normally distributed in the y-axis. In subsequent experiments, their effectiveness is verified.



(a) (b)

Figure 7. Distributions of instances location from training set. (a) Traffic sign instances distributed in various locations of the images. (b) The histogram of the instances location on the y-axis.

3) Shape-based Copy-paste for Testing and Validation Set

In order to generate synthetic testing and validation samples that are as realistic as possible, we only performed data augmentation for the categories that did not appear in the testing and validation set among the 204 categories, using the shape-based copy-paste method instead of UCP or NCP above. As is shown in Figure 9, for each image, only one traffic sign was pasted at the location having the same shape to avoid the ambiguity

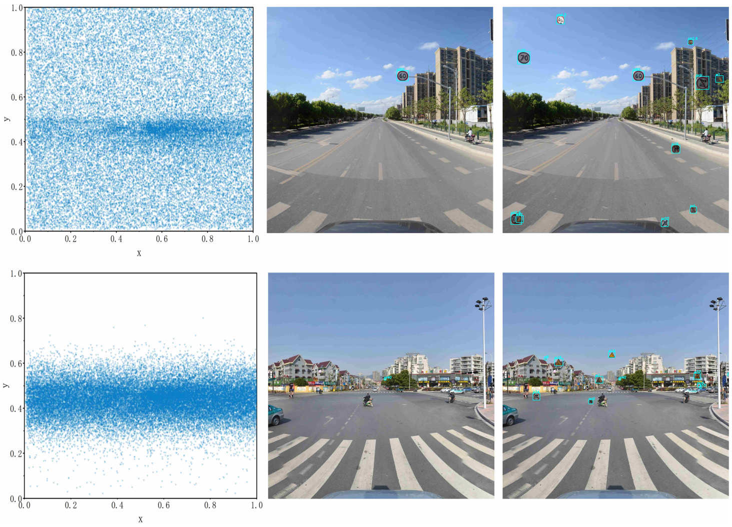


Figure 8. Training samples synthesized by different copy-paste. Top: instances synthesized by UCP and bottom: instances synthesized by NCP.

of the mixture of different shapes. To avoid randomness, the number of the augmented instances for each category was set to 5.

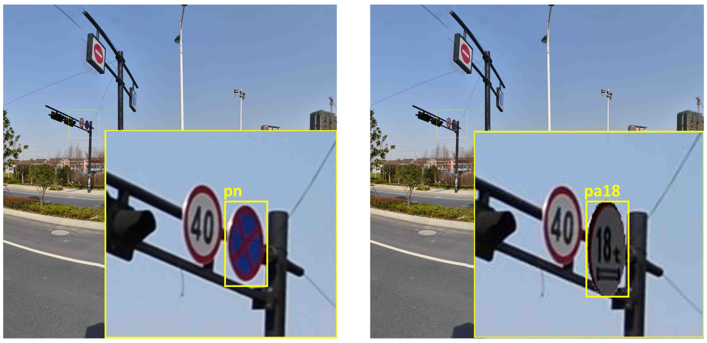


Figure 9. Testing samples synthesized by shape-based copy-paste augmentation. The large yellow box on the bottom right of each image is the enlarged image of the small yellow box inside the image.

3.1.3. *The Comprehensive TT100K Dataset.* We use the train-test split as presented in Section 3.1.1., and the 1548 other images are considered as a validation set. Using the augmentation techniques described above, we generated enough new instances to ensure that each category has at least 200 instances. As is shown in Figure 10, the CTT100K dataset is relative balanced. The ACP method resulted in around 32149 new traffic-sign instances spread over 3216 new synthetic training images, 90 instances located on new synthetic 90 testing images and 342 instances located on new synthetic validation images. After the ACP process, these synthetic images are blended with the real images for the final CTT100K dataset with 9371 training, 3161 testing and 1890 validation images.

3.2. **Adaptive Ensemble YOLOX.** This paper proposes a one-stage method called AE-YOLOX, the framework based on typical detector YOLOX [12] , which consists of the backbone CSPDarknet53, the neck SPP, the RRS module, double-level head module and the ACBF module. The overall framework of the proposed method is illustrated in Figure 11. The RRS module and double-level head are devoted to addressing the problems incurred by long-tailed data and difficult traffic signs, respectively. The ACBF module

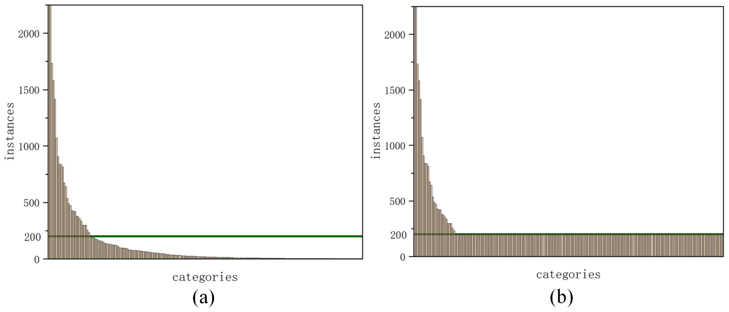


Figure 10. The distribution of number of instances over categories in the CTT100K dataset. (a) The instances of each category before ACP. (b) The instances of each category after ACP. Horizontal green line represents 200 instances per category, which we use as a cut-off point.

is used to further improve the detection efficiency. In this section, the proposed RRS module, double-level head and ACBF module are described in detail.

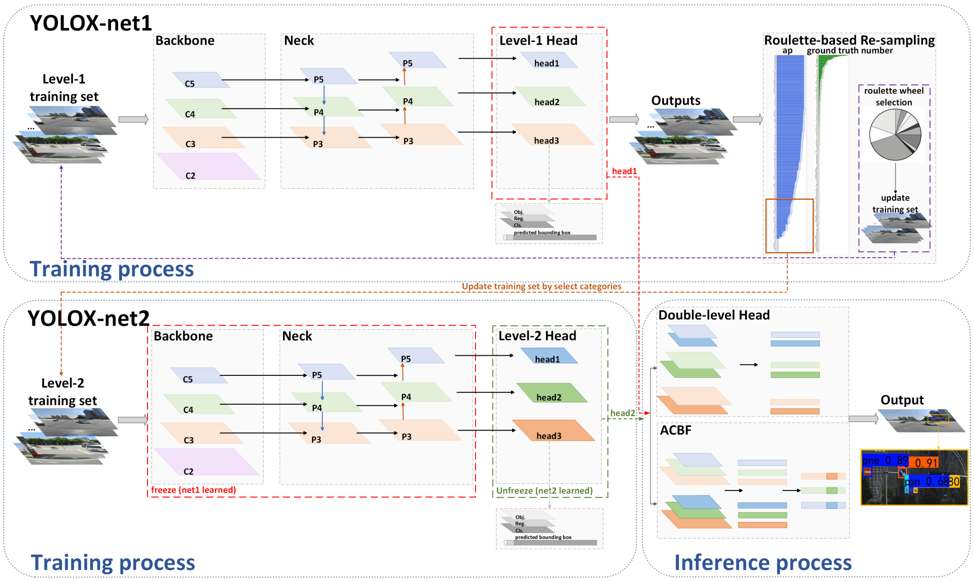
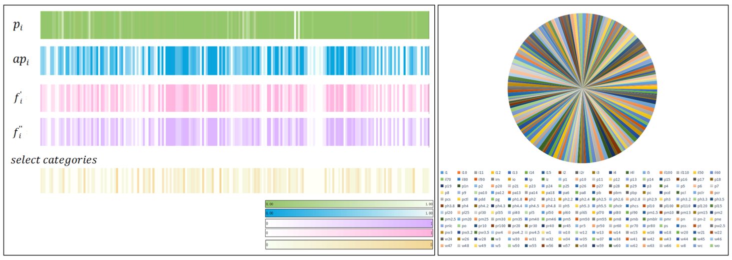


Figure 11. Overall framework of the proposed method that is divided into training process and inference process, including the backbone CSPDarknet53, neck Spatial Pyramid Pooling (SPP) layer, RRS module, double-level head module and ACBF module. In training process, backbone extracts features C2, C3, C4, C5, neck generates three feature maps P3, P4, P5, YOLO heads head1, head2, head3 detect the traffic signs, and RRS module updates the training set. Then, in inference process, stacked double-level heads net1 heads, net2 heads are used to detect the level-aware traffic signs. In addition, the ACBF module fuses the level-aware traffic signs to improve efficiency.

3.2.1. *Roulette-based Re-sampling for Long-Tailed Detection and Recognition.* The long-tailed distribution in the traffic sign dataset will generally cause the model fail to learn enough features in some categories. To alleviate this problem, we propose an effective RRS module to adaptively learn the resample ratio for each category, which is achieved based on the calculation of the average precision (AP, the calculation formulae is in Section 4.1) from validation set during the model training. Figure 12 shows the details of the implementation.



(a) (b)

Figure 12. A sample process of roulette-based re-sampling (RRS) module. (a) Visualization of parameter calculation required by RRS model (the deeper the color, the greater the probability). (b) Roulette wheel generated by RRS model (different colors represent different categories).

Roulette wheel selection is the simplest and most commonly used selection method, which is usually used in Genetic Algorithm (GA), in which the selection probability of each individual is proportional to its fitness value. The greater the fitness, the greater the probability of selection. However, in practice, the choice of individuals in roulette selection is often not based on the probability of individual selection, but on the 'cumulative probability' based on the following formulas.

$$p_i = \frac{N_i}{\sum_{i=1}^{n} N_i} \tag{1}$$

$$f'_i = \frac{(1 - AP_i)}{\sum_{i=1}^{n} (1 - AP_i)} \tag{2}$$

$$f''_i = \frac{(1 - AP_i) \bullet (1 - p_i)}{\sum_{i=1}^{n} (1 - AP_i) \bullet (1 - p_i)} \tag{3}$$

where n is the number of categories, N_i is the number of instances in category i, p_i denotes the percentage of the number of instances of each category to all instances, AP_i is the average precision calculated in category i, f'_i and f''_i are the probability of category i being selected by two different roulette methods. So the cumulative probability q_i can be calculated by

$$q_i = \sum_{j=1}^{i} f'_j \ or \ q_i = \sum_{j=1}^{i} f''_j \tag{4}$$

The q_i for category i is used to dynamically iterate the dataset during the training process by increasing the proportion of categories with low number of instances or poor performance in the dataset to optimally utilize the dataset and achieve better performance. The model is modified through the dataset to prevent overfitting.

The overall process of the RRS module flow is given in Algorithm 1. In this algorithm, after setting a reasonable ratio of expanded dataset's size and resampled classes, we can calculate the number of resampled classes N_r and the number of additional instances N_a. Then, f'_i and f''_i are calculated and the corresponding roulette is generated. By using the RRS module, we can perform data re-sampling and update the training set. It is worth noting that the initial training set is guaranteed to remain unchanged, and only the additional resampled part is modified each time, to ensure the training set not overemphasize the data of the difficult categories and the model not fall into overfitting.

Algorithm 1 Roulette-based Re-sampling (RRS)

```
Input: RRS(-) the training dataset, including all images and annotations from original
    training dataset, number of instances $D$, number of classes $C$, ratio of expanded
    dataset's size $a$, ratio of resampled classes $b$, interval epochs $k$, training epochs $T$,
    first training epochs $Q$.
Output: Updated training set
for $t = 1, 2, 3, \ldots, T$: do
    if $t <= Q$ then
        train
    else
        if $t \ \% \ k == 0$ then
            calculate $AP$, $N$, $f_i'$, $f_i''$, $q$ of all classes
            $Nr = b \bullet C$
            $Na = ( \ a \bullet D$ - $D \ ) \ / \ Nr$
            for $i = 1, 2, 3, \ldots, N_r$: do
                roulette wheel selection($i$, $q$)
            end for
            for $j = 1, 2, 3, \ldots, N_r$: do
                resample($j$, $N_a$)
            end for
        end if
    end if
    generate updated training dataset
    train
end for
```

In summary, the RRS module can expand the training data set reasonably and efficiently in a certain proportion, and emphasize only the data of difficult categories during model training to save the training time.

3.2.2. *Double-level YOLOX Ensemble for Difficult Traffic Sign Detection and Recognition.* Due to the accuracy and efficiency requirements of TSDR task, we proposed the double-level head for adaptively utilizing ensemble learning. Considering the high real-time performance of the one-stage method, we use YOLOX as the base learner. In addition, we only use YOLOX as the base learner in the whole integration process, because using the same model as an individual learner can avoid the ensemble ambiguity caused by the diversity of heterogenous learners.

After the training of the YOLOX-net1 model, the AP of each category is obtained that can be used to classify the difficulty of the traffic signs. Then, the YOLOX-net2 model that freezes the backbone and the neck is used to train the traffic signs of the difficult signs separately to learn the level-aware head. In inference process, different level-aware heads are integrated as a double-level head to obtain the final predicted bounding boxes. In practice, we develop a 6-step training process for the double-level YOLOX ensemble, as is shown in Algorithm 2.

In this algorithm, the YOLOX-net1 model is first trained as the base learner-1 using the level-1 training set for all categories in step1. This YOLOX-net1 model will generate N predicted bounding boxes for NMS post-processing to produce final detection results. After completing the training of YOLOX-net1 in step1, the AP of each category obtained from it is calculated in step2. According to the AP results, YOLOX-net1 may perform well in most categories, but it may not be effective for some categories. Therefore, in step3, the categories with AP less than a predesignated threshold are selected as the level-2 categories for subsequent training. Then in step4, in order to make the model focus on the category with poor detecting performance, the level-2 categories with poor performance are used to update the training set as level-2 training set for base learner-2 training.

In step5, the YOLOX-net2 model is trained using the level-2 training set. The backbone and neck of YOLOX-net2 that are learned in YOLOX-net1 are frozen without further learning and weight updating. Only the level-2 head is trained to generate N predicted bounding boxes. This will effectively address the problem of further learning of level-2 categories, which are difficult.

Finally, in the inference stage of step6, the level-1 head from YOLOX-net1 and level-2 head from YOLOX-net2 are stacked to form the double-level head, which generates $2N$ predicted bounding boxes for NMS post-processing, then the $2N$ bounding boxes are filtered using the ACBF module described in the next subsection to produce the final N bounding boxes. By sharing the backbone and neck, a good compromise between efficiency and accuracy is ensured, and the detection accuracy of TSDR is improved without significantly reducing efficiency.

Algorithm 2 Double-level YOLOX Ensemble

```
1: Step1: Train YOLOX-net1 by original training set (level-1 training set) for all categories.
2: Step2: Calculate $AP$ for each category obtained by YOLOX-net1.
3: Step3: Adaptively select categories while $AP <$ threshold as level-2 category.
4: Step4: Update training set (level-2 training set) by level-2 categories.
5: Step5: Train YOLOX-net2 using level-2 training set for level-2 categories.
6: Step6: Assemble YOLOX trained in Step1 and Step5 as the final model for inference.
```

3.2.3. *Adaptive Class Box Fusion.* Due to the direct stacking of different level-aware heads, the model may generate too many predicted bounding boxes, which will reduce the inference speed. Therefore, we propose a novel ACBF module to combine predictions from different learners. Unlike previous work, in which the weighted boxes fusion [46] or the soft-NMS [47] was used to fuse the boxes of multiple models into a new box or to reduce the confidence of the predicted bounding boxes by weighting, which usually significantly reduces the speed of NMS although improved the accuracy, our ACBF module combines the category information and location information output by different level-aware heads to filter predicted bounding boxes.

The ACBF module with detailed structure of the adaptively generated level mask is shown in Figure 13. Using the categories of different levels generated in Section 3.2.2), we can generate the corresponding class mask. Each level corresponds to a class mask matrix, which is used to filter the N bounding boxes of the current level.

Specifically, for each predicted bounding box, the ACBF module assigns the mask value to 1 if it is produced at this level and to 0 if it is not. Then the class mask is multiplied with the $2N$ bounding boxes to produce the N bounding boxes. The working principle of the ACBF module can be described as follows:

$$Value = Boxes \odot Maskr \tag{5}$$

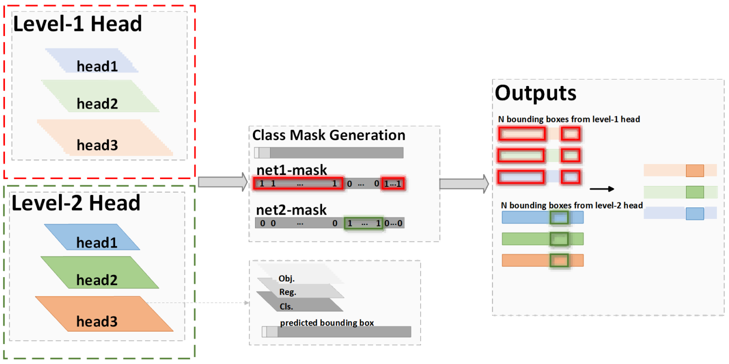


Figure 13. Principles of the adaptive class box fusion (ACBF) module.

In this way, the boxes of double-level head (level-1 head and level-2 head) are fused together through the ACBF module, then the NMS post-processing is performed on the fused bounding boxes to obtain the final predicted bounding boxes. The ACBF module will greatly reduce the number of boxes and improve the inference speed of the model.

4. **Experiment and Analysis.** We evaluate our proposed method against two typical one-stage methods: YOLOX [12] and YOLOv7 [13]. Initially, we conduct evaluations on the TT100K-2016 dataset to establish a baseline comparison. Subsequently, we evaluate our method on the newly proposed CTT100K dataset, accompanied by a comprehensive analysis of the proposed AE-YOLOX model.

4.1. **Evaluation Metrics.** In this study, various metrics are employed to assess the proposed approach. Precision and recall values serve as indicators for evaluating the model's performance, and these metrics can be calculated by:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

where TP represents the count of accurate detections, FP signifies the count of incorrect detections, and FN indicates the count of missed traffic signs.

We primarily use the COCO [43] metric Average Precision (AP) as the key measurement for assessing object detection performance. The mean Average Precision (mAP) is computed as the average of the AP values across all categories. The calculation for both AP and mAP involves the following steps:

$$AP = \int_0^1 P(R)\, dR \tag{8}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{9}$$

where n represents the number of categories, and APi denotes the average precision of the i-th category.

We employ two variants of mAP for evaluation: (i) mAP^{50}, utilizing the PASCAL visual object challenge [48] with an IoU overlap threshold of 0.50, and (ii) $mAP^{50:95}$, where the reported values are averaged across the IoU overlap range of [0.50, 0.95] with 0.05 increments, mirroring the range used in the COCO detection challenge [43]. Additionally, the COCO detection challenge [43] provides AP_S, AP_M and AP_L for assessing performance on small, medium, and large-scale objects, respectively.

In order to further compare the size and speed of the model, the evaluation also includes an assessment of floating-point operations per second ($FLOPs$) and frames per second (FPS).

4.2. **Experimental Setting.** The experiments were performed in the NVIDIA GeFerce RTX 3090 GPU, using PyTorch 1.12.1 framework, CUDA 11.6. The size of the input images is resized to 640 ×640 pixel. We further resize images due to memory limitations. The original 2048×2048 pixels high-resolution images to was resized to 640 ×640 pixels and caused a size loss of nearly 70%. Since traffic signs are small objects, mosiac and mixup methods are not used to avoid the negative impact of conventional methods on small objects.

In process of freeze training, the number of epochs set to 100, the learning rate was initialized at 0.001 and the training batch size was 32. In process of unfreeze training, the number of epochs set to 500, the learning rate was initialized at 0.0001 and the training batch size was 16.

4.3. **Comparison to the State-of-the-Art.** There are many existing methods that are mostly evaluated on non-public datasets, making it difficult to make reliable comparisons. Therefore, for a more fair comparison, we evaluated the proposed method on the TT100K-2016 dataset and indirectly compared it with other methods reported on it. We follow the train-test split of [14] , focusing only on 182 valid categories compared with 45 head categories (i. e., the number of instances per category is greater than 100). We calculated the measure of each category and then reported the average of all categories (our 182 categories eliminated 8 mislabeled categories to ensure the validity of the experiment). Detailed results on the TT100K-2016 dataset are reported in Table 1. It shows that our method on 45 categories and 182 categories achieved mAP^{50} of 89.3% and 81.5%, respectively.

Compared with related work, Feng et al. [49] obtained a better mAP^{50} of 93.5% in 45 categories. This is because they used 1024×1024 pixels as input, which is 2.56 times of ours. However, our method performs the best when applied to 182 categories. It can be seen that in the case of a small number of categories, the degree of improvement is limited. In the case of a large number of categories with long-tailed distribution, our method shows its superiority in improving the performance of most evaluation metrics.

4.4. **Evaluation on Comprehensive TT100K Traffic-Sign Dataset.** We use the train-test-val split as presented in Section 3.1.3 with 204 categories and 9371 images as training, 3161 images as testing and 1890 images as validation sets.

1) Data Augmentation Evaluation

We first evaluate two different copy-paste methods UCP and NCP applied to the dataset. This allows us to assess the quality of synthetic training images which were generated by the ACP module before they are used to full pipeline learning. We expand each category to 200 instances, and paste 10 traffic sign masks per image.

Results are reported in Table 2, uniform ACP method is better in most metrics. Although normal ACP method based on location distribution is more similar to the real-world distribution, it only performs slightly better than uniform ACP on large objects. This may be because the predicted boxes generated are uniformly distributed, while the instances generated by normal ACP are denser at the center of y-axis than those generated by uniform ACP, some of the instances at the center of y-axis may not be covered by the predicted boxes, which ultimately leads to poor results.

Table 1. Comparison of our method with other method on the TT100K-2016 dataset

Method	Class	Size	mAP^{50}	$mAP^{50:95}$	AP_S	AP_M	AP_L	R	P	$FLOPs(G)$	FP
Lu [32]	45	480x480	87.0	-	-	-	-	83.4	91.7	-	-
Chen [50]	45	608x608	90.2	-	-	-	-	-	-	-	-
Wu [51]	45	1024x1024	82.6	-	-	-	-	-	-	-	-
Feng [49]	45	1024x1024	93.5	74.5	59.7	80.1	85.4	-	-	44.3	-
YOLOv7	45	640x640	78.9	52.8	24.0	63.1	78.4	67.3	95.3	54.1	65
YOLOXm	45	640x640	85.2	50.7	28.3	58.6	76.8	80.9	89.9	37.1	89
Ours	45	640x640	89.3	54.9	32.5	63.0	77.9	86.4	91.1	48.5	71
Chen [52]	182	608x608	66.9	-	**47.5**	57.4	57.4	-	-	-	-
Wang [53]	182	608x608	65.1	-	41.5	57.8	58.2	-	-	**17.9**	-
YOLOv7	182	640x640	74.7	54.7	17.0	62.3	79.8	66.0	**88.2**	54.1	65
YOLOXm	182	640x640	74.4	50.6	25.8	56.6	76.6	70.4	83.7	37.1	**87**
Ours	182	640x640	**81.5**	**56.7**	25.9	**62.9**	**81.2**	**75.5**	86.3	49.0	72

Table 2. Comparison of our method with other method on the TT100K-2016 dataset

Method	Class	mAP^{50}	$mAP^{50:95}$	AP_S	AP_M	AP_L
YOLOXm with uniform ACP	204	73.5	**50.4**	**25.0**	**56.9**	75.8
YOLOXm with normal ACP	204	**74.4**	49.1	22.2	55.0	**78.9**
YOLOXm with uniform ACP	45	**87.5**	**54.5**	**32.4**	**62.7**	**78.9**
YOLOXm with normal ACP	45	86.1	50.1	26.9	58.3	78.2

2) Data Re-sampling Evaluation

Results are reported in Table 3. When we increase the least category to 200 instances, the model with RRS module obtains an mAP^{50} of 74.1%, which is 0.4 points higher than baseline YOLOXm, and it performs better on small and large objects. While the least category is increased to 500 instances, our method further improves the mAP^{50} to 76.0%. The reason is that a larger dataset can guarantee a more diversified data distribution to simulate the real-world situation and provide more valuable information for the model to learn more about the details and features of traffic signs. We can conclude that the RRS module performs better in the case of more instances, which is also consistent with the conclusion of [27].

Table 3. Comparison of different copy-paste methods

Method	mAP^{50}	$mAP^{50:95}$	AP_S	AP_M	AP_L
YOLOXm (200 instances)	73.7	**50.4**	26.0	**57.0**	75.4
YOLOXm with RRS (200 instances)-f'	**74.1**	50.2	**26.3**	56.1	**76.2**
YOLOXm with RRS (200 instances)-f''	71.6	48.0	25.2	52.6	76.5
YOLOXm with RRS (500 instances)-f'	76.0	52.4	26.2	58.4	77.6
YOLOXm with RRS (500 instances)-f''	**72.0**	**48.6**	**24.6**	**53.7**	**76.1**

3) Full Pipeline Evaluation

In the experiment, we set the threshold of select level-2 categories as 0.7. And as described above, we use uniform ACP to expand each category to 200 instances, including the baseline model. We report our results in terms of the mAP over all 204 categories as well as the FPS.

Results are reported in Table 4 that clearly shows that the best results are achieved using our AE-YOLOX. Nevertheless, the RRS module is not necessary for the AE-YOLOX model as the later has already had the capability of emphasizing difficult categories. Although the RRS and the double-level head module methods are used at the same time, the $mAP^{50:95}$ is improved by 3% compared with the baseline. Compared to the original YOLOXm, our method increases $mAP^{50:95}$, AP_S, AP_M and AP_L by 3.6%, 1.5%, 3.3% and 5.4%, respectively. It significantly improves mAP^{50} from 73.7% to 79.0%. Moreover, using the ACBF module improves the detection speed by 2 FPS compared to the original double-level head.

Table 4. Results on the Comprehensive TT100K dataset

Method	mAP^{50}	$mAP^{50:95}$	AP_S	AP_M	AP_L	FPS
YOLOXm (baseline)	73.7	50.4	26.0	57.0	75.4	87
YOLOX with RRS and double-level head	76.7	51.7	26.8	57.8	78.7	70
YOLOX with double-level head	**79.0**	**54.0**	**27.5**	**60.3**	**80.8**	70
YOLOX with double-level head and ACBF	**79.0**	**54.0**	**27.5**	**60.3**	**80.8**	**72**

Some exemplified results obtained from our method and the baseline YOLOXm on the CTT100K dataset are visualized in Figure 14. It can be seen that our method achieves better results on difficult objects and tail classes. In the first column, YOLOXm missed the small-scale traffic signs on the pole, while our method successfully detected traffic signs such as w42, pbm and io, which benefited from the introduction of ensemble learning approach. After the learning of level-2 traffic sign categories, the problem of miss-detection of these traffic signs is improved.

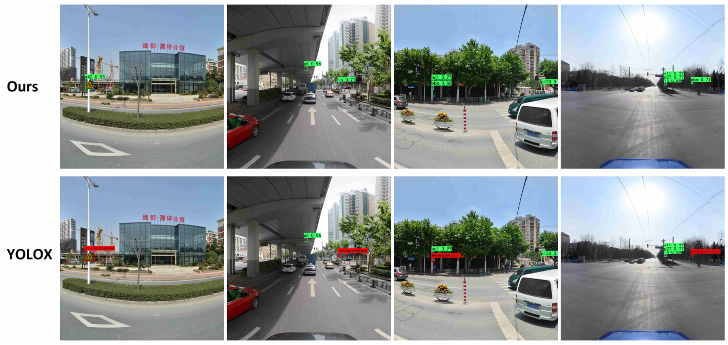


Figure 14. Detection results of our proposed method (top) and YOLOX (bottom) on the CTT100K dataset. The missed signs on the bottom are marked by red.

Some examples of our results in various challenging cases are shown in Figure 15. We can see that our method is robust to some challenging conditions (e.g., the small size of traffic signs, partial occlusion, angle variations, fading aging, illumination changes, weather and other complex environmental disturbances) mentioned above. The top row shows some examples of changes in lighting conditions such as underneath bridges and the entrances of tunnels, and the bottom row shows some examples of other various challenges, for which our method performed well.

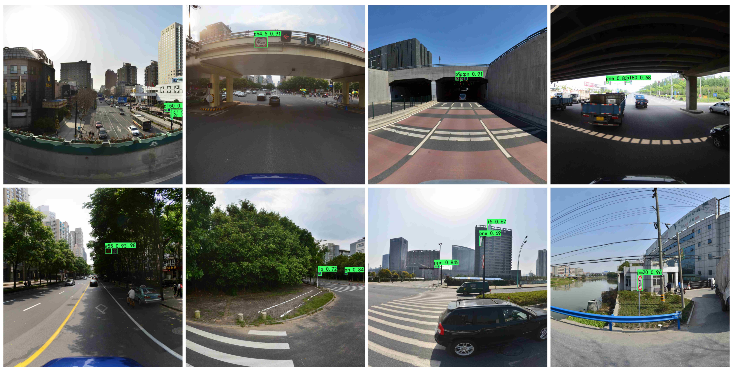


Figure 15. Results for some challenging cases in the CTT100K dataset. Top: cases of different lighting conditions (e.g., extremely bright areas under strong sunlight and dark areas under bridges and tunnels) and bottom: from left to right, there are challenges of extremely small scale objects, partial occlusion, angle change, fading aging, angle variations.

In some extremely difficult cases, Figure 16 shows some missed detections, which show the missed and false detection results of several complex examples of occlusion and fairly small objects. Even for humans, huge perspective changes and significant occlusion can cause difficulties.

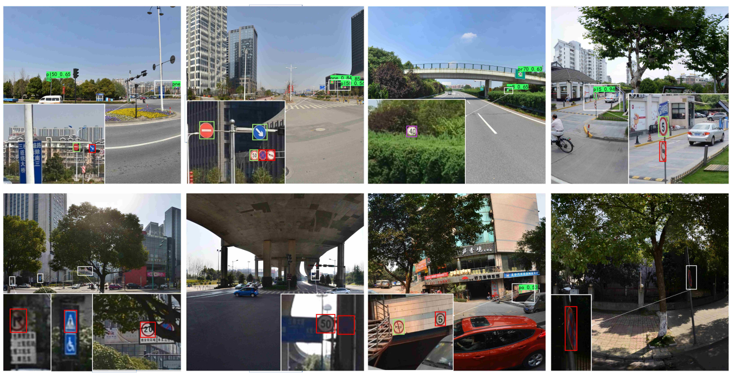


Figure 16. Examples of traffic signs with a fixed content but poor detection rate on the test set of the Comprehensive TT100K dataset. True-positive detections are marked in green, false positives in magenta and miss-detections (false negatives) in red.

4.5. **Ablation Study.** In order to verify the effectiveness of our proposed methods, ablation experiments are performed on the CTT100K dataset. Table 5 shows the ablation results of incrementally integrating the ACP method, RRS module, double-level head and ACBF module into the baseline YOLOXm.

In Table 5, ACP-U represents the uniform ACP method, which increases the mAP^{50} and $mAP^{50:95}$ by 27.5% and 19.8%, respectively. This confirms the effectiveness of the uniform ACP in improving the model learning using a suitable dataset. Then, the RRS module further increases the mAP^{50} by 0.4%, which indicates the effectiveness of our roulette-based re-sampling method. DL represents the double-level head module. Through the analysis of previous experiments, RRS module and double-level head are not necessary for simultaneous use. After the introduction of double-level head module, mAP^{50} and $mAP^{50:95}$ achieve 79.0% and 54.0%. Furthermore, integrating the uniform ACP method and double-level head achieves the best performance on both mAP^{50} and $mAP^{50:95}$ with acceptable additional computational cost. In addition, the ACBF module increases the inference speed by 2 FPS. These results validate the effectiveness of the ACP method, the RRS module on the long-tail data problem, and the effectiveness of double-level head and the ACBF module on difficult traffic signs.

Table 5. Ablation experiments of the proposed method on CTT100K dataset

ACP-U	RRS	DL	ACBF	mAP^{50}	$mAP^{50:95}$	$FLOPs(G)$	FPS
				46.2	30.6	7.1	87
✓				73.7	50.4	37.1	87
✓	✓			74.1	50.2	37.1	70
✓		✓		79.0	54.0	49.0	70
✓		✓	✓	**79.0**	**54.0**	**49.0**	**72**

5. **Discussion and Conclusion.** Collecting and annotating the right data for supervised learning is an expensive and challenging task but can significantly improve the performance of any perception-based autonomous driving system. Based on the long-tailed data problem that naturally exists in the traffic sign dataset, we presented the ACP method, a data augmentation approach based on shape and copy-paste. With no additional data collection effort, we synthesize annotated images and presented a new dataset, named the CTT100K dataset, with enough instances for a large number of categories. To solve the problem of difficult categories, this paper proposed a novel ensemble method to improve YOLOX [12], called the adaptive ensemble YOLOX (AE-YOLOX) model.

Our experimental results show a strong improvement in the performance by using our AE-YOLOX model with the mAP^{50} having been increased by 5.3% compared to the baseline model on the CTT100K. A highlight of the proposed method is its ability to guarantee high accuracy without significantly degrading the processing efficiency. Furthermore, our method is also applicable to other fields, e.g., the medical field, where the data is generally scarce and expensive to collect.

Our study demonstrates that ensemble learning is very effective in improving the detection accuracy, and makes the model more reliable under the condition that it basically does not affect the efficiency. The method proposed in this paper is plug-and-play, and we believe that our work can be combined with other existing object detection methods. In the future, further work will focus on the following three directions:

1. To explore more efficient ensemble mode and reduce network complexity without compromising accuracy;

2. To further improve the algorithm to achieve better detection accuracy for the case of missed false detections in small objects;

3. To propose traffic sign datasets under bad weather conditions for real driving scenarios where the traffic signs may be blurred by such as rain, fog or other accidental damage, and to improve the performance of TSDR task.

Acknowledgment. This work was supported in part by the Open Project Program of The Key Laboratory of Cognitive Computing and Intelligent Information Processing of Fujian Education Institutions, Wuyi University (KLCCIIP202303), the Fujian Provincial Transportation Science and Technology Demonstration Project (Grant No. FJJT-KJSF2021-01), and the Natural Science Foundation Project of Fujian Province (No.

2022J01953), titled "Research on Key Technologies of Quasi-Real-Time Multi-Citation Blockchain Network".

Data availability statement. The original data in our study is openly available from TT100K dataset (http://cg.cs.tsinghua.edu.cn/traffic-sign/) and the processed data, the CTT100K dataset, are available upon request by contacting the corresponding author.

REFERENCES

[1] J. F. Khan, S. M. Bhuiyan, and R. R. Adhami, "Image segmentation and shape analysis for road-sign detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 83–96, 2010.

[2] C. Sakaridis, D. Dai, S. Hecker, and L. Van Gool, "Model adaptation with synthetic and real data for semantic dense foggy scene understanding," in *Proceedings of the European Conference on Computer Vision (ECCV).* Springer, 2018, pp. 687–704.

[3] J. Greenhalgh and M. Mirmehdi, "Real-time detection and recognition of road traffic signs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1498–1506, 2012.

[4] Y. Yang, H. Luo, H. Xu, and F. Wu, "Towards real-time traffic sign detection and classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 2022–2031, 2015.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2014, pp. 580–587.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV).* IEEE, 2017, pp. 2961–2969.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV).* Springer, 2016, pp. 21–37.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2016, pp. 779–788.

[10] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2017, pp. 7263–7271.

[11] ——, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[12] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[13] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2023, pp. 7464–7475.

[14] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The german traffic sign detection benchmark," in *The 2013 International Joint Conference on Neural Networks (IJCNN).* IEEE, 2013, pp. 1–8.

[15] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2016, pp. 2110–2118.

[16] N. Dvornik, J. Mairal, and C. Schmid, "Modeling visual context is key to augmenting object detection datasets," in *Proceedings of the European Conference on Computer Vision (ECCV).* Springer, 2018, pp. 364–380.

[17] H.-S. Fang, J. Sun, R. Wang, M. Gou, Y.-L. Li, and C. Lu, "Instaboost: Boosting instance segmentation via probability map guided copy-pasting," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV).* IEEE, 2019, pp. 682–691.

[18] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV).* IEEE, 2017, pp. 1301–1310.

[19] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2021, pp. 2918–2928.

[20] S. Ando and C. Y. Huang, "Deep over-sampling framework for classifying imbalanced data," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part I 10.* Springer, 2017, pp. 770–785.

[21] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[22] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV).* IEEE, 2017, pp. 2980–2988.

[23] Z. Chen, V. Casser, H. Kretzschmar, and D. Anguelov, "Gradtail: Learning long-tailed data using gradient-based sample weighting," *arXiv preprint arXiv:2201.05938*, 2022.

[24] Y. Xia, X. Shi, and N. Zhao, "Learning for classification of traffic-related object on rgb-d data," *Multimedia Systems*, vol. 23, pp. 129–138, 2017.

[25] D. Wang, X. Hou, J. Xu, S. Yue, and C.-L. Liu, "Traffic sign detection using a cascade method with fast feature extraction and saliency test," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3290–3302, 2017.

[26] J. Xu, W. Wang, H. Wang, and J. Guo, "Multi-model ensemble with rich spatial information for object detection," *Pattern Recognition*, vol. 99, p. 107098, 2020.

[27] Y. Yang and Z. Xu, "Rethinking the value of labels for improving class-imbalanced learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 290–19 301, 2020.

[28] E. Gao, W. Huang, J. Shi, X. Wang, J. Zheng, G. Du, and Y. Tao, "Long-tailed traffic sign detection using attentive fusion and hierarchical group softmax," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24 105–24 115, 2022.

[29] F. Zhang, T.-Y. Wu, J.-S. Pan, G. Ding, and Z. Li, "Human motion recognition based on svm in vr art media interaction environment," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, p. 40, 2019.

[30] F. Zhang, T.-Y. Wu, and G. Zheng, "Video salient region detection model based on wavelet transform and feature comparison," *EURASIP Journal on Image and Video Processing*, vol. 2019, no. 1, p. 58, 2019.

[31] J. Gao, H. Zou, F. Zhang, and T.-Y. Wu, "An intelligent stage light-based actor identification and positioning system," *International Journal of Information and Computer Security*, vol. 18, no. 1-2, pp. 204–218, 2022.

[32] Y. Lu, J. Lu, S. Zhang, and P. Hall, "Traffic signal detection and classification in street views using an attention model," *Computational Visual Media*, vol. 4, pp. 253–266, 2018.

[33] D. Tabernik and D. Skočaj, "Deep learning for large-scale traffic-sign detection and recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1427–1440, 2019.

[34] T. Yang and C. Tong, "Real-time detection network for tiny traffic sign using multi-scale attention module," *Science China Technological Sciences*, vol. 65, no. 2, pp. 396–406, 2022.

[35] Y. Li, J. Li, and P. Meng, "Attention-yolov4: A real-time and high-accurate traffic sign detection algorithm," *Multimedia Tools and Applications*, vol. 82, no. 5, pp. 7567–7582, 2023.

[36] R. Chatterjee, A. Chatterjee, S. H. Islam, and M. K. Khan, "An object detection-based few-shot learning approach for multimedia quality assessment," *Multimedia Systems*, vol. 29, no. 5, pp. 2899–2912, 2023.

[37] W. Bai, J. Zhao, C. Dai, H. Zhang, L. Zhao, Z. Ji, and I. Ganchev, "Two novel models for traffic sign detection based on yolov5s," *Axioms*, vol. 12, no. 2, p. 160, 2023.

[38] C.-Z. Wang, X. Tong, J.-H. Zhu, and R. Gao, "Ghost-yolox: A lightweight and efficient implementation of object detection model," in *2022 26th International Conference on Pattern Recognition (ICPR).* IEEE, 2022, pp. 4552–4558.

[39] R. E. Schapire and Y. Freund, "Boosting: Foundations and algorithms," *Kybernetes*, vol. 42, no. 1, pp. 164–166, 2013.

[40] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.

[41] ——, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[42] H. Ren and Z.-N. Li, "Object detection using boosted local binaries," *Pattern Recognition*, vol. 60, pp. 793–801, 2016.

[43] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of the European Conference on Computer Vision (ECCV).* Springer, 2014, pp. 740–755.

[44] Y. Yuan, X. Chen, X. Chen, and J. Wang, "Segmentation transformer: Object-contextual representations for semantic segmentation." *arXiv preprint arXiv:1909.11065*, 2019.

[45] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2016, pp. 3213–3223.

[46] R. Solovyev, W. Wang, and T. Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models," *Image and Vision Computing*, vol. 107, p. 104117, 2021.

[47] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-nms–improving object detection with one line of code," in *Proceedings of the IEEE international Conference on Computer Vision (ICCV)*, 2017, pp. 5561–5569.

[48] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2010.

[49] A. Feng, X. Wu, and T. Xu, "Real-time traffic sign detection algorithm combining attention mechanism and contextual information," *Journal of Frontiers of Computer Science and Technology*, vol. 17, no. 11, p. 2676, 2023.

[50] J. Chen, K. Jia, W. Chen, Z. Lv, and R. Zhang, "A real-time and high-precision method for small traffic-signs recognition," *Neural Computing and Applications*, pp. 1–13, 2022.

[51] Y. Wu, Z. Li, Y. Chen, K. Nai, and J. Yuan, "Real-time traffic sign detection and classification towards real traffic scene," *Multimedia Tools and Applications*, vol. 79, pp. 18 201–18 219, 2020.

[52] Y. Chen, J. Wang, Z. Dong, Y. Yang, Q. Luo, and M. Gao, "An attention based yolov5 network for small traffic sign recognition," in *2022 IEEE 31st International Symposium on Industrial Electronics (ISIE).* IEEE, 2022, pp. 1158–1164.

[53] J. Wang, Y. Chen, Z. Dong, and M. Gao, "Improved yolov5 network for real-time multi-scale traffic sign detection," *Neural Computing and Applications*, vol. 35, no. 10, pp. 7853–7865, 2023.