Meta-YOLO: Meta-Learning for Few-Shot Traffic Sign Detection via Decoupling Dependencies

Xinyue Ren [1], Weiwei Zhang [1,2,3,*], Minghui Wu [1], Chuanchang Li [1] and Xiaolan Wang [1]

1   School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China; xinyueRen0816@163.com (X.R.); wmhui@yeah.net (M.W.); lccsues@163.com (C.L.); xiaolanwang@sues.edu.cn (X.W.)
2   Shanghai Smart Vehicle Integration Innovation Center Co., Ltd., Shanghai 201620, China
3   School of Vehicle and Mobility, Tsinghua University, Beijing 100089, China
*   Correspondence: zwwsues@163.com; Tel.: +86-158-3885-1181

Abstract: Considering the low coverage of roadside cooperative devices at the current time, automated driving should detect all road markings relevant to driving safety, such as traffic signs that tend to be of great variety but are fewer in number. In this work, we propose an innovative few-shot object detection framework, namely Meta-YOLO, whose challenge is to generalize to the unseen classes by using only a few seen classes. Simply integrating the YOLO mechanism into a meta-learning pipeline will encounter problems in terms of computational efficiency and mistake detection. Therefore, we construct a two-stage meta-learner $\mathcal{F}$ model that can learn the learner initialization, the learner update direction and learning rate, all in a single meta-learning process. To facilitate deep networks with learning, the fidelity features of the targets improve the performance of meta-learner $\mathcal{F}$, but we also design a feature decorrelation module (FDM), which firstly transforms non-linear features into computable linear features based on RFF, and secondly perceives and removes global correlations by iteratively saving and reloading the features and sample weights of the model. We introduce a three-head module to learn global, local and patch correlations with the category detection result outputted by the aggregation in meta-learner $\mathcal{F}$, which endows a multi-scale ability with detector ϕ. In our experiments, the proposed algorithm outperforms the three benchmark algorithms and improves the mAP of few-shot detection by 39.8%.

Keywords: traffic signs detection; few-shot detection; feature decorrelation; meta-learning

1. Introduction

Traffic sign detection is the premise for driverless cars to understand traffic information, avoid traffic congestion and accidents, and ensure the safe and orderly driving of vehicles. It is also an essential submodule of driver assistance systems. Currently, CNNs are widely used for traffic sign detection [1–4], which relies heavily on a large number of accurate bounding box annotations and artificially balanced training classes. When the contextual information of the training and testing sets is distributed unevenly, a serious mistake will occur that fails to generalize. It is a great challenge to guarantee the accuracy and robustness of detection results when samples are limited because of the large variation in object scales, such as vehicle speed, and the inconsistencies in traffic signs due to regional differences.

Meta-YOLO draws inspiration from classical CNNs and few-shot techniques. We outline some of the salient works here to set the context.

Traffic Sign Detection by CNNs. In recent years, researchers have generally adopted a visual scheme based on deep CNNs (convolutional neural networks) to achieve the task of traffic signs detection. A one-stage detector is based on the regression method, which directly outputs the location and category of the bounding boxes densely in a single-shot, such as YOLOv1-v4 [5–8], SSD [9] and RetinaNet [10]. A two-stage detector is based on

a region proposal, including R-CNN [11], Fast R-CNN [12], Faster R-CNN [13] and Mask R-CNN [14].

In view of the complex background and unbalanced sample distribution, Li et al. [15], on the basis of a fully study of the relationship between different traffic signs with digital characters, designed an SE block that could automatically learn the importance of each channel from global information. This method simplifies the detection of a wide variety of numerical traffic signs to 10 digital categories, but it is difficult to distinguish similar false targets in complex real traffic scenes. Min et al. [16] proposed an LW-RefineNet to segment the scene and obtain the information regarding the spatial positional at pixel level, and then the constraint model is constructed to establish the search regions. Experiments show that this method alleviates the mis-detection of small traffic signs. However, it is limited in scenarios where both sides of the road, and other scenarios (such as intersections), have ineffective detection. The above research shows that fully understanding and representing the real features of the extracted traffic signs is an effective solution to distinguish similar objects and filter false associations, which provides a basis for the design of the FDM module in this algorithm. Too small traffic signs are one of the main causes of mis-detection, and improving the multi-scale ability of detection algorithms is a common method to solve the challenge of small target detection. Cao et al. [17] presented an improved Sparse R-CNN and constructed hierarchical residual-like connections within each single radix block, while a cross-channel attention mechanism was added in the RoI division process to fuse shallow feature information. However, due to its single attention mechanism, global correlation in RPN may suffer from spatial scale dislocation, and local correspondence between objects may be ignored. Wang et al. [18] applied an inception and channel attention mechanism to a superclass detector and directly concatenated feature maps of different channels, which overcomes Cao et al.'s [17] complex backbone problems. At the same time, there is a negative impact on robustness because the importance of the different feature channels is not considered. Although the traffic sign detection algorithm based on CNN has achieved remarkable results in real-time performance and accuracy, most methods require a good deal of labeled sample data, and, in fact, our dataset cannot exhaust all traffic scenes. Based on this consideration, we combined the meta-paradigm with CNN to promote the robustness to unseen classes' tasks.

Few-shot object detection. Since the performance of large neural networks is limited by the size of the training set, a small number of samples within the training set can easily lead to overfitting of the network and failure to realize the potential of deep networks. Few-shot learning is a method of deep learning training and prediction with insufficient sample data. In few-shot tasks, models trained with small samples can easily fall into overfitting to small samples as well as underfitting to the target task.

Meta-learning has shown great potential in solving few-shot problems [19–22], and its unique implementation can improve the accuracy of target detection for the classification of novel categories. The main idea is to use meta-info accumulated from historical tasks as prior knowledge, and then learn a small number of target samples to quickly master new tasks. This method has strong adaptability and robustness to unseen scenarios. Meta-learning approaches used for few-shot detection are roughly divided into three groups: gradient-based [23,24], nearest-neighbor [25,26] and model-based [27,28]. Zhao et al. [29] proposed a multi-scale few-shot detection model based on fine-tuning, which utilizes residual involution blocks to construct the all feature learning architecture as well as design PAM to aggregate from all feature levels. This method exploits shallow feature semantic information for object location in the first stage and is partly fine-tuned on a small balanced dataset in the second stage. However, Zhang et al. [30] visualized the feature distribution of samples in the pre-training space, proving that fine-tuning has limited performance improvement in meta-learning and can easily increase the risk of base task overfitting. Therefore, in this work, we take meta-info to update the meta-learner to replace fine-tuning. Whang et al. [31] proposed a general object detection system, which combines the feature-based domain attention mechanism with sequence and exception networks, and assigns

network activation to different domains through SE adapter library learning, so as to automatically obtain the importance of each feature channel. The core idea of SENet is to learn the feature weight according to the loss, so that the weight of an effective feature map is large, and the weight of an ineffective feature map is small, so as to achieve better results. However, the general detection system ignores the problem of spatial dislocation, which leads to the poor performance in detecting traffic signs with small targets and a chaotic background. Han et al. [32] improved the problem of training on base training to generate candidate proposals for novel classes and missing high IOU boxes in the RPN stage. A coarse-grained prototype matching network (meta-RPN) was proposed, which takes a non-linear classifier based on metric learning to replace the traditional linear target classifier, dealing with the similarity between anchor boxes and novel classes in query images, so as to improve the recall of the few novel class candidate boxes. A fine-grained prototype matching network (meta-classifier) was designed. The network has spatial feature alignment and foreground attention modules to deal with the similarity between noise and novel classes, so as to enhance the overall detection accuracy. However, within the meta optimizer lies the problem of prototype deviation. The reason for this problem is the use of an average-based method to roughly estimate the gradient when the labeled samples are limited in each category.

Given the above considerations, in this work, our aim is to address the challenging few-shot traffic sign detection problem, as shown in Figure 1. Specifically, given the problem of unbalanced sample distribution, we aim to obtain a model that can detect both base and novel objects at test time. We believe that this eliminates the impact of distribution shifts between training and testing data, which is vital for improving the detection accuracy and generalization ability of the constructed model. Therefore, we use the method of Random Fourier Features and sample weighting to effectively remove the statistical correlation between relevant and irrelevant features.
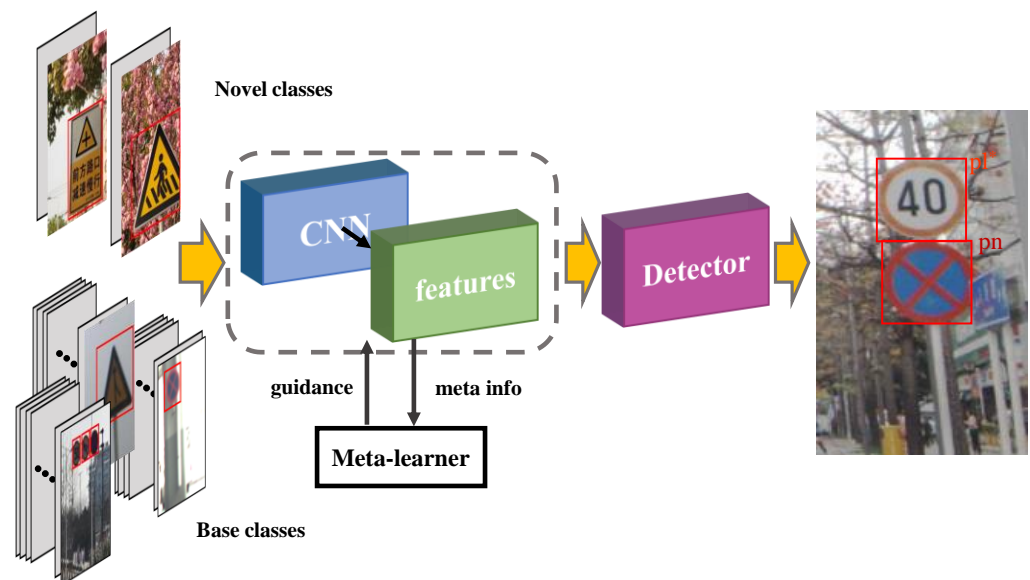
Figure 1. We aim to obtain a few-shot traffic sign detection model by training on the base classes with meta-learner guidance at image level; consequently, the model is able to detect novel objects on testing images with learning from a limited annotated sample.

Traffic sign detection is a key part of autonomous driving technology with good performance for both speed and accuracy. This requires a detector with the ability to detect unseen classes accurately in real time. Few-shot is a feasible approach to the above problem, which can detect new classes with only a small number of labeled samples needing to be trained. The current transfer learning is more effective for solving few-shot problems, but

it needs to be trained on the source domain and then fine-tuned on the target domain with few samples, which is not suitable for dynamic environments and urgent tasks.

Meta-learning provides a new and feasible solution to the above few-shot problem. We designed a traffic sign detector that unifies the few-shot learning ability by two-stage meta-learner ϕ learning class features from base classes at the image-level and a multi-scale ability to predict novel classes in conformity with limited support examples.

The proposed meta-learning FSD framework, shown in Figure 2 contains a meta-learner model F and a detector model ϕ. We extract query features and support features by utilizing the YOLOv4 backbone network, gaining the ability to quickly adjust parameters to be at the optimal level for new tasks with the meta-learner. Here, the learning process exists at two terms: the initialization term and adaptation term, shown in Figure 3, improving a strong discriminative ability of a detector to distinguish different categories from the multi-relation detector module. Our framework boils down exactly to a typical meta-learning paradigm, encouraging the name Meta-YOLO.
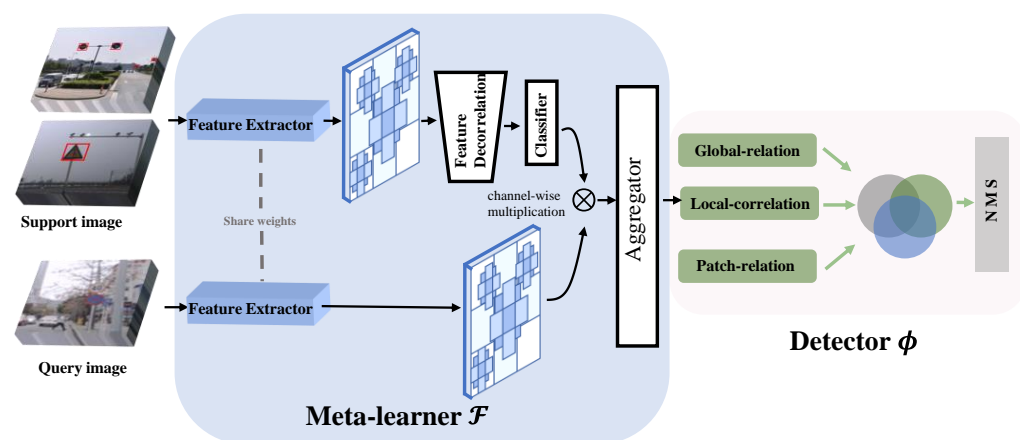
Figure 2. The architecture of our proposed Meta-YOLO. It comprises a meta-learner model and a detector model. The weights of the obtained query features and support features are shared by the feature extractor utilizing the YOLOv4 backbone network. The aggregator outputs the category detection results by combining the query features with the category features associated to the target instance in the support images received by the decorrelation module and the classifier. To perform feature multi-scale detection, a multi-relational detector module with a three-head mechanism is employed.

Our contributions are as follows:

1. We present Meta-YOLO, a novel few-shot traffic sign detection framework that unifies image-level meta-info of object localization and classification into a one-stage module.
2. We design a feature decorrelation module (FDM) that gets rid of spurious correlations and, in turn, focuses more on the real connection between discriminative features and labels. This module can overcome the complex backgrounds' interference in the detection process, thus enhancing the robustness of the system.
3. We introduce a three-head mechanism that allows the detector to jointly attend to information from the spatial position relationship of different levels. The application of this mechanism's advance ability allows a detector to distinguish between different categories as well as to detect different scale targets.

The rest of this paper is organized as follows: in Section 2 we discuss details of the feature decorrelation module, while the architecture and implementation details of our network are presented in Section 3. We give experimental results in Section 4 and conclusions in Section 5.
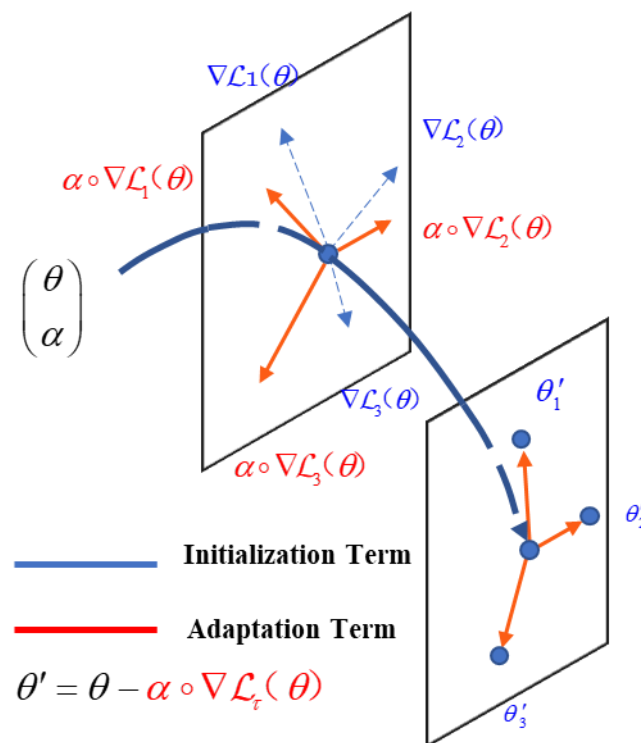
Figure 3. Illustrating the two-stages of meta-learner. Initialization term fuses cross-task information in specific scenarios, at meta-space (θ, α) that learns the meta-learner. Adaptation term is carried out by the meta-learner in the learner space θ that learns task-specific learners.

2. Sample Weighting for Decoupling Dependencies

This section mainly describes the construction method of the feature decorrelation module (FDM). Inspired by [33–36], we propose a method to perceive and decouple global correlations by iteratively saving and reloading the features and sampled weights of the model. The architecture of the feature decorrelation module is shown in Figure 4.
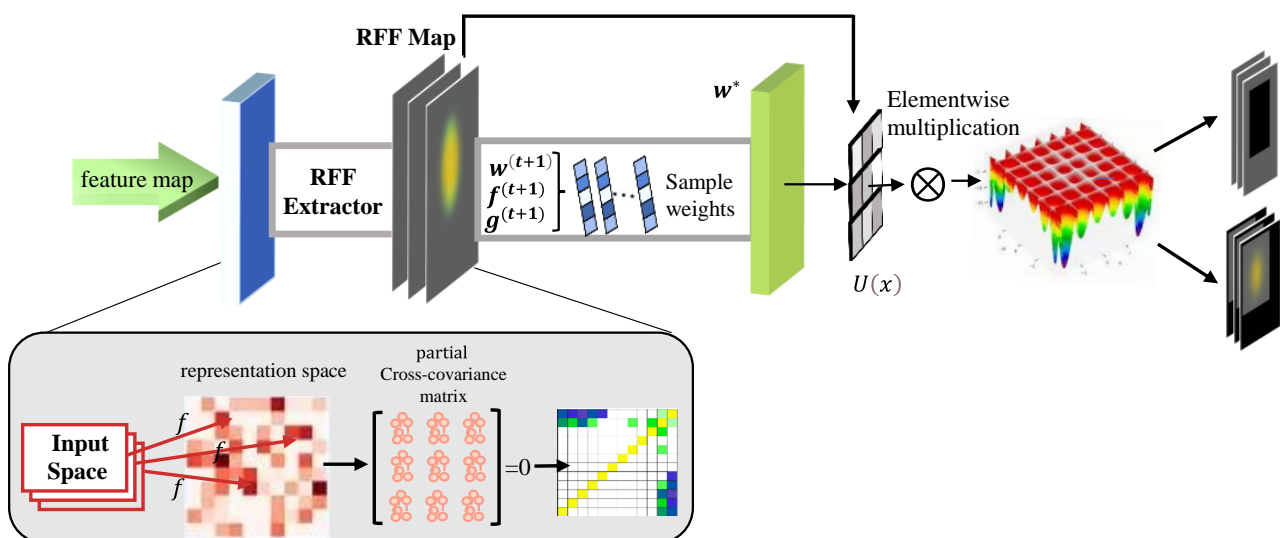
Figure 4. The architecture of feature decorrelation module. The overall process takes place during the training phase and consists of removing all linear and non-linear relationships between features (Section 3) as well as global weight optimization (Section 2).

Notations: $\mathcal{X} \subset \mathbb{R}^{m_X}$ denotes the space of input, $\mathcal{Y} \subset \mathbb{R}^{m_Y}$ denotes the space of outcome space, $\mathcal{Z} \subset \mathbb{R}^{m_Z}$ denotes the space of representation. m_X, m_Y, m_Z are the dimensions of space $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$, respectively. If we suppose that there are two different random variables M, N in the representation space (their sample information and label information are stored in $\mathcal{X}$, $\mathcal{Y}$, respectively), u and v are Random Fourier Features mapping functions. f: $\mathcal{X} \rightarrow \mathcal{Z}$ denotes the representation function and g: $\mathcal{Z} \rightarrow \mathcal{Y}$ denotes the prediction function.

If we suppose that most of the traffic signs are located in front of trees in support sets S_C, then RFF will extract the features of traffic signs and trees, while mapping the original m_X dimension features data extracted in the input space $\mathcal{X}$ to the higher dimensional (m_Z-dimensional) representation space $\mathcal{Z}$ by the representation function f; therefore, the samples are linearly separable in this feature space. Since the datasets do not contain all the features of "trees", we first multiply the difference between the predicted and ground truth with the weights of the features to obtain the optimization target. Then we iteratively change the weights of different features to obtain the features with the correct correlation to traffic signs.

2.1. RFF-Based Feature Decorrelation

According to [33] the independence testing statistic I_{MN} is defined as the Frobenius norm of the partial cross-covariance matrix, i.e., $I_{MN} = \|\hat{\Sigma}_{MN}\|_F^2$. Notice that I_{MN} is always non-negative, and two variables M, N tend to be independent as I_{MN} is equal to zero. Thus, I_{MN} can effectively measure the independence between random variables. Actually, the Frobenius norm corresponds to the Hilbert–Schmidt norm in Euclidean space [37], and, according to the Hilbert–Schmidt norm [38], which needs the squared Hilbert–Schmidt norm of Σ_{MN} to be zero, it can be used as a criterion to supervise feature decorrelation. Thus, based on the Frobenius norm, the partial cross-covariance matrix is as follows

$$\hat{\Sigma}_{MN} = \frac{1}{n-1} \sum_{i=1}^{n} \left[\left(u(M_i) - \frac{1}{n}\sum_{j-1}^{n} u(M_j) \right)^T \left(v(N_i) - \frac{1}{n}\sum_{j-1}^{n} v(N_j) \right) \right] \tag{1}$$

where

$$\begin{aligned} u(M) &= \left(u_1(M),\, u_2(M),\, \dots u_{n_M}(M),\, u_j(M) \right) \in \mathcal{H}_{RFF},\ \forall j; \\ u(N) &= \left(v_1(N),\, v_2(N),\, \dots v_{n_M}(N),\, v_j(N) \right) \in \mathcal{H}_{RFF},\ \forall j. \end{aligned} \tag{2}$$

Here $\mathcal{H}_{RFF}$ denotes the function space of Random Fourier Features. The results are calculated using the partial cross-covariance matrix for random variables M and N after weighting as follows

$$\hat{\Sigma}_{MN;w} = \frac{1}{n-1} \sum_{i=1}^{n} \left[\left(w_i u(M_i) - \frac{1}{n}\sum_{j=1}^{n} w_j u(M_j) \right)^T \left(w_i v(N_i) - \frac{1}{n}\sum_{j=1}^{n} w_j v(N_j) \right) \right] \tag{3}$$

We set the weight optimization method as follows, and this optimization strategy can minimize the dependency between features.

$$w^* = \underset{w \in \Delta n}{argmin} \sum_{1 \le i \le j \le m_Z} \|\hat{\Sigma}_{MN;w}\|_F^2 \tag{4}$$

The iterative equation is as follows

$$^{(t+1)}, g^{(t+1)} = \underset{w \in \Delta n}{argmin} \sum_{i=1}^{n} w_i^{(t)} \mathcal{L}_{FDM}(g((X_i)), \mathcal{Y}_i), \tag{5}$$

$$w^{(t+1)} = \underset{w \in \Delta n}{argmin} \sum_{1 \le i \le j \le m_Z} \|\hat{\Sigma}_{M^{(t+1)}N^{(t+1)};w}\|_F^2. \tag{6}$$

where $\mathcal{L}_{FDM}(.,.)$ represents the cross-entropy loss function and t represents the time stamp.

2.2. Global Optimization

In practice, especially for deep learning tasks, it needs enormous storage and computational cost to learn sample weights globally. Moreover, SGD tends to fall into local minima and saddle points, and the noise from small batch sampling makes the loss oscillate back and forth; hence, global weights for all samples cannot be learned. In this section, supported by the findings in [35], we propose a method to perceive and remove global correlations though iteratively saving and reloading the features and sampled weights of the model.

Our main idea is to maintain the upper bounds on the piecewise linearity of the sample features $Z(x)$ and weights $w(x)$ while using them to determine the estimate of x in each optimization step. Then, the upper bound of piecewise linear function $U(x)$ can be determined by the following function [35]

$$
\begin{aligned}
U_1(x) &= \min\left[Z(x_i) + \sqrt{\sigma_i + (x - x_i)^T K (x - x_i)} \right]; \\
U_2(x) &= \min\left[w(x_i) + \sqrt{\sigma_i + (x - x_i)^T K (x - x_i)} \right]
\end{aligned}
\tag{7}
$$

where σ_i is a noise term and K is a diagonal matrix with k terms of the Lipschitz constant. We point out that we adopt the k-value estimation method of C. Malherbe et al. [36]. This approach makes the algorithm run faster by using a random search of $U(x)$ to check if the upper bound of the new point is better than the existing optimal point and, if so, update it to the new optimal value. Moreover, sample features $Z(x)$ and weights $w(x)$ are updated at the end of each batch and represent the global information of the whole training dataset.

Although the local maximum region can be reached quickly using the above method, it cannot be moved to the optimal position rapidly. In the light of this problem, we introduce the classical confidence domain method of Powell et al. [39] to fit the quadratic surface of the current optimal solution, and then the next iteration to the quadratic surface extreme value point at a certain distance from the current optimal solution.

3. Methodology

Our aim is to solve the problem of few-shot traffic sign detection. We first define the problem and task of FSD under meta-learning. Following recent work, we propose the solution: Meta-YOLO, which is implemented by integrating YOLOv4 into a meta-learning pipeline. The traffic sign detection framework proposed in this paper, works in a meta-learning process, composed of meta-learner and object detector.

3.1. Task Definition

Given two sets of categories, sets C_{base} and C_{novel}, which are mutually disjointed, $C_{base} \cap C_{novel} = \varnothing$; corresponding categories with a base dataset $D_{base} = \{ (\mathcal{X}_i^{base}, \mathcal{Y}_i^{base})_{i=1}^{n_1} \} \sim p(\tau)$ contains abundant annotated instances in each novel class C_{base} instance, and a novel dataset $D_{novel} = \{ (\mathcal{X}_i^{novel}, \mathcal{Y}_i^{novel})_{i=1}^{n_2} \} \sim p(\tau)$ contains very few samples in each novel class. We propose a distribution $p(\tau)$ lies in the relevant task space and sample randomly from this distribution.

Simulating the meta-learning paradigm, meta-training tasks $meta - train(\tau)$ are comprised of support set S_C and query set Q_C, and meta-testing tasks $meta - test(\tau)$ are comprised of support set $S_C{}'$ and query set $Q_C{}'$ to keep consistency.

Arbitrary target categories and datasets corresponding to it are randomly sampled from C_{base}, a part of it as S_C, another part as Q_C. Likewise, arbitrary target categories and datasets corresponding to it are randomly sampled from C_{base}, a part of it as S_C, another part as Q_C. N classes are randomly selected from C_{novel}, K + § samples are randomly selected from D_{novel} corresponding to classes where K samples will be used as $S_C{}'$ and § samples as $Q_C{}'$. The task is to find all the attributes from query support category and label them with tight bounding boxes.

Our goal is to build detection algorithms that work in a meta-learning process that allows a meta-learner to learn how to quickly adjust its parameters to the optimum for a new task in the presence of new classes with very few instances by training on a series of seen classes, thus teaching the detector ϕ to quickly detect unseen targets and output the predicted categories y and locations t. Among others, the meta-learner model should be trained on the distribution $p(\tau)$ according to the principle that the training and testing processes should be consistent.

3.2. Network Description

We adopt the backbone of YOLOv4 (i.e., CSPDarknet53) to implement the feature extractor; support features share all learnable parameters with query feature following the philosophy [40]. We filter out irrelevant information within the support image via FDM whose further details have been shown in Section 2. Meta-YOLO is conceptually simple and aims to quickly and accurately detect unseen traffic signs. The framework is shown in Figure 2, which is mainly composed of two parts: meta-learner model $\mathcal{F}$ and detection model ϕ.

We set a learnable learning rate for all parameters so that the meta-learning system can learn good initialization as well as fast adaptation strategies. Given a query image, first the feature extractor generates its feature map and then adopts 1×1 convolution to make the feature map's channel dimension compatible with the downstream modules. Unlike query feature, which retains image-level information, for support image it is necessary to extract the category features of the certain object instance. Therefore, we utilize Fourier features combined with sample weights to decouple dependencies and remove the irrelevant information present in the support set. The design of aggregator follows previous work [41], reweighting the query features f^{cls} according to the output of the support set class feature f^{qry}. The support class features are combined with query features as in Equation (8) to obtain the detection results of the corresponding category.

$$\mathcal{A}\left(f^{qry}, f^{cls}\right) = f^{qry} \otimes f^{cls} \tag{8}$$

where $\otimes$ represents channel-wise multiplication. The support set is utilized in a loop to construct a classifier for the sample class. The aggregator is then used to cascade the information matching to the query set with the feature relationships and categories corresponding to each image in the support set.

The detector we desire is with a powerful discriminative capacity to distinguish between different categories as well as to detect different scale targets. Based on this, three-heads, the global-relation head, the local-correlation head and patch-relation head, are constructed to learn a deep embedding for global matching, pixel-wise and depth-wise cross-correlation between support and query sets and a deep non-linear metric for patch matching, respectively.

Revisiting Yolov4 backbone. The proposed Meta-YOLO utilizes the successful YOLOv4 for FSD to detect traffic signs, shown in Figure 2. YOLO [5–8] is the representative work of one-stage model; the core idea is to input the complete image into the network and regress the classes and coordinates of the bounding box in the output layer to achieve end-to-end training. YOLOv4 [8] improves on YOLO v3 [7] to sit at the faster end of the speed–accuracy trade-off. Its accuracy is already comparable to or even surpasses two-stage target detection algorithms while maintaining a very high speed. The YOLOv4 backbone uses CSPDarknet53 with the mish activation function to increase the network's learning ability and gradient transmission efficiency; the network's regularization process is upgraded and improved extensively by employing a better Dropblock. CSPDarknet53 is a backbone structure based on the Yolov3 backbone network Darknet53, which draws on the experience of CSPNet 2019. The use of this network structure enhances the learning capability of CNNs, enabling a light weight while sustaining accuracy, decreasing computational bottlenecks

and lowering memory costs. This is of great significance to YOLO, which not only ensures the speed and accuracy of inference but also reduces the size of the model.

This work aims to address the problem of few-shot traffic sign detection under the blessing of meta-learning. Then we evaluate the feasibility of the framework by equipping it with an object detection architecture; to this end, relatively simple and efficient detectors should be selected, i.e., YOLOv4. Meta-YOLO extends the YOLOv4 framework by integrating meta-learning within an end-to-end convolutional neural network-based detection framework. This creative design not only enhances the image feature extraction capability, but also helps to improve the accuracy of one-stage classification on new categories; in fact, the FSD accuracy and speed are greatly improved.

Meta-learner: With only a few labeled instances, it is intractable to decide how to initialize parameters and learning rate as well as when to halt the learning process to shun overfitting. We need to maximize the generalization capability rather than data fitting to determine all the learning factors and eventually bring the model to the point where it can acquire learning in just a few iterations while speeding up learning so that the detector can learn and respond in a rapidly changing environment. Inspired by X. Yang et al. [42] and B. Kang et al. [37], we introduce the following end-to-end meta-learner, which consists of two terms: initialization and adaptation, as shown in Algorithm 1.

$$\theta' = \theta - \alpha \circ \nabla \mathcal{L}_\tau(\theta) \tag{9}$$

where θ and α are parameters of the meta-learner to be learned, and $\circ$ stands for element-wise product. $\alpha \circ \nabla \mathcal{L}_\tau \theta$ is a vector whose direction corresponds to the update direction and whose length corresponds to the learning rate. With some loss function ℓ, $\nabla \mathcal{L}_\tau \theta$ is the gradient of $\mathcal{L}_\tau(\theta)$, where $\mathcal{L}_\tau(\theta)$ consists of a conventional cross-entropy loss ℓ_{cls} for classification, a smoothed-L1 loss ℓ_{reg} for bounding box regression and a cross-entropy loss function for acquiring the relationship between different classes from class features, as shown in Equation (10).

$$\mathcal{L}_\tau \theta = \frac{1}{|\tau|} \sum_{\tau \sim p\tau} \ell_{cls}(\phi_{\theta'}) + \ell_{reg}(\phi_{\theta'}) + \ell_{meta}(\phi_{\theta'}) \tag{10}$$

Our method first initializes the learner parameters with θ then adjusts it to θ' in just one phase at a time in direction $\alpha \circ \nabla \mathcal{L}_\tau(\theta)$ and by adopting a learning rate implicitly implemented in $\alpha \circ \nabla \mathcal{L}_\tau(\theta)$.

With the above definition, the objective of meta-learning is

$$\min_{\theta,\alpha} \sum_{\tau \sim p(\tau)} \mathcal{L}_{meta-test(\tau)}(\theta') = \sum_{\tau \sim p(\tau)} \mathcal{L}_{meta-test(\tau)}\left(\theta - \alpha \circ \nabla \mathcal{L}_{meta-train(\tau)}(\theta)\right) \tag{11}$$

The meta-learner is updated iteratively from random initialization using gradient descent, as shown in Figure 3.

$$\theta^t = \theta^{t-1} - \alpha \nabla \mathcal{L}_\tau\left(\theta^{t-1}\right) \tag{12}$$

Meta-learning generally occurs on batches of tasks where the detector ϕ is trained on $meta - train\ \tau$ and use $s\mathcal{L}_\tau$ as a meta-info to update the meta-learner, which is then tested on $meta - test\ \tau$. The meta-learning process is repeated until the meta-learner is able to learn how to tune the detector to give it the best performance. The improved end-to-end two-stage meta-learner $\mathcal{F}$, which learns not only the initialization of the learner but updates direction and learning rate of the learner, is equipped to have positive performance for a new task with scant instances.

Algorithm 1: Meta-Learner for Meta-YOLO

Input: task distribution $p(\tau)$, meta-learning rate β
Output: detector's parameters θ, detector's learning rate α
1: Initialize θ, α;
2: **While** not end **do**
3:       Sample batch of tasks $\tau \sim p(\tau)$;
4:       **for** all τ **do**
5:               $\mathcal{L}_{meta-train(\tau)}(\theta) = \frac{1}{|meta-train(\tau)|} \sum\limits_{\tau \sim p\tau} \ell_\tau(\phi_\theta)$;
6:               $\theta' = \theta - \alpha \circ \nabla \mathcal{L}_\tau(\theta)$;
7:               $\mathcal{L}_{meta-test(\tau)}(\theta) = \frac{1}{|meta-test(\tau)|} \sum\limits_{\tau \sim p\tau} \ell_\tau(\phi_{\theta'})$;
8:       **end for**
9:       $(\theta, \alpha) \leftarrow (\theta, \alpha) - \beta \nabla_{(\theta, \alpha)} \sum\limits_\tau meta - test(\tau)(\theta')$;
10: **end while**

---

4. Experiments

This section evaluates the effectiveness of Meta-YOLO on few-shot traffic sign detection both qualitatively and quantitatively by testing it with two challenge datasets and comparing with the baselines. Section 4.1 gives a brief introduction to the datasets used in the experiment. Section 4.2 presents the setup of the experiment. The experiment and comparisons of the three baselines and state-of-the-art methods are presented in Section 4.3.

4.1. Datasets

Empirically, three standard TSD benchmark datasets are considered: (1) the GTSDB dataset [43], (2) the TT100K dataset [44] and (3) the MSTD [45] dataset These datasets share a common characteristic in that they cover a variety of fine traffic sign categories, which fit the actual detection demands and the evaluation of our claim on the robustness of few-shot traffic sign detection. The main characteristics of the datasets used in the experiments are shown in Table 1.

Table 1. Main characteristics available in TSD datasets.

	Images	Annotated Signs	B-Boxes	Classes	Annotated Sign Size	Acquisition Location
GTSDB [4]	900	1206	√	43	16–128 longer edge	Germany
TT-100K [40]	100,000	30,000	√	45	2×7 to 397×394	China
MTSD [45]	52,453	80,000	√	313	256 × 256	global

GTSDB The German traffic sign detection benchmark (GTSDB) is a single-image traffic sign detection dataset and is widely used to evaluate traffic sign detectors. This dataset consists of 900 images with 1206 traffic signs that are split into a training set of 600 images with 846 traffic signs and a testing set of 300 images with 360 traffic signs. The resolution of this dataset is 1360 × 800.

TT100K The TT100K dataset is composed of 100,000 images with 30,000 traffic sign instances, which are annotated with a sign type, a pixel map and a bounding box. It consists of 45 categories of a relatively large real-world traffic benchmark. The image resolution in this benchmark is 2048 × 2048 and covers large variations in lighting and weather. This dataset has unbalanced category distribution.

MTSD: The Mapillary Traffic Sign dataset covers multiple locations on six continents and consists of 52,453 high-resolution images with more than 80,000 annotated signs. This dataset includes 313 categories and has variations in weather, season, moment, camera and perspective.

4.2. Experimental Setup

We set an object detection benchmark for traffic sign detection processed in the following way. Out of TT100K's 45 instance categories, we randomly selected nine classes as the novel ones, while keeping the other 36 as the base. During the initialization term of meta-learning, only the base class objects are considered. In the adaptation term, there are K-shot annotated bounding boxes for objects in each novel class and 3K annotated bounding boxes for objects in each base class for training, where K is set at 1, 2, 3, 5 and 10. We adopted the mean Average Precision (mAP) as the evaluation metric and a qualified prediction that ought to be more than 0.5 IoU with the ground truth. To design a few-shot learning setup, we considered three different novel/base class split settings, i.e., ("w1", "w27", "pd", "ip", "i6"/rest); ("ip", "pg", "w27", "w1", "i6"/rest) and ("w1", "p7", "w27", "i6", "pd"/rest).

Similarly, on the GTSDB dataset, we had 300 images from the validation set for evaluation, and the remaining images for training. Out of its 43 instance categories, we randomly selected 10 classes overlapped with TT100K as the novel ones while keeping the remaining 33 as the base. We also considered the proposed model learning on the 43 base classes from GTSDB and exploiting it to detect the two novel objects in TT100K. This setting features a cross-dataset problem that we called GTSDB to TT100K.

We took a similar approach to the MIST dataset as the previous two datasets. We selected 55 classes as the novel classes, while keeping the rest as base classes.

As for the computer platform settings, a standard PC was used for all the experiments, whose hardware and software configuration are listed as follows:

- NVIDIA GeForce RTX 2080Ti GPU
- Dynamic Memory: 128G DDR3 RAM
- Python + Pytorch(GPU)

4.3. Performance

4.3.1. Comparison with Baseline

We compared our model with three competitive baselines. We constructed baselines as follows: **YOLO-joint** jointly trains the YOLOv4 detector on the base and novel classes, and uses the identical number of iterations as Meta-YOLO to train this baseline model. **YOLO-based** takes the same training strategy in Meta-YOLO. **YOLO-ft** applies the same training strategy of ours but trains the detector to fully converge.

TT100K The K-shot traffic sign detection is performed based on K = (1, 2, 3, 5, 10) across three novel/base class splits. As shown in the experimental evaluation results in Table 2, our proposed model significantly outperforms the baseline. It reveals the generalization weakness of the one-stage detector YOLOv4 with a small number of labels in the training term. Comparing the performance results of the YOLO-joint and YOLO-based baselines shows that joint training biases the detection results toward the base class and nothing about the new class. Note that, YOLO-ft performs significantly better than the other two baselines, which proves the necessity of a two-stage strategy for a meta-learner. Furthermore, the attention maps are visualized in Figure 5. The attention maps show that these highlighted regions are almost correlated with traffic sign instances, which indicates that the feature decorrelation module effectively removes irrelevant features and enhances the generalization of meta-learned representations.

GTSDB: We evaluated 10-shot/30-shot setups on the GTSDB [4] benchmark and report the standard GTSDB metrics. The results on the novel classes are shown in Table 3. In both cases, meta-YOLO outperforms the other baselines. The results show that the performance of the baselines in detecting new classes is extremely poor, which is caused by the unbalanced setting of the number of base classes and new classes, further revealing the vulnerability of YOLO to the generalization problem. In Figure 6, some comparisons between YOLO-ft and Meta-YOLO are visualized when detecting novel class traffic signs; the results indicate our method is indeed effective.

Table 2. Few-shot detection mAP on TT100K test set for novel classes. The baselines are evaluated using three separate sets of novel classes.

	Novel Set 1					Novel Set 2					Novel Set 3				
Method/Shot	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
YOLO-joint	0.0	0.8	1.9	4.3	6.2	0.3	0.8	5.5	7.3	12.4	0.0	0.9	2.6	5.3	8.4
YOLO-based	2.4	7.4	11.2	19.0	20.5	3.2	5.8	19.6	27.2	28.6	4.3	5.2	10.8	21.7	22.7
YOLO-ft	10.5	14.6	18.3	28.8	30.2	11.0	18.4	26.7	33.5	35.0	7.4	12.1	20.2	30.6	37.2
Ours	**18.3**	**21.7**	**26.8**	**30.0**	**45.3**	**19.7**	**22.9**	**29.3**	**37.5**	**42.1**	**15.6**	**19.9**	**29.7**	**39.4**	**45.8**
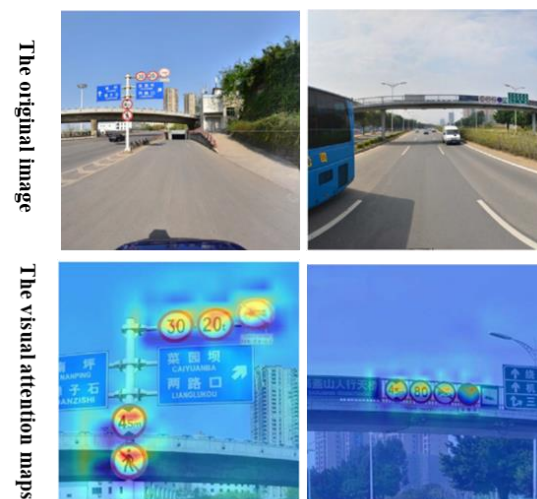
Figure 5. Visualization of attention maps. With feature decorrelation module introduced, clear responses for classes are observed, manifesting this module's effectiveness in promoting generalization of meta-learned representations.

Table 3. Few-shot performance on GTSDB test set for novel classes. For the novel categories, we analyze performance for various numbers of training shots.

Shot	Baselines	Average Precision			Average Recall		
		AP	AP50	AP75	AR1	AR10	AR100
10	YOLO-based	2.4	5.3	1.7	5.3	7.9	7.9
	YOLO-ft	6.2	13.5	5.4	8.0	12.6	12.7
	ours	9.8	18.1	7.2	10.2	17.1	17.2
30	YOLO-based	2.6	5.8	1.9	7.3	9.4	9.4
	YOLO-ft	12.7	18.3	10.2	13.6	20.5	20.7
	ours	17.2	25.4	13.5	15.0	24.4	24.7

GTSDB to TT100K: In this cross-dataset few-shot traffic sign detection setup, all the baselines are trained with 10-shot objects in novel classes on GTSDB, while they are evaluated on the TT100K test set. Distinct from the previous experiments that concentrate on evaluating cross-category model generalization, this setup goes further to manifest the cross-domain generalization ability. The mAP of YOLO-based and YOLO-ft achieve the detection performance of 20.1% and 32.6%, respectively. Instead, Meta-YOLO achieves 39.8%, while this performance is poorer than when using base classes in TT100K (which has a mAP of around 42%).

Figure 6. The visualization of novel class traffic signs by YOLO-ft and Meta-YOLO. Comparing with Meta-YOLO, YOLO-ft is inferior: b-boxes in the first columns are omitted; in the middle column they are wrong; in the last columns they are almost duplicated.

Adaptation speed. Our two-stage meta-learner model $\mathcal{F}$ is able to rapidly learn all parameters and directions during the adaptation term. We use the AP, obtained from Equations (13) and (14), to plot versus the number of training iterations to analyze the adaptation speed of the models. Here we select the TT100K base/novel split three to train under our method and the baseline, respectively, and its AP variations under different iterations as shown in Figure 7. The results show that Meta-YOLO exhibits remarkable fast adaptation ability, shown in Figure 7. It is noted that in the experiments shown in Table 3, YOLO-based and YOLO-ft need 2.5K iterations for them to fully converge; however, ours only requires 0.1K iterations to converge to a higher accuracy.

$$AP = \frac{1}{R_{num}} \sum_{R=\{r_1 \cdots r_{num}\}} \rho_{interp}(r) \tag{13}$$

$$\rho_{interp}(r) = \max_{\tilde{r}:\tilde{r} \geq r} \rho(\tilde{r}) \tag{14}$$

where R_{num} is the number of positive samples, $R = \{r_1 \cdots r_{num}\}$ is the recall of positive samples and $\rho(\tilde{r})$ is the precision corresponding to the $R = \{r_1 \cdots r_{num}\}$.
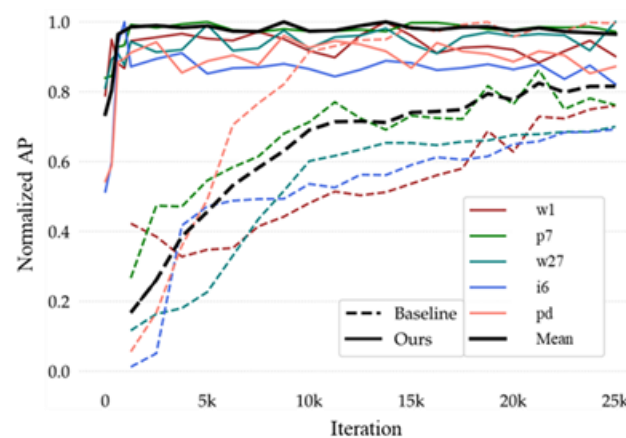
Figure 7. Performance variations under different iterations. We plot the AP against number of training iterations that reflect the adaptation speed of algorithm. Meta-YOLO manifests much faster adaption speed.

4.3.2. Comparison with State-Of-The-Art Methods

In order to ensure the justness of the experiments, in this section, the meta-YOLO proposed by us is compared with state-of-the-art methods by Han et al. [46], Min et al. [16], Zhang et al. [47] and Fan et al. [48]. The experiments were performed based on K=10 across MIST datasets, and the results on mAP50, base and novel classes are shown in Table 4. The results show that our method outperformed most algorithms. The performances of meta-YOLO were 46.2 and 51.7 on mAP50, which are the best for all methods. Fan et al. [48] combined a two-way contrastive training strategy and attention RPN to construct an object detection frame that solved the problem of the poor generalization of few-shot. Although the attention RPN method mitigates the dependence on region proposals to some extent, its framework is still RPN and the detection backgrounds are complex traffic scenes, so its performance is worse than ours with scarce training samples. It is worth noting that the meta-DETR framework proposed by Zhang et al. [47] is on a par with our model, and even better than ours in the novel class. We think this is mainly due to the semantic alignment mechanism (SAM) that, using a residual connection, aligns the high-level and low-level semantics and raises the function of regularization.

Table 4. Comparison of performance with other detectors under 10 shots.

Method/Shot	mAP		mAP$_{base}$		mAP$_{novel}$	
	5-Shot	**10-Shot**	**5-Shot**	**10-Shot**	**5-Shot**	**10-Shot**
Han et al. [46]	29.2	38.9	46.4	56.1	10.8	12.4
Min et al. [16]	32.1	43.4	50.1	58.6	10.4	11.7
Zhang et al. [47]	46.1	51.2	58.4	**68.0**	**37.8**	**42.2**
Fan et al. [48]	46.0	50.2	57.0	65.4	35.8	38.3
Ours	**46.2**	**51.7**	**58.7**	67.9	37.6	41.9

4.4. Ablation Studies

We constructed extensive ablation experiments to research how our proposed individual components contribute to the detection performance. The experiments used the TT100K base/novel split1 with 10-shot data on novel classes.

Effect of Feature Decorrelation Module (FDM). We introduced FDM into the model to utilize RFF to map non-linear indivisible features of raw space to a higher dimensional space, and transform these into linearly separable features, thereby obtaining the real features of the object. As shown in Table 5 FDM is effective in hindering the reliance on category-specific features. Without FDM the method has a strong effect on the performance on novel classes, but is only slightly affected on base classes. With FDM included, the detection performance of the base and novel classes improves, which shows that the more generalizable representations are learned effectively.

Table 5. Ablation studies of several design choices.

Design Choice			Base	Novel
FDM	**Three-Head Mechanism**	$\mathcal{L}_{FDM}$		
√			64.9	43.5
√		√	65.0	44.2
	√	√	65.4	43.1
√	√		66.2	44.7
√	√	√	67.9	45.3

Effect of meta-learner. Since Meta-YOLO is formally designed as a meta-learner, it is crucial to observe whether our method truly improves the performance. To verify our claim, we ablated the initialization term of the meta-learner to observe the object detection performances in the base and novel classes. As illustrated in Table 6, our two-stage strategy significantly boosts the model generalization and learning capability.

Table 6. The ablation of meta-learner module $\mathcal{F}$.

Initialization Term	Adaptation Term	Base	Novel
	$\checkmark$	58.2	21.8
$\checkmark$	$\checkmark$	67.9	45.3

Effect of three-head mechanism. We introduced a three-head mechanism with three different scale modules that complemented each other to predict objects within their respective specified ranges. As shown in Table 6, the three-head mechanism achieves a significant improvement in the detection performance for base classes. With a limited number of base categories, it unavoidably has poor performance on novel classes.

5. Conclusions

In this work, we are devoted to addressing the problem of few-shot traffic sign detection. Firstly, we propose a feature decorrelation module that can remove the statistical correlation between relevant and irrelevant features, exploiting the characteristics of Random Fourier Features and sample weighting. Secondly, we design a few-shot traffic sign detection framework based on meta-learning, Meta-YOLO, based on a network feature decorrelation module (FDM), which can prompt the network to take full advantage of features from different scales and is able to learn the general knowledge and proper fast adaptation strategies with the learnable learning rate set for each parameter. Thirdly, meta-YOLO outperforms the three competitive baselines and improves the mAP of few-shot detection by 39.8%. Comparing with state-of-the-art methods, our performance is also better than most other detectors. The results of meta-YOLO's performance variation under different iterations show that two-stage meta-learner model $\mathcal{F}$ owns the ability to quickly learn parameters. A large number of ablation studies confirm the positive impact of the FDM, meta-learner and three-head mechanism during detection.

We designed a three-head mechanism to obtain the information regarding different categories and levels, but we did not completely integrate the information obtained by different heads. This is the limitation of our work. Perhaps the idea of residual connection is helpful to alleviate this problem and we will continue to improve this problem in the future.

Author Contributions: Conceptualization, X.R. and W.Z.; methodology, X.R.; software, X.R.; validation, X.R., M.W. and C.L.; formal analysis, X.R.; investigation, X.R.; resources, X.R.; data curation, X.W.; writing—original draft preparation, X.R.; writing—review and editing, X.R.; project administration, W.Z.; funding acquisition, X.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by National Natural Science Foundation of China (No. 51805312, 52172388).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to express their appreciation to the developers of Pytorch and OpenCV, the authors of Grad-CAM and dataset provider Shanghai Smart Vehicle Integration Innovation Center Co., Ltd. and Shanghai University of Engineering Science.

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

References

1. Haque, W.A.; Arefin, S.; Shihavuddin, A.S.M.; Hasan, M.A. DeepThin: A novel lightweight CNN architecture for traffic sign recognition without GPU requirements. *Expert Syst. Appl.* **2021**, *168*, 114481. [CrossRef]
2. Zheng, Y.; Bao, H.; Meng, C.; Ma, N. A method of Traffic police detection based on attention mechanism in natural scene. *Neurocomputing* **2021**, *458*, 592. [CrossRef]
3. Li, C.J.; Qu, Z.; Wang, S.Y.; Liu, L. A Method of Cross-layer Fusion Multi-object Detection and Recognition Based on Improved Faster R-CNN Model in Complex Traffic Environment. *Pattern Recognit. Lett.* **2021**, *145*, 127. [CrossRef]
4. Alghmgham, D.A.; Latif, G.; Alghazo, J.; Alzubaidi, L. Autonomous Traffic Sign (ATSR) Detection and Recognition using Deep CNN. *Procedia Comput. Sci.* **2019**, *163*, 266. [CrossRef]
5. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; p. 779.
6. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; p. 6517.
7. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
8. Bochkovskiy, A.; Wang, C.Y.; Liao, H. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
9. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the 2016 European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; p. 21.
10. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; p. 2999.
11. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; p. 580.
12. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; p. 1440.
13. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137. [CrossRef] [PubMed]
14. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2007 IEEE International Conference on Computer Vision (ICCV), Rio De Janeiro, Brazil, 14–21 October 2007; p. 2980.
15. Li, Z.; Chen, M.; He, Y.; Xie, L.; Su, H. An Efficient Framework for Detection and Recognition of Numerical Traffic Signs. In Proceedings of the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 22–27 May 2022; p. 2235.
16. 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**, 1–14. [CrossRef]
17. Cao, J.; Zhang, J.; Jin, X. A Traffic-Sign Detection Algorithm Based on Improved Sparse R-cnn. *IEEE Access* **2021**, *9*, 122774. [CrossRef]
18. 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. [CrossRef]
19. Snell, J.; Swersky, K.; Zemel, R.S. Prototypical networks for few-shot learning. *arXiv* **2017**, arXiv:1703.05175.
20. Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; Wierstra, D. Matching networks for one shot learning. In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; p. 3637.
21. Wang, Y.X.; Girshick, R.; Hebert, M.; Hariharan, B. Low-Shot Learning from Imaginary Data. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; p. 7278.
22. Wang, Y.X.; Hebert, M. Learning to Learn: Model Regression Networks for Easy Small Sample Learning. In Proceedings of the 2016 European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; p. 616.
23. Ravi, S.; Larochelle, H. Optimization as a model for few-shot learning. In Proceedings of the 2017 International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
24. Finn, C.; Abbeel, P.; Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 2017 Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017*; Volume 70, p. 1126.
25. Cai, Q.; Pan, Y.; Yao, T.; Yan, C.; Mei, T. Memory Matching Networks for One-Shot Image Recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; p. 4080.
26. Fort, S. Gaussian Prototypical Networks for Few-Shot Learning on Omniglot. *arXiv* **2017**, arXiv:1708.02735.
27. Mishra, N.; Rohaninejad, M.; Chen, X.; Abbeel, P. A Simple Neural Attentive Meta-Learner. In Proceedings of the 2018 International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
28. Munkhdalai, T.; Yuan, X.; Mehri, S.; Trischler, A. Rapid Adaptation with Conditionally Shifted Neurons. *arXiv* **2018**, arXiv:1712.09926.
29. Zhao, Z.; Tang, P.; Zhao, L.; Zhang, Z. Few-Shot Object Detection of Remote Sensing Images via Two-Stage Fine-Tuning. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 8021805. [CrossRef]

30. Zhang, B.; Li, X.; Ye, Y.; Huang, Z.; Zhang, L. Prototype Completion with Primitive Knowledge for Few-Shot Learning. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; p. 3753.
31. Wang, X.; Cai, Z.; Gao, D.; Vasconcelos, N. Towards Universal Object Detection by Domain Attention. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; p. 7281.
32. Han, G.; Huang, S.; Ma, J.; He, Y.; Chang, S.F. Meta Faster R-CNN: Towards Accurate Few-Shot Object Detection with Attentive Feature Alignment. *arXiv* **2017**, arXiv:2104.07719.
33. Zhang, X.; Cui, P.; Xu, R.; Zhou, L.; He, Y.; Shen, Z. Deep Stable Learning for Out-Of-Distribution Generalization. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; p. 5368.
34. Hua, T.; Wang, W.; Xue, Z.; Ren, S.; Wang, Y.; Zhao, H. On Feature Decorrelation in Self-Supervised Learning. In Proceedings of the 2021 IEEE International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; p. 9598.
35. Malherbe, C.; Vayatis, N. Global optimization of Lipschitz functions. In Proceedings of the 2017 34th International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; Volume 70, p. 2314.
36. Malherbe, C.; Contal, E.; Vayatis, N. A ranking approach to global optimization. In Proceedings of the 2016 33rd International Conference on Machine Learning (ICML), New York, NY, USA, 20–22 June 2016; Volume 48, p. 1539.
37. Strobl, E.; Zhang, K.; Visweswaran, S. Approximate Kernel-based Conditional Independence Tests for Fast Non-Parametric Causal Discovery. *J. Causal Inference* **2019**, 7. [CrossRef]
38. Gretton, A.; Fukumizu, K.; Teo, C.; Song, L.; Schölkopf, B.; Smola, A. A Kernel Statistical Test of Independence. In Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 3–6 December 2007; p. 585.
39. Powell, M.J.D. *The BOBYQA Algorithm for Bound Constrained Optimization without Derivative*; Department of Applied Mathematics and Theoretical Physics, University of Cambridge: Cambridge, UK, 2009.
40. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese Neural Networks for One-shot Image Recognition. *ICML Deep. Learn. Workshop (ICML)* **2015**, 2, 2015.
41. Kang, B.; Liu, Z.; Wang, X.; Yu, F.; Feng, J.; Darrell, T. Few-shot Object Detection via Feature Reweighting. In Proceedings of the 2019 IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; p. 8420.
42. Yang, X.; Marlet, R. Few-Shot Object Detection and Viewpoint Estimation for Objects in the Wild. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; p. 4013.
43. Houben, S.; Stallkamp, J.; Salmen, J.; Schlipsing, M.; Igel, C. Detection of traffic signs in real-world images: The German traffic sign detection benchmark. In Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 4–9 August 2013; p. 1.
44. Zhu, Z.; Liang, D.; Zhang, S.; Huang, X.; Li, B.; Hu, S. Traffic-Sign Detection and Classification in the Wild. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; p. 2110.
45. Ertler, C.; Mislej, J.; Ollmann, T.; Porzi, L.; Neuhold, G.; Kuang, Y. The Mapillary Traffic Sign Dataset for Detection and Classification on a Global Scale. In *ECCV 2020. Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2020; p. 12368.
46. Han, C.; Gao, G.; Zhang, Y. Real-time small traffic sign detection with revised faster-RCNN. *Multimed. Tools Appl.* **2018**, 78, 13263. [CrossRef]
47. Zhang, G.; Luo, Z.; Cui, K.; Lu, S. Meta-DETR: Few-Shot Object Detection via Unified Image-Level Meta-Learning. *arXiv* **2021**, arXiv:2103.11731.
48. Fan, Q.; Zhuo, W.; Tang, C.K.; Tai, Y.W. Few-Shot Object Detection with Attention-RPN and Multi-Relation Detector. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; p. 4012.