IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received June 14, 2021, accepted August 13, 2021, date of publication August 20, 2021, date of current version August 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3106350

Integrated Feature Pyramid Network With Feature Aggregation for Traffic Sign Detection

QING TANG, (Member, IEEE), GE CAO, (Member, IEEE), AND KANG-HYUN JO, (Senior Member, IEEE)

Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 44610, South Korea

Corresponding author: Kang-Hyun Jo (acejo@ulsan.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant through the Korea Government Ministry of Science and ICT (MSIT) under Grant 2020R1A2C2008972.

ABSTRACT Traffic sign detection is a critical task in the visual system of the Advanced Driver Assistance System (ADAS) and the Automated Driving System (ADS). Although the general object detection has achieved promising results by using Feature Pyramid Network (FPN) in recent years, we still observed that FPN cannot obtain satisfactory results in traffic sign detection because the size and class distribution of traffic signs are extremely unbalanced. To overcome this problem, a novel Plug-and-Play neck network Integrated Feature Pyramid Network with Feature Aggregation (IFA-FPN) is proposed in this paper based on the statistical characteristics of traffic signs. First, a lightweight operation is introduced to fully utilize the model and improve the inference speed of the model. Second, an Integrated Operation (IO) is introduced to solve the imbalance problem of Region-of-Interests (RoIs) in pyramid levels. Third, we introduce a Feature Aggregation (FA) structure to strengthen the feature representation capacity of feature maps, thereby enhancing the network robustness against the size discrepancy of traffic signs. The experiments are performed on three mainstream datasets, i.e., the German Traffic Sign Detection Benchmark (GTSDB), Swedish Traffic Sign Dataset (STSD), and Tsinghua-Tencent 100k dataset (TT100k). The experimental results demonstrate the superiority of the proposed IFA-FPN in the traffic sign detection tasks. Specifically, when the proposed IFA-FPN is applied to the Cascade RCNN, it achieves 80.3% mAP in GTSDB which surpasses FPN by 9.9%, 65.2% in mAP in STSD which surpasses FPN by 3.5%, and 93.6% in mAP in TT100k which surpasses FPN by 1.6%.

INDEX TERMS Automated driving system, driver assistance system, feature aggregation, small object detection, traffic sign detection.

I. INTRODUCTION

With the development of the driver-assistance system and autonomous vehicle, the Traffic Sign Detection (TSD) system has been heavily studied over the past decade. A suitable traffic sign detection system helps vehicles perceive the surrounding environment. In the Advanced Driver Assistance System (ADAS), the traffic sign detection system reminds drivers of traffic constraints. In Automated Driving System (ADS), except for perceiving the surrounding environment, the traffic sign detection system can also provide traffic sign location information to the vehicle navigation system. The location information can be used as distinct landmarks for generating High Definition Map (HD Map).

The appearance of traffic signs is designed for attracting human attention easily and quickly. Methods [1]–[6] utilize the appearance characteristics of traffic signs to extract better features in the feature extraction step. These hand-craft features-based methods are not robust enough for distinguishing between real and fake signs in real-world because many objects have similar appearance with traffic signs. It is hard to use the low level hand-craft features to represent the distinguishing characteristics of traffic signs.

Thanks to the development of deep learning algorithms, object detection using Convolution Neural Network (CNN) made remarkable achievements. CNN-based architectures such as Fast R-CNN [7], Faster R-CNN [8], Cascade R-CNN [9], Single Shot multibox Detector (SSD) [10], and YOLO [11] became mainstream detectors that achieve remarkable performance. Different from general objects, traffic signs are relatively small-scale. The scale of most traffic signs in the Swedish Traffic Sign Dataset (STSD) [5], [6] and Tsinghua-Tencent 100k Dataset (TT100k) [12] is less than 100 pixels, shown in Fig. 2. It means that most traffic

The associate editor coordinating the review of this manuscript and approving it for publication was Sudhakar Radhakrishnan.

signs occupy less than 0.8% of an image in STSD, and less than 0.2% in TT100k. Detecting small objects is more challenging than large objects because the CNN extracts features using multi-level convolution and pooling operations to obtain deeper semantic features. Those operations result in that small objects only can exist in the shallow layers but the shallow feature is not powerful enough in complex traffic scenes because of lack of deep semantic information. To obtain deeper semantic features in shallow layers, many works [13]–[18] utilized the Feature Pyramid Network (FPN) [19] or feature fusion architecture to merge deep and shallow feature layers. The original detection models without FPN extract features only using bottom-up pathway C_i, thus the strong semantic features only exist in deep layers C_4 and C_5. FPN merge the feature maps from a top-down pathway P_i and lateral connections L_i to build high-level semantic feature maps P_2-P_5 for the following predictions. Specifically, the shallow feature layers P_2-P_4 contain strong semantics features as deep feature layers C_5, P_5, and P_6.

The FPN [19] is designed for detecting general objects and it achieved promising results in general object detection. However, traffic signs are relatively small-scale and size distribution of them are unbalanced. We observe that the FPN cannot achieve satisfactory performance in traffic signs detection. To design a more suitable neck network for traffic sign detection methods, we analyze the statistical characteristics of traffic signs including the size distribution and the usage of pyramid levels P_2-P_5 in Region-of-Interest (RoI) Alignment step. The architecture of the proposed Integrated Feature Pyramid Network with Feature Aggregation (IFA-FPN) is shown in Fig. 1. The IFA-FPN is designed based on the following three ideas:

1) We found that deep pyramid levels play a minor role in traffic sign detection and therefore we remove deep layers for reducing inference time.
2) We found that dispersedly mapping RoIs into different pyramid levels is unsuitable in traffic sign case because the usage of pyramid levels is extremely unbalanced. Dispersedly mapping RoIs leads to the weak generalization ability of infrequently used pyramid levels. To solve this problem, we proposed an Integrated Operation (IO) to integrating all RoIs into a specific pyramid level. Although some researchers [19] demonstrate dispersed mapping RoIs helps general object detectors, it is noteworthy that we aim to demonstrate integrated mapping RoIs is more suitable for traffic sign detection.
3) In FPN [19], the pyramid level P_2 only need to represent features of traffic signs which size in (0, 112]. In IFA-FPN, the P_2 need to represent features of all size of traffic signs because of the proposed integrated operation. Therefore, enhancing the feature representation capability of P_2 is necessary. Therefore, a Feature Aggregation (FA) structure is introduced to strengthen the feature representation capability of P_2 by aggregating features from different depths so as to enhance the network robustness against size discrepancy.

The contributions of this work are summarized as follows:

1) To overcome the size and class imbalance problem of traffic signs, we proposed an IO which integrates all scale RoIs into a certain pyramid level. To the best of our knowledge, this paper is the first proof that integrated mapping RoIs helps the performance of traffic sign detection.
2) To better represent data with large variance in size, we proposed the FA structure to increase the feature representation capacity of a layer by attaching FA before the layer. The FA structure aggregates multi-scale features to obtain features with high representation capacity. To improve inference speed of it, a Light-FA is further proposed by replacing ensemble-like structure in FA using the residual-shape structure. The ablation experiments show that different paths act as different roles in different size traffic signs.
3) The proposed IFA-FPN is a Plug-and-Play neck network that can be applied in mainstream object detectors to improve performance with similar inference speed. Specifically, IFA-FPN improves performance of Faster RCNN by 14.6% mAP, and improves performance of Cascade-RCNN by 9.9% mAP in the German Traffic Sign Detection Benchmark (GTSDB) dataset [20].
4) Comprehensive experiments have been done to evaluate the performance of the proposed method on three mainstream datasets including GTSDB, STSD [5], [6], and TT100k [12]. The proposed method achieve superior performance on STSD and TT100k dataset. Specifically, the proposed method obtain 80.3% mAP in GTSDB, 65.2% mAP in STSD, and 93.6% mAP in TT100k.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III describes the proposed methods. In section IV, the datasets, evaluation metrics, and the experiments details and analyses are introduced. Finally, Section V concludes this paper.

II. RELATED WORK

Because the appearance of traffic signs is designed for attracting human attention easily and quickly, the traffic signs are designed with regular shapes and high saturation color. The traditional methods [1]–[4] utilized the appearance characteristics of traffic signs to extract better features. Reference [1] utilized the early visual features: red, green, and blue channel of the input images to create three sets of feature maps, i.e., color pairs opponency maps, center-surround differences maps, and local orientation maps, which provide robust features for the subsequent classifier. Based on the feature extraction method in [1], [2] further proposed an enhanced color pairs opponency maps based on categories of traffic signs. After obtaining robust features, the traditional methods applied various classifiers on these features to pursue a robust detector. The traditional methods have two shortcomings, one shortcoming is that hand-craft features are not robust enough for distinguishing traffic signs in the real world.

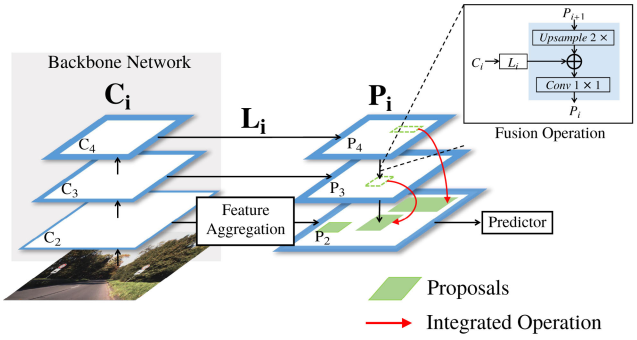


FIGURE 1. The architectures of the proposed IFA-FPN. The blue outlines indicate feature maps. Thicker blue outlines denote semantically stronger features.

Another shortcoming is running a complex feature extractor and classifier is time-consuming.

The deep learning-based traffic sign detectors have made huge progress because they can solve the above two shortcomings well. Deep learning-based methods utilized CNN to extract useful and generalized features autonomously by training CNN on extensive images. Reference [21] first adopted a fully convolutional network (FCN) [22] to obtain potential traffic sign regions and then used CNN to classify the region's class. It achieved good performance, but the computation cost is expensive because of FCN. Lu *et al.* [14] proposed two sub-networks for traffic signs detection. First, some attention regions that are likely to contain traffic signs are obtained by using an Attention Proposal Modeler (APM). Then, it localizes and classifies traffic signs in these attention regions by an Accurate Locator and Recognizer (ALR). The computation cost is low because the high-resolution images are resized to lower resolution images in APM, but the recall accuracy is not satisfactory. Subsequently, for improving the detection performance of small traffic signs, a popular solution is to combine shallow and deep feature maps. Yuan *et al.* [15] proposed a multi-resolution conv-deconv feature fusion network that connects convolution and de-convolution layers to magnify the feature maps and obtain higher semantic features simultaneously. Tian *et al.* [16] proposed a multi-scale recurrent attention network which includes a multi-scale attention module and a recurrent attention module. Same with [15] and [16] obtained the multi-scale feature maps by the de-convolution operation which is time-consuming. Instead of de-convolution operation, Tabernik and Skoaj [18] adopted FPN to generate high-resolution feature maps by up-sampling operations. In the meantime, [18] extended their traffic sign detector with several improvements. The improvements include the data augmentation and Online Hard-Example Mining (OHEM) [23].

Because the size and class imbalance problem in traffic sign datasets are extreme, the above methods [2], [5], [15], [16], [18], [21] did not use the complete dataset to evaluate their methods. References [5], [15], [18], [21] only aimed to detect large-scale and visible traffic signs in a dataset. For example, they only considered the visible traffic signs in STSD with at least 50 × 50 pixels. Reference [16] classified traffic signs into superclasses rather than classes. Specifically, [16] divided traffic signs in GTSDB [20] into four superclasses: prohibitory signs, mandatory signs, danger signs, and others though GTSDB provides 43 classes in total. We consider classify traffic signs into superclasses is impractical in real-world application because traffic signs in same superclass still contain different information, such as "speed limit 20" and "speed limit 80". References [2], [5] only considered six main classes, and [18], [21] considered ten classes in STSD though STSD provides 20 classes in total. These inconsistencies of the evaluation metrics make comparison difficult.

In this paper, based on the distribution characteristics of traffic signs, we propose an IFA-FPN by modifying the existing FPN [19] structure so that IFA-FPN is suitable for extracting traffic signs features. The proposed IFA-FPN is a Plug-and-Play neck network that can be applied in mainstream object detectors to improve performance. To carry out the comprehensive experiments, our proposed IFA-FPN is evaluated on three mainstream traffic sign detection datasets. To compare results in a fair manner, all results are re-performed by open MMLab Detection Toolbox and Benchmark (MMDetection) [13] under the same hardware environment in our local computer. The details of datasets will describe in Section IV.

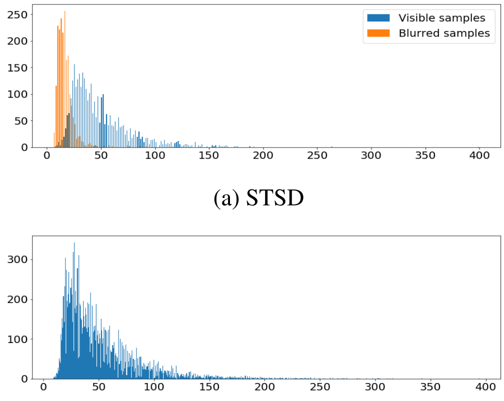
III. PROPOSED METHOD

The proposed method IFA-FPN is designed based on the FPN structure but it solves what the FPN cannot do well in traffic sign detection. In this section, the feature pyramid structure for feature extraction is first described. Then, we introduce the motivation and details of the proposed integrated operation (IO). Finally, three types of multi-scale feature aggregation (FA) structures are described subsequently.

A. STRUCTURE OF FEATURE PYRAMID

The Fig. 1 illustrates the architecture of our proposed IFA-FPN. The IFA-FPN uses the bottom-up pathway C_i, the top-down pathway P_i, and lateral connections L_i to build pyramid levels. The bottom-up pathway C_i is the feed-forward computation of the output of i-th stage blocks of backbone network, such as, ResNet-50, ResNet-101, and ResNext-50, etc. The size of feature map C_i is 2^{-i} times of the size of input images. $i = \{0, \ldots, I\}$ indicates the i-th stage blocks of backbone network, where $i = 0$ indicate the input image, and $I = 4$ is total numbers of stages are used in IFA-FPN. Previous detectors Fast RCNN [7] and Faster RCNN [8] without FPN only use features on the bottom-up pathway C_i to predict class. They cannot achieve satisfied performance because the semantic features in shallow layers of C_i are weak.

To enhance network performance, the top-down pathway P_i and lateral connections L_i are built to generate high-level semantic feature maps. Then, prediction is performed on the generated high-level semantic feature maps P_i. With the help



(a) STSD

(b) TT100k

FIGURE 2. The size distribution of traffic signs in (a) STSD dataset. The orange bars are the blurred samples, and the blue bars are the visible samples, and (b) TT100k dataset. The horizontal axis is the size (in pixel) of traffic signs. The vertical axis is the number of traffic signs.

of P_i and L_i, the shallow feature layers P_2 and P_3 contain strong semantics features as the deep feature layers P_4 as shown in Fig. 1. The top-down feature P_i is computed by P_{i+1} and C_i as follows,

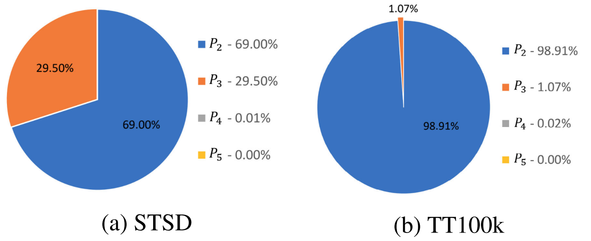
$$P_i = \begin{cases} F_i(P_{i+1}, L_i(C_i)), & i < I \\ L_i(C_i), & i = I \end{cases} \tag{1}$$

where F_i is the i-th fusion operation which includes three steps in detail, shown in the top right of Fig. 1. The first step is up-sampling P_{i+1}. Then, up-sampled P_{i+1} and lateral information are merged by element-wise addition. The third step is to process the merged feature maps by a 1×1 convolution layer. P_{I+1} is a stride two down-sampling of P_I. The L_i denotes the i-th lateral connections. L_i is denoted as follows,

$$L_i : \begin{cases} \mathrm{FA}(\cdot), & i = 2 \\ \mathrm{Conv1} \times 1(\cdot), & otherwise \end{cases} \tag{2}$$

where Conv $1 \times 1(\cdot)$ indicates a 1×1 convolution layer. FA$(\cdot)$ denotes the feature aggregation module which will described in Sec III.C.

Compared with the original FPN, the deep pyramid levels C_5, P_5, and P_6 in the FPN are removed in our proposed IFA-FPN to fully utilize the model and improve the inference speed of the model. There are two reasons. One is deep pyramid levels play a minor role in feature extraction step. Deeper features cannot provide more accurate and useful information of traffic signs because traffic signs are relatively small-scale, as shown in Fig. 2. Another is deep pyramid levels also play a minor role in RoI Alignment step. The usage of pyramid levels P_2-P_5 in RoI Alignment step of original FPN are reported in Fig. 3. In STSD, 0.01% RoIs are mapped into pyramid levels P_4 and P_5. In TT100k, 0.02% RoIs are mapped into pyramid levels P_4 and P_5. Considering the trade-off between accuracy and efficiency, the deep pyramid levels are not used in our proposed IFA-FPN.



(a) STSD

(b) TT100k

FIGURE 3. The usage of pyramid levels P_2-P_5 in region-of-interest (RoI) alignment of original FPN [19]. (a) STSD dataset. (b) TT100k dataset.

B. THE INTEGRATED OPERATION (IO)

In FPN, the RoI Alignment is performed on different pyramid levels based on the scale of RoI. an RoI is assigned to the feature pyramid level $P_{\{i|i=k\}}$ according its scale s by:

$$k = \begin{cases} k_0, & s < 112 \\ \left\lfloor k_0 + log_2 \frac{s}{56} \right\rfloor, & otherwise \end{cases} \tag{3}$$

$$s = \sqrt{wh} \tag{4}$$

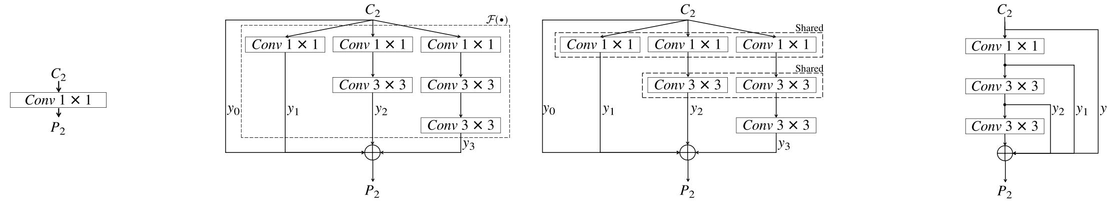
where k_0 is the bottom level to map the RoI, default as $k_0 = 2$ in FPN [19]. Specifically, when a RoI with $s < 112$, $P_{\{i|i=2\}}$ is the target level to map the RoI; when a RoI with $112 \leq s < 224$, the RoI will be mapped into a low-resolution pyramid level $P_{\{i|i=3\}}$. w and h is the width and height of RoIs corresponding to the input image.

The FPN achieved promising results in general objects detection, but it cannot achieve satisfactory performance in traffic sign detection. It is because mapping RoIs into features pyramid dispersedly is unsuitable in traffic sign detection task. To solve this problem, the Integrated Operation (IO) is proposed to integrate all RoIs from different feature pyramid levels to a certain high-resolution pyramid level P_2. The usage of pyramid levels P_2-P_5 in RoI Alignment step are reported in Fig. 3.

There are two advantages of the IO module for detecting traffic signs. One is that integrating large RoIs into P_2 improves generalization ability of P_2. Compared with small-scale traffic signs, large-scale traffic signs provide better features. The distribution of the quality of traffic signs in STSD is reported in Fig. 2(a) which shows small traffic signs contain many blurred signs, and the quality of most large-scale traffic signs are visible. Also, during the feature extraction step, the small traffic signs lose information easier than large traffic signs with the increasing depth due to the max-pooling operator. Thus, integrating RoIs of large traffic signs into P_2 can providing more accurate and useful information, thereby improving small traffic sign detection performance. Another advantage is that IO eliminates the impact of low generalization ability of P_3. Fig. 3 indicates that small part of RoIs are mapped on P_3 which leads to the weak prediction ability of P_3 because of lack of training samples.

C. THE LIGHT MULTI-SCALE FEATURE AGGREGATION (FA) STRUCTURE

To make the IFA-FPN work as expected, enhancing the feature representation capability of P_2 is necessary. Because not



(a) Baseline (b) Full-FA (c) Shared-FA (d) Light-FA

FIGURE 4. The structures of (a) original lateral connection L_i and (b-d) three types of feature aggregation modules.

only small RoIs ($s < 112$) are mapped to P_2, but also the large RoIs ($s \geq 112$) need to assigned to P_2 in IFA-FPN. To help P_2 better represent data which has large variance in size, we convert lateral connection layer L_2 to the proposed feature aggregation structure, which aggregate multi-scale features from different convolution layers path. Three types of feature aggregation structures, including Full-FA, Shared-FA, and Light-FA, will be introduced step by step.

The baseline lateral connection used in FPN is illustrated in Fig. 4(a), which is a 1 × 1 convolution layer. Instead of only using one 1 × 1 layer in FPN, the proposed Full-FA is designed as residual-shape which contain an identity function path y_0 to learn simple features and a mapping function $\mathcal{F}(\cdot)$ to learn complex features. As shown in Fig. 4 (b), given the feature maps C_2, the Full-FA aims to learn a residual $\widetilde{C_2}$ with a mapping function $\mathcal{F}(\cdot)$ as follows,

$$P_2 = C_2 + \widetilde{C_2} \quad \text{s.t.} \quad \widetilde{C_2} = \mathcal{F}(C_2) \tag{5}$$

where $\mathcal{F}(\cdot)$ including several convolution operations. Moreover, to achieve multi-scale feature learning, we design the function $\mathcal{F}(\cdot)$ as a multi-stream building block, which consists of multiple convolution streams y_1, y_2, y_3. The first stream contains a 1 × 1 layer that learns single-scale features (scale = 1). The second stream consists of one 1 × 1 layer and one 3 × 3 layer to learn larger scale features (scale = 3). The third stream further increases receptive field (scale = 5) by adding a 3 × 3 layer. Then, the element-wise summation is applied to aggregate the C_2, y_1, y_2, and y_3 to obtain multi-scales feature map P_2.

Then, we introduce the Share-FA and Light-FA that illustrated in Fig. 4(c) and Fig. 4(d). Veit *et al.* [24] reveal that the paths in residual networks show ensemble-like behavior, and they do not strongly depend on each other. In other words, the residual-shape architecture can be seen as a collection of different paths. Inspired by [24], we further propose a Light-FA to reduce the inference time of model because the Full-FA is heavy and time-consuming. The Light-FA is designed to a residual-shape to represent ensemble-like behaviors of Full-FA. The structure of Light-FA is shown in Fig. 4(d). The Light-FA is equivalent to the Full-FA if the convolutional kernels in Full-FA share weights as Fig. 4 (c). The Shared-FA can be represented as follows,

$$P_2 = C_2 + y1 + y2 + y3 \tag{6}$$

$$\begin{cases} y_1 = f_1(C_2) \\ y_2 = f_2(f_1(C_2)) \\ y_3 = f_3(f_2(f_1(C_2))) \end{cases} \tag{7}$$

where f_1 denotes the first 1 × 1 convolutional layer, and f_2 denotes the 3 × 3 convolutional layer, and f_3 denote the another 3 × 3 convolutional layer. The Light-FA can be represented as follows,

$$P_2 = C_2 + f_1(C_2) + f_2(f_1(C_2)) + f_3(f_2(f_1(C_2))) \tag{8}$$

From Eq.(5), Eq.(6) and Eq.(7), it is clear that Light-FA is equivalent to the Shared-FA in convolution operations. Note that the ReLU-activated layers after each convolutional layer are ignored in notation in Fig. 4. Finally, the Light-FA are built to obtain P_2 with high representation capacity. In this paper, the FA is defaulted to Light-FA.

IV. EXPERIMENTS

A. DATASETS AND EVALUATION METRICS

To carry out the comprehensive experiments, the proposed IFA-FPN is evaluated on three mainstream traffic sign detection datasets: GTSDB [20], STSD [5], [6] and TT100k [12]. The detailed information of datasets is reported in Table 1.

GTSDB: The German Traffic Sign Detection Benchmark provides 600 images and 815 traffic signs for training, and 300 images and 353 traffic signs for testing. The image size in GTSDB is 1360 × 800. There are 43 classes in total.

STSD: The Swedish Traffic Sign Dataset [5], [6] provides 6617 images and 6651 labeled traffic signs in total. It contains two sets (set1 and set2) of images with resolution 1280 × 960 that were captured from Swedish highways and city roads. Each set contains 5 parts and has 20% labeled images. Set1Part0 is used for training, and Set2Part0 is used for testing in this paper. There are 20 classes in total.

TT100k: The TT100k is a large-scale traffic sign detection dataset released by Tsinghua University and Tencent Corporation. Compared with GTSDB and STSD, TT100k provides a large number of images with high resolution 2048 × 2048. The number of traffic signs in TT100k is 19.9 and 3.5 times much than in GTSDB and STSD, respectively.

Same to COCO [25], the scale s of objects is used to separate different size groups: Small ($s < 32$), Medium ($32 \leq s < 96$), Large ($96 \leq s < +\infty$) to report the detection result, thereby analyzing the impact of IFA-FPN in different

TABLE 1. The number of traffic sign of different size group in each dataset.

Size Group / Dataset		Image size	Small: [0,32)	Medium: [32,96)	Large: [112,+∞)	All
GTSDB	Train	1360 × 800	302	479	34	815
	Test		114	230	9	353
STSD	Train	1280 × 960	1805	1228	135	3169
	Test		2327	1027	128	3482
TT100k	Train	2048 × 2048	5462	8599	1337	15487
	Test		2770	4161	709	7706

TABLE 2. Ablation study on individual components of IFA-FPN. Baseline: Detector adopted the FPN as the neck network. S: Small size group. M: Medium size group. L: Large size group. FA: defaulted as Light-FA.

Methods	GTSDB					STSD					TT100k				
	S	M	L	All	FPS	S	M	L	All	FPS	S	M	L	All	FPS
Faster RCNN (Baseline)	71.1	67.2	100.0	63.4	12.2	37.5	86.5	93.3	55.4	11.9	74.7	95.6	96.1	89.9	2.7
+ IO	81.6	73.7	100.0	68.7	12.6	40.7	**88.6**	**94.5**	58.2	11.5	73.5	95.7	**96.7**	89.8	2.8
+ IO + FA	**86.5**	**80.5**	**100.0**	**78.0**	11.0	**44.7**	84.1	94.4	**60.2**	10.0	**79.2**	**95.9**	95.9	**91.3**	2.4
Cascade RCNN (Baseline)	73.8	74.4	100.0	70.4	9.3	43.8	88.3	97.2	61.7	9.6	79.6	96.2	95.7	92.0	2.5
+ IO	84.2	78.9	100.0	75.2	9.8	45.2	**94.2**	**97.8**	64.3	9.7	80.4	96.5	96.8	92.3	2.6
+ IO + FA	**86.6**	**85.2**	**100.0**	**80.3**	9.2	**48.6**	94.1	97.3	**65.2**	8.3	**84.6**	**96.7**	**97.0**	**93.6**	2.3

TABLE 3. Ablation study on removing different pyramid levels *i* in FPN on TT100k. The performance in mAP (%) is reported. S: Small size group. M: Medium size group. L: Large size group.

Used Pyramid layers	TT100k				
	S	M	L	ALL	FPS
$C_{2,3,4,5} + P_{2,3,4,5,6}$	79.6	96.2	**95.7**	92.0	2.5
$C_{2,3,4} + P_{2,3,4}$	**81.0**	**96.3**	95.6	**92.2**	2.6
$C_{2,3} + P_{2,3}$	79.9	96.2	94.9	91.5	2.9
$C_2 + P_2$	69.9	91.9	30.6	78.6	**3.0**

scales. s is computed as,

$$s = \sqrt{w^2 + h^2} \tag{9}$$

where w and h are the width and height of a traffic sign, respectively.

The mean average precision(mAP), which is commonly used as the evaluation criteria in object detection dataset [25], [26], is used as the evaluation measurement in this paper. Average Precision (AP) is the area under the precision-recall curve, which reliably describes the trade-off between the precision and the recall. AP is calculated for one class object, and mAP is the average value of AP over all considered classes. In this paper, a fixed Intersection-over-Union (IoU) with value 0.5 is used for computing mAP.

B. IMPLEMENTATION DETAIL

All experiments have been tested on a desktop with Intel Core i5-6600 3.30-GHz CPU and 1 NVIDIA GeForce Titan 1080Ti GPU with 11 GB memory. MMDetection [13] is used to implement the experiments and evaluate the results. We follow the default pre-processing techniques and hyper-parameters in MMDetection to perform all experiments on three dataset, i.e., data augmentation techniques and anchor setting. The backbone network, default as ResNet-50 [27], is pre-trained on ImageNet [29] to extract feature. The Stochastic Gradient Descent (SGD) optimization algorithm with 0.9 momentum is employed. For GTSDB and STSD, the network is trained in 20 epochs, and the learning rate is 0.01 for the first 15 epochs and 0.001 for the following epochs. The training batch size is 2 in GTSDB and STSD. For TT100k, the network is trained in 10 epoch. The initial learning rate is 0.001, which decreases to 0.0001 at the 8th epoch. The training batch size is 1 in TT100k because of the limited memory of the GPU.

C. ABLATION STUDY

In this section, we first analyze the influence of individual components IO and FA of IFA-FPN by mAP and Frames Per Second (FPS). Then, the ablation study of the effect of removing more pyramid feature layers are performed. After that, the comparison results of three types FA structure including Full-FA, Shared-FA and Light-FA are reported in Table 4 by mAP, FPS and GPU memory usage. Subsequently, the effect of each skip connection in FA are reported. At last, we report the performance of IFA-FPN with different backbone network.

1) THE EFFECT OF INDIVIDUAL COMPONENTS

We evaluate the performance and efficiency of individual components, including IO and FA, by mAP and FPS in GTSDB, STSD, and TT100k dataset, the results are summarized in Table 2. The top part of Table 2 performs the experiments using Faster RCNN as the detector, and the bottom part reports the results of Cascade RCNN. We show the effect of the IO and FA by adding them into the baseline model one by one. Baseline denotes that the detector adopted the original FPN [19] as the neck network. The performance of baseline is not satisfactory, especially in small and medium traffic sign detection. After integrating all scale RoIs into a certain pyramid level by IO, the performance of Faster RCNN and Cascade RCNN are enhanced remarkably, especially in small and medium traffic signs. All methods achieve 100.0% mAP in large size group of GTSDB because of the limited number

TABLE 4. The comparison results of using different FA structures in lateral connection L_2. 'w/o FA' is the baseline result.

Detectors	Lateral Connection	GTSDB						STSD					
		S	M	L	All	FPS	GPU Mem	S	M	L	All	FPS	GPU Mem
Faster RCNN	w/o FA	81.6	73.7	100.0	68.7	12.6	3546M	40.7	**88.6**	94.5	58.2	11.5	3908M
	w/ Full-FA	81.0	79.2	100.0	75.3	9.9	4644M	43.1	86.3	**96.1**	**60.6**	8.8	4974M
	w/ Shared-FA	78.0	77.8	100.0	73.4	9.9	4634M	42.8	84.6	92.3	58.9	8.9	4974M
	w/ Light-FA	**86.5**	**80.5**	**100.0**	**78.0**	11.0	4233M	**44.7**	84.1	94.4	60.2	10.0	4670M
Cascade RCNN	w/o FA	84.2	78.9	100.0	75.2	9.8	3869M	45.2	94.2	**97.8**	64.3	9.7	4227M
	w/ Full-FA	85.3	80.7	100.0	76.8	8.4	4962M	46.8	92.5	97.6	64.5	7.6	5448M
	w/ Shared-FA	85.6	82.6	100.0	78.6	8.4	4953M	47.2	**94.3**	**97.8**	64.9	7.5	5440M
	w/ Light-FA	**86.6**	**85.2**	**100.0**	**80.3**	9.2	4554M	**48.6**	94.1	97.3	**65.2**	8.3	4994M

TABLE 5. Ablation experiments of removing certain skip connection in test step. All use: our proposed Light FA, all skip connections are used. Del y_i: the skip connection y_i is deleted while other skip connections are keep. Top part reports the results of Faster RCNN, and bottom part reports the results of Cascade RCNN.

	Small	Medium	Large	All
All used	44.7	84.1	94.4	60.2
Del y_0	41.6 (-3.1)	84.0 (-0.1)	92.9 (-1.5)	58.1 (-2.1)
Del y_1	42.9 (-1.8)	84.0 (-0.1)	93.5 (-0.9)	58.8 (-0.4)
Del y_2	35.0 (-9.7)	82.5 (-1.6)	93.5 (-0.9)	51.7 (-8.5)
All used	48.6	94.1	97.3	65.2
Del y_0	45.8 (-2.8)	94.2 (-0.1)	96.2 (-1.1)	62.5 (-2.7)
Del y_1	45.6 (-3.0)	92.3 (-1.8)	94.5 (-2.8)	61.6 (-3.6)
Del y_2	30.8 (-17.8)	81.3 (-12.8)	91.2 (-6.1)	47.9 (-17.3)

of traffic signs shown in Table 1. IO also improves the performance of large traffic signs in STSD and TT100k, which indicates that IO can bring stable performance enhancements. These results show that the behaviors of IO are consistent with our intuition mentioned in Sec III.B. Moreover, the inference speed of detectors with IO are faster than before because of removing the deep pyramid levels C_5, C_6, and P_6. These results demonstrate the effectiveness of our proposed IO.

We also evaluate the effect of FA. As reported in Table 2, it is clear that FA further improves the detector performance by large margins. Specifically, "Faster RCNN + IO + FA" improve the mAP from 68.7% to 78.0% and 58.2% to 60.2% in GTSDB and STSD, respectively. The IO and FA bring more substantial performance gains in small datasets (GTSDB and STSD) than big dataset (TT100k) because the size and class imbalance problem is more obvious in small datasets. Experimental results show that each component in our method boost performance, and the combination of them achieves the best performance.

2) THE EFFECT OF REMOVING DIFFERENT PYRAMID FEATURE LAYERS

As mentioned in Sec III.A, the deep pyramid feature layers in the original FPN [19] are removed in our proposed IFA-FPN to reduce the model inference time. We perform the ablation experiments to investigate the influence of removing more pyramid levels. The results are reported in Table 3. In Table 3, The detector is Cascade RCNN [9], and the backbone network is ResNet-50 [27], the IO and FA modules are not applied. The original original FPN used full pyramid layers $C_{2,3,4,5} + P_{2,3,4,5,6}$. When more deep pyramid feature layers are dropped, the inference speeds of models are improved. The best performance is achieved when pyramid feature layers $C_{2,3,4} + P_{2,3,4}$ are used. Dropping more pyramid feature layers $C_{3,4}$ and $P_{3,4}$ accelerate the inference speed but they cannot further improve the model performance. It is clear that the performance significantly declined when we only use $C_2 + P_2$. Considering the trade-off between accuracy and efficiency, only the C_5, P_5, and P_6 are removed in our proposed IFA-FPN finally.

3) THE ANALYSIS OF FA STRUCTURE

Table 4 evaluates the architectural design choices of FA that are shown in Fig. 4. The 'w/o FA' in Table 4 denotes the baseline module that adopts 1 × 1 convolution layer as the lateral connection L_2 illustrated in Fig. 4(a). It is clear that all types of FA can consistently improve the results. It indicates the necessity of enhancing the feature representation capability of P_2 and the validity of the proposed FA structure. Among three FA structures, the primary structure is Light-FA. There are two reasons. One is that two detectors with Light-FA can achieve the best performance in all datasets. Another is that Light-FA reduces the model inference time and the model size. Specifically, Cascade RCNN with Full-FA occupies 4962MB GPU memory, and Cascade RCNN with Light-FA occupies 4554MB GPU memory. The experiments demonstrate the superior performance and efficiency of the proposed FA.

Furthermore, we investigate the effect of skip connections in Light-FA by removing one of them from the trained model in testing step. The effects of skip connections are observed by mAP and its fluctuation in STSD shown in Table 5. We start from the baseline results that shown in the first row of detectors as "All used" and progressively measure the impact of removing each skip connection y_0, y_1, and y_2. The fluctuation of mAP of Faster RCNN and Cascade RCNN are consistent. The performance of small and medium group size is greatly affected by removing the certain skip connection, while the performance of large group size has little effect. This indicates that skip connections play important roles in small and medium traffic sign detection, especially in small traffic signs. The results demonstrate the validity of the proposed FA structure in multi-scale feature learning.

4) ANALYSIS OF SCALABILITY OF IFA-FPN

Table 6 demonstrates that our proposed IFA-FPN can be applied in mainstream object detectors with different

TABLE 6. Ablation experiments with different detectors, neck networks, and backbone networks.

Detector	Neck	Backbone	GTSDB				STSD				TT100k			
			S	M	L	All	S	M	L	All	S	M	L	All
Faster RCNN [8]	BFP [30]	ResNet-50	72.6	69.9	100.0	64.4	41.9	91.1	88.1	58.2	-	-	-	-
	CARAFE [31]	ResNet-50	74.6	64.9	100.0	60.7	41.3	94.6	95.7	59.2	74.9	95.9	93.6	89.9
	FPN [19]	ResNet-50	71.1	67.2	100.0	63.4	37.5	86.5	93.3	55.4	74.7	95.6	96.1	90.0
		ResNext-50	78.9	78.8	100.0	74.8	42.0	86.9	96.0	59.9	78.0	96.0	95.7	91.1
		ResNet-101	66.5	64.4	100.0	59.9	37.4	87.7	95.6	56.5	73.8	95.6	96.4	89.5
		ResNext-101	77.5	71.1	100.0	67.1	40.8	93.4	96.0	58.9	-	-	-	-
	IFA-FPN	ResNet-50	86.5	80.5	100.0	78.0	**44.7**	84.1	94.4	60.2	79.2	95.9	95.9	91.3
		ResNext-50	**86.5**	**85.1**	100.0	**81.1**	44.0	87.4	95.4	**61.0**	**80.9**	**96.5**	94.8	**92.2**
		ResNet-101	80.1	82.2	100.0	78.1	39.4	**91.5**	**95.9**	57.5	77.4	95.5	**96.6**	90.5
		ResNext-101	88.1	81.4	100.0	77.7	41.0	91.3	97.4	59.8	-	-	-	-
Cascade-RCNN [9]	FPN [19]	ResNet-50	73.8	74.4	100.0	70.4	43.8	88.3	97.2	61.7	79.6	96.2	95.7	92.0
		ResNext-50	84.4	79.7	100.0	75.1	47.5	93.9	97.5	65.7	82.9	96.6	96.8	93.1
		ResNet-101	77.0	76.7	100.0	70.9	45.1	93.2	97.2	63.0	-	-	-	-
		ResNext-101	83.1	80.3	100.0	75.9	46.7	91.5	97.3	64.6	-	-	-	-
	IFA-FPN	ResNet-50	**86.6**	85.2	100.0	80.3	48.6	94.1	97.3	65.2	84.6	96.7	**97.0**	93.6
		ResNext-50	85.0	**88.5**	100.0	**84.4**	**50.7**	**95.0**	97.5	**67.9**	**86.4**	**97.3**	97.0	**94.5**
		ResNet-101	81.6	83.0	100.0	77.6	46.2	93.5	97.2	63.9	-	-	-	-
		ResNext-101	84.9	88.3	100.0	83.7	50.4	94.3	**97.6**	67.6	-	-	-	-

TABLE 7. Comparison experiments with feature-based detectors on GTSDB† and STSD†. P.: Prohibitory (%), M.: Mandatory (%), D.: Danger (%). Rec.: Recall (%), Prec.: Precision (%), F1.: F1-measure (%).

Dataset	Feature-based Methods	P.	M.	D.
GTSDB†	wgy@HIT501 [4]	99.9	**99.9**	99.9
	AdaBoost + SVR [2]	**100.0**	**99.9**	**100.0**
	Ours	**100.0**	99.0	98.3
Dataset	Feature-based Methods	Rec.	Prec.	F1
STSD†	Fourier [5]	77.0	91.4	84.3
	AdaBoost + SVR [2]	80.8	**94.5**	87.1
	Ours	**97.2**	87.7	**92.4**

TABLE 8. Comparison experiments with detectors on GTSDB in Rec.: Recall (%), Prec.: Precision (%), F1.: F1-measure (%).

Methods	(%)	S	M	L
SSD512 [10]	Rec.	43.3	80.0	78.8
	Prec.	41.6	76.0	69.3
	F1	42.3	77.9	73.7
MF-SSD [32]	Rec.	45.6	79.2	88.1
	Prec.	28.8	67.5	82.6
	F1	35.9	73.7	85.8
Faster RCNN [8]	Rec.	87.3	77.0	100.0
	Prec.	74.0	66.7	100.0
	F1	80.1	71.4	100.0
Cascade-RCNN [9]	Rec.	87.2	84.3	100.0
	Prec.	84.1	73.4	100.0
	F1	85.6	78.4	100.0
Faster RCNN + IFA-FPN (ours)	Rec.	**96.3**	84.2	100.0
	Prec.	85.1	79.9	100.0
	F1	**90.3**	82.0	100.0
Cascade-RCNN + IFA-FPN (ours)	Rec.	89.2	**87.3**	100.0
	Prec.	**88.9**	**84.1**	100.0
	F1	89.0	**85.6**	100.0

backbone networks to improve performance of them consistently with similar inference speed. We perform the comparison experiments between the FPN and our proposed IFA-FPN with the different backbone network including ResNet-50, ResNext-50, ResNet-101, and ResNext-101 [27], [28]. It is clear that IFA-FPN brings significant improvement over FPN in all backbone network cases, which are consistent with the ResNet-50 results in Table 2. The experiments demonstrate IFA-FPN has good scalability with other detectors and backbone networks, which can be considered as a Plug-and-Play neck network.

To further demonstrate the superiority of our proposed neck network IFA-FPN, we compare IFA-FPN with two state-of-the-art neck networks i.e., Balanced Feature Pyramid (BFP) [30] and Content-Aware ReAssembly of FEatures (CARAFE) [31] in Table 6. Both BFP and CARAFE are designed based on the original FPN architecture, and they achieve better performance than FPN based on ResNet-50 in GTSDB and STSD. Compared with BFP and CARAFE, the proposed IFA-FPN still brings more substantial and consistent performance gains in all traffic sign datasets GTSDB, STSD, and TT100k. It is because BFP and CARAFE are designed for detecting general objects rather than traffic signs. BFP and CARAFE still mapped RoIs dispersedly in different pyramid levels, which is demonstrated to be an unprofitable factor for traffic sign detection in this paper. Due to the GPU memory limitations, some results, such as BFP, and ResNet-101 and ResNext-101 results in TT100k are not provided, notated as "-".

D. COMPARISON PERFORMANCE

1) COMPARISON WITH FEATURE-BASED METHODS

We first compare our method with feature-based traffic sign detectors wgy@HIT501 [4] and AdaBoost + SVR [2] in GTSDB, and Fourier [5] and AdaBoost + SVR [2] in STSD. The above feature-based traffic sign detection methods [2], [4], [5] did not use the complete GTSDB and STSD datasets to evaluate their methods. Following the previous works [2], [4], [5], we generate and notate these partial GTSDB and STSD datasets as GTSDB† and STSD†, respectively.

The GTSDB† divided traffic signs in GTSDB into three superclasses: Prohibitory signs (P.), Mandatory signs (M.), and Danger signs (D.) though GTSDB provides 43 classes in total. The STSD† only considered the visible traffic signs in STSD with at least 50 × 50 pixels. Moreover, STSD†

TABLE 9. Comparison experiments with state-of-the-art CNN-based one-stage or two-stage detectors.

CNN-based Methods	GTSDB				STSD				TT100k			
	S	M	L	All	S	M	L	All	S	M	L	All
SSD300 [10]	5.5	28.5	90.0	20.4	5.4	48.1	81.8	21.2	7.0	24.6	53.1	19.4
SSD512 [10]	20.5	74.8	63.1	38.3	21.2	76.7	60.3	38.9	23.6	58.6	91.4	48.2
ATSS [33]	24.2	23.5	66.0	19.8	33.4	70.2	82.7	47.3	56.6	82.4	83.3	73.5
FCOS [34]	25.8	21.6	85.3	18.3	29.1	57.5	66.3	39.3	65.6	90.2	89.8	83.3
RetinaNet [35]	35.1	34.9	90.0	29.0	32.3	75.4	86.0	48.0	47.1	70.0	74.4	61.9
YOLO [11]	62.6	56.2	90.0	53.7	24.3	72.5	80.9	40.1	58.7	66.3	59.0	62.0
Attention [14]	-	-	-	-	-	-	-	-	-	-	-	87.0
Faster RCNN [8]	71.1	67.2	100.0	63.4	37.5	86.5	93.3	55.4	74.7	95.6	96.1	89.9
Cascade-RCNN [9]	73.8	74.4	100.0	70.4	43.8	88.3	97.2	61.7	79.6	96.2	95.7	92.0
Faster RCNN +IFA-FPN (ours)	86.5	80.5	100.0	78.0	44.7	84.1	94.4	60.2	79.2	95.9	95.9	91.3
Cascade-RCNN + IFA-FPN (ours)	**86.6**	**85.2**	**100.0**	**80.3**	**48.6**	**94.1**	**97.3**	**65.2**	**84.6**	**96.7**	**97.0**	**93.6**

only considered six main classes, including PEDESTRIAN CROSSING, PASS RIGHT SIDE, NO STOPPING NO STANDING, 50 SIGN, PRIORITY ROAD, and GIVE WAY. Following the rules in [2], [4], [5], our proposed method is performed and the results are reported in Table 7. For GTSDB†, the performances are reported in the area under the curve (AUC) of three superclasses. For STSD†, the performances are reported in recall (Rec.), precision (Pre.), and F1-measure (F1.).

As reported in Table 7, the feature-based methods wgy@HIT501 [4] and AdaBoost + SVR [2] achieved better performance than our method on small and fewer class datasets GTSDB†. The wgy@HIT501 and AdaBoost + SVR considered that traffic signs within each superclass share the same color and shape, therefore they recognized the traffic signs by extracting their color HOG features. These hand-craft feature-based methods work well in simple and small datasets but they cannot achieve satisfying performance on larger datasets STSD†. Our method outperforms other feature-based methods on larger datasets STSD†. It is because our method uses CNN to extract features. The hand-craft features are not robust enough for learning discriminative features to represent the general characteristics of traffic signs, and therefore feature-based methods evaluate their methods only using partial high-quality traffic signs or parts of classes. Subsequently, we compare our method with CNN-based traffic sign detection methods.

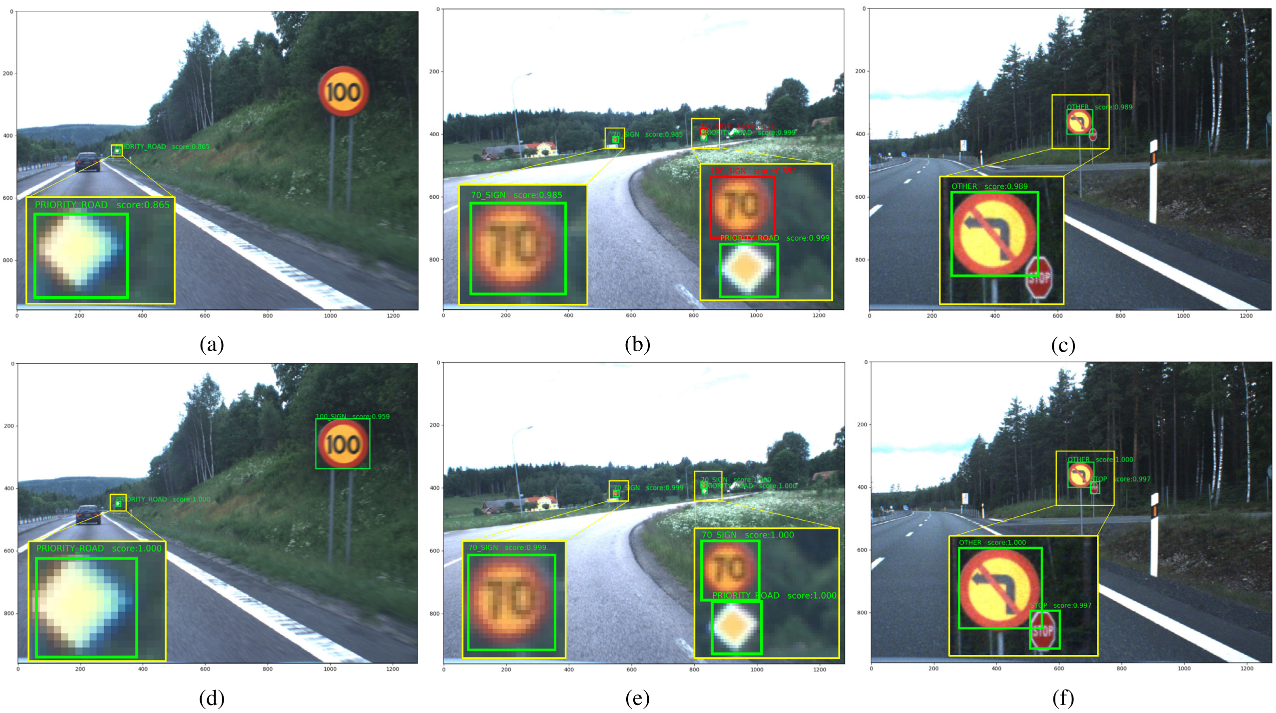
2) COMPARISON WITH CNN-BASED METHODS

Due to the evaluation metrics in several state-of-the-art studies are different, we reported the performance of our proposed methods in two types of evaluation metrics in Table 8 and Table 9. The methods in Table 8 are evaluated using recall (Rec.), precision (Pre.), and F1-measure (F1.), and the methods in Table 9 are evaluated using mAP.

As shown in Table 8, without bells and whistles, the proposed neck network IFA-FPN greatly improves the performance of Faster RCNN and Cascade-RCNN from 80.1% to 90.3%, and 85.6% to 89.0% in F1-measure for small-size groups, respectively. The SSD [10] and MF-SSD [32] cannot achieve satisfying performance because SSD-based detectors need to reduce the original image resolution to a fixed and smaller resolution before forwarding them to the network. This zoom-out function declines the traffic sign detection performance because it removes image information.

For fair and comprehensive comparisons among different architecture, we performed all experiments in Table 9 under the same hardware limitations in MMDetection on three datasets. The results are reported in Table 9. We compare our method with single-stage detectors SSD300, SSD500 [10] and YOLO [11], of which the input images need to be resized to 300 × 300, 500 × 500, and 608 × 608 to fed into them, respectively. Therefore, these single stage detectors show poor performance in small and medium traffic sign detection, which occupy big proportion in datasets. The input image sizes of other methods are consistent with the original image provided by datasets. Attention [14] has no open-source implementation, hence we did not re-perform it in the local computer, and directly use the results reported in the paper. When our proposed IFA-FPN is applied to the Cascade RCNN, the best performances are achieved by 80.3% mAP in GTSDB, 65.2% mAP in STSD, and 93.6% mAP in TT100k. Our proposed IFA-FPN consistently improves the performance of Faster RCNN and Cascade RCNN by a large margin in three datasets, which demonstrate the effectiveness of the IFA-FPN in traffic sign detection.

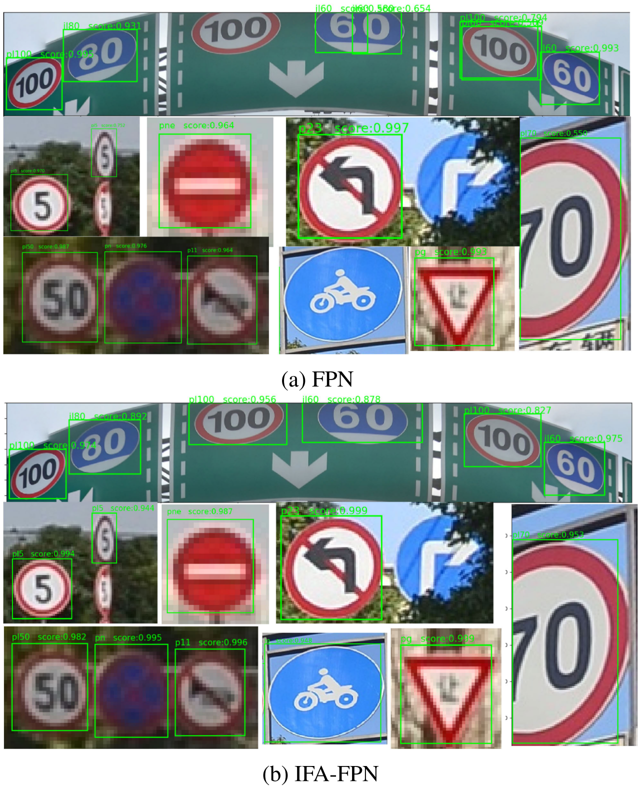
To make straightforward illustrations of the superiority of our proposed IFA-FPN, qualitative detection results on STSD and TT100k by Faster RCNN with FPN and our IFA-FPN are shown in Fig. 5 and Fig. 6, respectively. The green bounding boxes are the true positive (correct) detection, and the red bounding box is the false positive detection. The predicted class and confidence score of the traffic sign are written on the boxes. We observe that the predicted bounding boxes by IFA-FPN are well aligned with the ground truth of traffic sign regions, which indicate IFA-FPN outperforms FPN in both STSD and TT100k. On STSD, IFA-FPN can reduce false positive detection shown in Fig. 5(b) and in Fig. 5(e). IFA-FPN can detect occluded traffic signs shown in Fig. 5(c) and in Fig. 5(f). On TT100k, IFA-FPN can perfectly detect traffic signs with deformation caused by the camera distortion, while the FPN cannot generate well-fitting box for deformed traffic signs or even cannot detect it. IFA-FPN can detect traffic sign 'io' shown in middle of the third row of Fig. 6, while the FPN failed. Moreover, IFA-FPN can always output more confident (higher) scores than FPN for the same



(a) (b) (c)

(d) (e) (f)

FIGURE 5. Qualitative results in STSD dataset. The first row (a-c) are the detection result of FPN, and the second row (d-f) are the detection result of IFA-FPN. The sub-figures at the bottom of each image are zoomed in from the yellow boxes. The green boxes are the true positive (correct) detection, and the red box is the false positive detection.



(a) FPN

(b) IFA-FPN

FIGURE 6. Qualitative results of the detection results by (a) Faster RCNN with FPN and (b) Faster RCNN with IFA-FPN in TT100k.

target in both STSD and TT100k dataset. The illustrations in Fig. 5 and Fig. 6 show that IFA-FPN gets more stable results than the FPN.

V. CONCLUSION

This paper proposed a Plug-and-Play neck network called IFA-FPN that can be applied in mainstream object detectors to improve the performance of a traffic sign detector while a similar inference speed is maintained. An integrated operation is introduced to overcome the size and class imbalance problem in traffic sign datasets by integrating all scale RoIs into a certain pyramid level. Three types of feature aggregation structures are proposed and compared that can enforce multi-scale features learning. The experiments have been done to evaluate the performance of the proposed method on three mainstream datasets including GTSDB, STSD, and TT100k. The experimental results demonstrate the superiority of the proposed IFA-FPN.

In the future, we will focus on light-weighting the network to achieving superior performance in both accuracy and efficiency, then we wish to integrate the proposed method in the ADAS or ADS of a real vehicle.

REFERENCES

[1] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[2] T. Chen and S. Lu, "Accurate and efficient traffic sign detection using discriminative AdaBoost and support vector regression," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 4006–4015, Jun. 2016.

[3] Z. Huang, Y. Yu, J. Gu, and H. Liu, "An efficient method for traffic sign recognition based on extreme learning machine," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 920–933, Apr. 2017.

[4] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang, "A robust, coarse-to-fine traffic sign detection method," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–5.

[5] F. Larsson and M. Felsberg, "Using Fourier descriptors and spatial models for traffic sign recognition," in *Proc. 17th Scandin. Conf. Image Anal.*, 2011, pp. 238–249.

[6] F. Larsson, M. Felsberg, and P. E. Forssen, "Correlating Fourier descriptors of local patches for road sign recognition," *IET Comput. Vis.*, vol. 5, no. 4, pp. 244–254, Jul. 2011.

[7] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1440–1448.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[9] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[11] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[12] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2110–2118.

[13] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, and Z. Zhang, "MMDetection: Open MMLab detection toolbox and benchmark," 2019, *arXiv:1906.07155*. [Online]. Available: http://arxiv.org/abs/1906.07155

[14] Y. Lu, J. Lu, S. Zhang, and P. Hall, "Traffic signal detection and classification in street views using an attention model," *Comput. Vis. Media*, vol. 4, no. 3, pp. 253–266, Sep. 2018.

[15] Y. Yuan, Z. Xiong, and Q. Wang, "VSSA-NET: Vertical spatial sequence attention network for traffic sign detection," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3423–3434, Jul. 2019.

[16] Y. Tian, J. Gelernter, X. Wang, J. Li, and Y. Yu, "Traffic sign detection using a multi-scale recurrent attention network," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4466–4475, Dec. 2019.

[17] Z. Liu, J. Du, F. Tian, and J. Wen, "MR-CNN: A multi-scale region-based convolutional neural network for small traffic sign recognition," *IEEE Access*, vol. 7, pp. 57120–57128, 2019.

[18] D. Tabernik and D. Skočaj, "Deep learning for large-scale traffic-sign detection and recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1427–1440, Apr. 2020.

[19] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[20] 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 *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–8.

[21] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu, "Traffic sign detection and recognition using fully convolutional network guided proposals," *Neurocomputing*, vol. 214, pp. 758–766, Nov. 2016.

[22] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[23] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 761–769.

[24] A. Veit, M. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 550–558.

[25] 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 *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[26] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 98–136, Jun. 2010.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[28] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.

[29] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[30] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 821–830.

[31] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin, "CARAFE: Content-aware ReAssembly of FEatures," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3007–3016.

[32] Y. Jin, Y. Fu, W. Wang, J. Guo, C. Ren, and X. Xiang, "Multi-feature fusion and enhancement single shot detector for traffic sign recognition," *IEEE Access*, vol. 8, pp. 38931–38940, 2020.

[33] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9756–9765.

[34] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 9626–9635.

[35] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 9756–9765.

QING TANG (Member, IEEE) received the bachelor's degree in vehicle engineering from the School of Automotive Engineering, Shanghai University of Engineering Science, Shanghai, in 2015. She is currently pursuing the Ph.D. degree in electrical and computer engineering with the Graduate School of Electrical Engineering, University of Ulsan, Ulsan, South Korea. Her main research interests include computer vision, machine learning, object detection, and object re-identification.

GE CAO (Member, IEEE) received the bachelor's degree in vehicle engineering from the School of Automotive Engineering, Shanghai University of Engineering Science, Shanghai, in 2019. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the Graduate School of Electrical Engineering, University of Ulsan, Ulsan, South Korea. His research interests include pattern recognition, computer vision, and machine learning.

KANG-HYUN JO (Senior Member, IEEE) received the Ph.D. degree in computer controlled machinery from Osaka University, Osaka, Japan, in 1997.

After a year of experience with ETRI as a Postdoctoral Research Fellow, he joined the School of Electrical Engineering, University of Ulsan, Ulsan, South Korea. He is currently serving as the Faculty Dean for the School of Electrical Engineering, University of Ulsan. His research interests include computer vision, robotics, autonomous vehicles, and ambient intelligence. He has served as the Director or an AdCom Member for the Institute of Control, Robotics and Systems, The Society of Instrument and Control Engineers, and the IEEE IES Technical Committee on Human Factors Chair, an AdCom Member, and the Secretary, in 2019. He has also been involved in organizing many international conferences, such as International Workshop on Frontiers of Computer Vision, International Conference on Intelligent Computation, International Conference on Industrial Technology, International Conference on Human System Interactions, and the Annual Conference of the IEEE Industrial Electronics Society. He is an Editorial Board Member of international journals, such as the *International Journal of Control, Automation, and Systems* and *Transactions on Computational Collective Intelligence*.