Parallel Residual Bi-Fusion Feature Pyramid Network for Accurate Single-Shot Object Detection

Ping-Yang Chen, *Student Member, IEEE,* Ming-Ching Chang, *Senior Member, IEEE,* *Jun-Wei Hsieh, *Member, IEEE,* and Yong-Sheng Chen *Member, IEEE*

***Abstract*—This paper proposes the Parallel Residual Bi-Fusion Feature Pyramid Network (PRB-FPN) for fast and accurate single-shot object detection. Feature Pyramid (FP) is widely used in recent visual detection, however the top-down pathway of FP cannot preserve accurate localization due to pooling shifting. The advantage of FP is weakened as deeper backbones with more layers are used. In addition, it cannot keep up accurate detection of both small and large objects at the same time. To address these issues, we propose a new parallel FP structure with bi-directional (top-down and bottom-up) fusion and associated improvements to retain high-quality features for accurate localization. We provide the following design improvements: (1) A parallel bifusion FP structure with a bottom-up fusion module (BFM) to detect both small and large objects at once with high accuracy. (2) A concatenation and re-organization (CORE) module provides a bottom-up pathway for feature fusion, which leads to the bi-directional fusion FP that can recover lost information from lower-layer feature maps. (3) The CORE feature is further purified to retain richer contextual information. Such CORE purification in both top-down and bottom-up pathways can be finished in only a few iterations. (4) The adding of a residual design to CORE leads to a new Re-CORE module that enables easy training and integration with a wide range of deeper or lighter backbones. The proposed network achieves state-of-the-art performance on the UAVDT17 and MS COCO datasets.**

***Index Terms*—Feature pyramid network, parallel residual bi-fusion, object detection, single-shot, CNN, feature fusion.**

I. Introduction

Visual object detection has improved significantly in the state-of-the-art (SoTA) models. Recent deep models including FPN [1], YOLOv3 [2], and SSD [3] typically consist of three components: (1) a deep *feature extraction backbone e.g.* DarkNet-53 [4] or ResNet-101 [5], (2) a *feature pyramid* (FP), and (3) an *object classifier*. To ensure high detection accuracy, most SoTA object detectors adopt deep CNN structures that can achieve impressive performance in detecting large and medium sized objects. However the performance for detecting smaller objects are still inferior [6]. This is mainly because the feature map resolution is reduced after simple pooling in the FP. Tiny objects ($< 32 \times 32$ pixels) can turn into about a single-pixel feature vector in the last layer of FP, causing insufficient spatial resolution for accurate discrimination. On the other hand, using a shallow backbone increases the computational efficiency. This comes with the drawback of reduced detection performance, as the capability to retain contextual and semantic features also decreases directly.

In general, detecting small objects is more difficult than detecting large objects. Both high-level and low-level features are required to discriminate and localize objects among background and other objects. YOLOv3 [2] maintains detailed grid features to retain detection accuracy of small objects. However, the effectiveness is limited, as accurate detection of both small and large objects cannot be kept together at the same time. The best performing method from LPIRC 2019 challenge [7] shows improvement on detecting general-sized objects but not small objects on the COCO dataset [8].

How to design fast and accurate network that can effectively detect all object sizes is still an open question. One solution to retain accurate feature localization is to add a bottom-up pathway to offset the lost information from low-level feature maps. In [9], the adding of a gating module on the SSD frame leads to a gated bi-directional FP; however such gated network is not easily trainable. In [10], a bottom-up path aggregation network was proposed for object segmentation. The bi-directional network of [6] can efficiently circulate both low-level and high-level semantic information for small object detection. In [11], a BiFPN was proposed based on NAS-FPN [12] to better detect small objects with high efficiency. In YOLOv4 [13], path aggregation [10] was modified by replacing the addition with concatenation for better detection of small objects. However, such BiFPN structure still cannot keep up accurate detections of both small and large objects all together.

We propose a new **Parallel Residual Bi-Fusion Feature Pyramid Network (PRB-FPN)** with a parallel design and multiple improvements that can retain both deeper and shallower features for fast and accurate single-shot object detection. Different from other bi-fusion FPN structures such as PANet [10], NAS-FPN [12], and BiFPN [11], we create a parallel bi-fusion structure to fuse three-layers of feature maps in parallel to generate three prediction maps at the same time, see Fig. 1. Without losing efficiency, these three-way prediction maps can retain more accurate semantic and localization information to better detect both tiny and large objects. In this parallel structure, we introduce a new concatenation and re-organization (CORE) module for data fusion, where output features can be further purified to retain contextual information. We introduce a "residual" design (motivated from the spirit of ResNet [5]) into our bi-fusion pipeline,

Ping-Yang Chen and Yong-Sheng Chen are with the Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan.

Jun-Wei Hsieh is with the College of Artificial Intelligence and Green Energy, National Yang Ming Chiao Tung University, Hsinchu, Taiwan.

Ming-Ching Chang is with the Computer Science Department, University at Albany, SUNY, NY, USA.

*Jun-Wei Hsieh jwhsieh@nctu.edu.tw is the corresponding author.

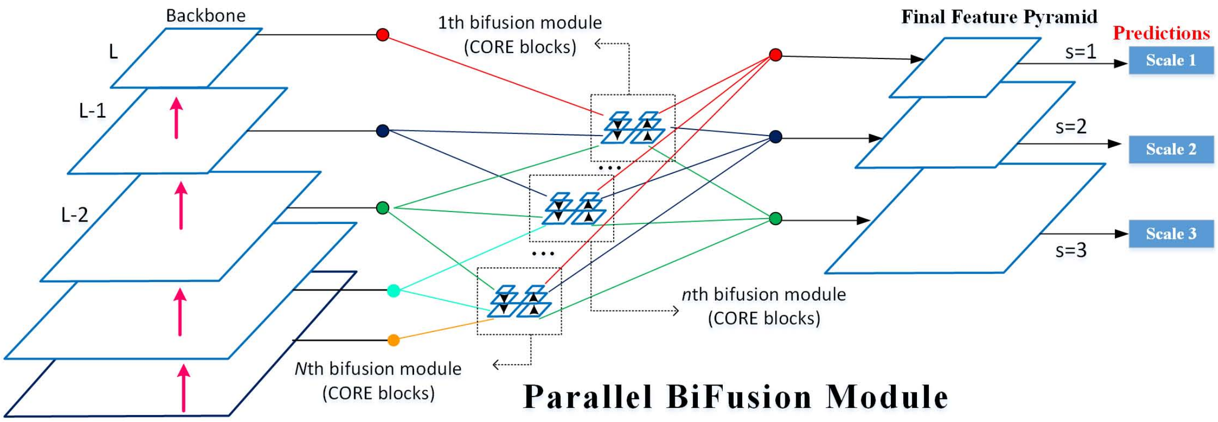


Fig. 1. Overview of the proposed **Parallel Residual Bi-Fusion Feature Pyramid Network (PRB-FPN)**.

which enables easy training and integration with a number of popular backbones. We will show that our residual FP design outperforms other bi-directional methods [6], [14] in Section IV. In comparison, methods based on traditional FPs [1], [2], [3], [4] can only learn un-referenced features, thus they are not suitable for detecting both large and small objects. Our residual FP retains semantic richer features in higher layers that can better detect small objects.

A key novelty in our design is the adding of *parallelization* to the bi-fusion FPN architecture. This parallel design is more effective in feature representation, *i.e.* for capturing features to identify and localize objects in either small or large sizes without losing efficiency. In comparison, most existing bi-directional FP methods [6], [15], [14], [16] directly concatenate large feature maps in a memory-consuming way, which ends up with an even larger feature map.

The proposed PRB-FPN is simple, efficient, and suitable for generic object detection for multiple object classes and sizes (small, mid, and large). We will show in Section IV that our approach is generalizable in combining with mainstream backbones including Pelee [17] and DarkNet53 [2]. It can run in real-time and is easily deployable to edge devices. Main contributions of this paper are summarized in the following:

- We propose a new Parallel Residual Bi-Fusion Feature Pyramid Network (PRB-FPN) that can effectively fuse both deep and shallow feature layers in parallel for fast and accurate one-shot object detection.
- The parallel design of PRB-FPN makes it well-suited for detecting objects in both small and large sizes with higher accuracy.
- The PRB-FPN can be easily trained and integrated with different backbone thanks to the residual design. A newly proposed bottom-up fusion module (BFM) can improve the detection accuracy of both small and large objects.
- Extensive experiments on Pascal VOC [18] and MS COCO [8] datasets show that PRB-FPN achieves the SoTA results for accurate and efficient object detection. Results also show great generalization ability on various object sizes and types.

II. Related Works

Object detection is a very active field in computer vision since the blooming of deep learning. The extensive amount of literature can be organized into two categories based on their network architectures: *two-stage* proposal-driven and *one-stage* (single-shot) approaches. In general, two-stage methods can achieve high detection accuracy but with longer computation time, while one-stage methods run faster with inferior accuracy. We focus on the survey of one-stage object detectors. RefineDet [19] employs an encode-decode structure in the deeper network with the use of up-sampling deeper scale features to enrich contextual information. PeLee [17] is a variant of DenseNet [20] that outperforms SSD+MobileNet by 6.53% on the Stanford Dogs dataset [21] based on a much shallower network. However, PeLee [17] does not detect small objects well on MS COCO [8]. PFPN [15] adopts the VGGNet-16 backbone [22] and SPP to generate a feature pyramid by concatenating multi-scale features.

One-stage object detectors mostly consist of a backbone network and a predictor. The backbone is a stacked feature map representing the input image in high feature resolution (but low spatial resolution for abstraction). The backbone network can be pre-trained as an image classifier on a large dataset such as ImageNet. OverFeat [9] was the first CNN-based one-stage object detector developed in 2013 with a sliding-window paradigm. Two years later, the first version of YOLO [23] achieved state-of-the-art performance by integrating bounding box proposals and subsequent feature resampling in a single stage. SSD [3] employed in-network multiple feature maps for detecting objects with varying shapes and sizes. The multi-map design enabled SSD with better robustness over YOLOv1 [23]. For better detection of small objects, The Feature Pyramid Network (FPN) [1] based on FP can achieve higher detection accuracy for small objects. YOLOv3 [2] was developed by adopting the concept of FPN. By changing the backbone from DarkNet-19 [4] to DarkNet-53, YOLOv3 achieves superior performance in 2018. Similarly, RetinaNet [24] combines FPN [1] and ResNet [5] as the backbone. RetinaNet used focal loss to significantly reduce false positives in a single stage, such that the weights of each anchor box can be dynamically adjusted.

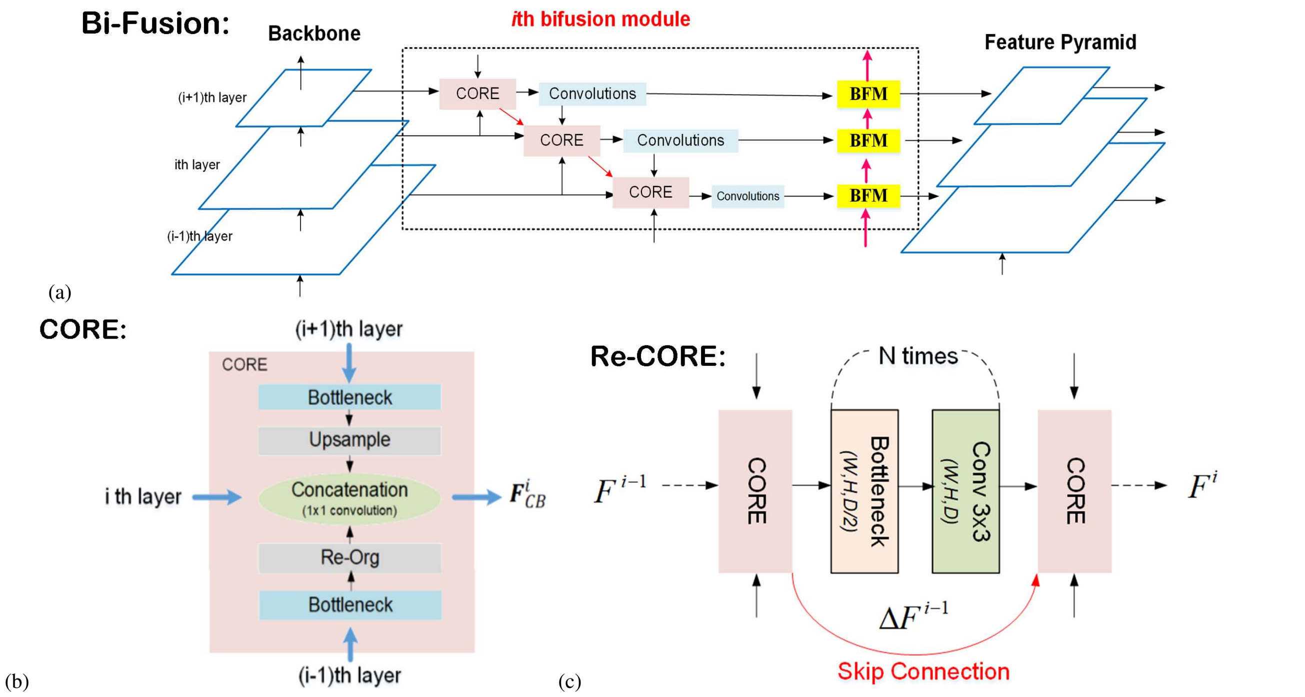


Fig. 2. Detailed network architecture for the proposed modules (refer to Fig. 1 for the overall architecture of PRB-FPN): (a) the **Bi-Fusion** module for concurrent fusion of contextual features from adjacent layers, (b) the **concatenation and re-organization (CORE)** module for recursively fusion of contextual features from adjacent layers, and (c) the **Residual CORE (Re-CORE)** design in combining CORE with the residual design inspired from ResNet.

Shift-invariance in CNNs was originally achieved using subsampling layers. The work of [25] evaluated the effect of small geometry perturbations on CNN and suggested that max-pooling is more effective in object detection and classification. In [26], a pooling-after-blurred technique was proposed by combing blurring and sub-sampling techniques to ensure shift-invariance.

Feature pyramid (FP) is widely used in SoTA detectors for detecting objects at different scales, where spatial and contextual features are extracted from the last layer of the top-down path for accurate object detection. This top-down aggregation is now a common practice for improving scale invariance in both two-stage and one-stage detectors. Popular FPs used for this purpose include the pyramidal feature hierarchy (bottom-up), hourglass (bottom-up and top-down), FPN [1], SPP [27], and PFPN [15]. It is also well-known that the top-down pathway in FP cannot preserve accurate object localization due to the shift-effect of pooling.

Bi-directional FP can recover lost information from shallow layers to improve **small object detection** in several works [6], [14], [16]. A gating module was used to control the feature flow direction in [14]. A light-weight scratch network and a bi-directional network were constructed in [6] to efficiently circulate both low- and high-level semantic information. M2Det [28] is a one-stage detector that outperforms most 2019 methods on all multi-scale categories on MS COCO [8]. However, the M2Det model is complicated and time-consuming, thus is not suitable for real-time object detection. Inspirited by NAS-FPN [12], a BiFPN was proposed in [11] to better detect small objects with higher efficiency. The recent YOLOv4 [13] modified the path aggregation method [10] by replacing the addition with concatenation to better detect small objects. However, this BiFPN structure still cannot keep up accurate detection of both small and large objects all together.

Multi-Scale Object Detectors face the challenge of small-size false positives due to the inadequacy of low-level features, which result in small receptive field size and weak semantic capabilities. The work of [29] demonstrates that independent predictions from different feature layers on the same region are beneficial in reducing false positives. In [30], a novel paradigm of multi-scale deep network is developed to model the spatial contexts surrounding different pixels at various scales. In [31], deep convolutional networks are used to obtain multi-scaled features, where deformable convolutional structures are added to overcome geometric transformations.

Anchor-free methods [32], [16], [33], [34] do not reply on handcrafted anchors, thus are free of issues commonly associated with anchor-based designs. In [32], corner features are detected for object detection. By inheriting the architecture of R-CNN, ME R-CNN [35] used multiple stream pipelines for accurate anchor-free object detection, where one pipeline is an expert for processing a certain type of ROIs and controlled by an expert assignment network. A cascade anchor refinement module is proposed in [16] to refine pre-designed anchors. This is then injected into a bidirectional FP, which can detect objects with highly accurate localization. However, one pass of regression during training is not accurate enough for detection in this anchor-free approach. In [36], an attention CoupleNet was proposed by designing a cascade attention structure to generate class-agnostic attention maps of target regions so that a discriminative feature representation can be formulated for part-based object detection. In [37], Jin *et al.* used an adaptive anchor generator to generate all possible anchor boxes. They then proposed a semi-anchor-free network for object detection with an enhanced feature pyramid which consists of two modules, *i.e.*, adaptive feature fusion module (AFFM) and self-enhanced module (SEM). In [33], a number of low-quality bounding boxes are predicted and further verified with a *centerness* branch that can detect objects without using any pre-defined anchor boxes. In [38], a hierarchical shot detector is used to predict detection bounding boxes via regression. These regression based methods are more accurate but less efficient. Compared with CornerNet [32], FoveaBox [39] does not require any embedding or grouping techniques at post-processing stage to locate real bounding boxes. However, its latency is higher and results in lower efficiency.

Network Transferring The above detectors can be trained well enough from a large set of sufficiently representative data. However, as there exists numerous application scenarios in which only a few training samples (*e.g.* tumor images in medical applications) are available, transfer learning can be used to customize the model and adopt to the tasks [40]. For example, in [41], a weakly-shared Deep Transfer Network (DTN) was proposed to hierarchically learn and transfer semantic knowledge from web texts to images for image classification. In [42], a novel generalized DTNs was proposed to solve the problem of insufficient training images by transferring label information across heterogeneous domains, such as transferring from the textual to visual domain for image classification. Moreover, in [43], a transfer learning system named GAIA was proposed to provide powerful pre-trained weights, select models, and collect relevant data for object detection when only a few training samples were given. However, although network transferring methods can provide performance improvements, they cannot outperform ordinary methods trained with sufficient samples.

III. Method

We first motivate the design of our proposed network architecture by addressing the limitation of the Feature Pyramid (FP) for visual object detection. In Section III-A, details of our new parallel bi-fusion scheme are described. The adding of *parallelization* to the bi-fusion FPN architecture can better capture features for both small and large objects without degrading efficiency. Section III-B describes our new feature concatenation and re-organization scheme that can effectively circulate semantic and localization information. Section III-C further adopts a *residual recursive* formulation into our pipeline, which enables easier training and better performance for small object detection. Finally, Section III-D adds one more design of bottom-up location feature fusion that can further improve object localization. Figs. 1, 2 and 3 depict the complete pipeline of our proposed network architecture. Details are provided in the following sessions.

A. Parallel Concatenation and Re-organization Feature Bi-Fusion Architecture

Feature Pyramid (FP) is widely-used in top-down feature aggregation that can collect semantically rich features to effectively discriminate objects with scale invariance. However, it is well-known that FP cannot preserve accurate localization for small objects due to pooling and quantization. The winning methods of LPIRC 2019 challenge [7] show improvements on detecting general-sized objects but not on small objects. There object prediction was carried out using information from both each pyramid layer and the respective lower layers. This coincide with thoughts from several SoTA bi-directional methods [25], [15], [14] in leveraging new feature streams from lower feature layers (or the raw image itself) to keep track of features from smaller objects and achieve more accurate localization. Such bi-fusion modules specially designed for improving small object detection still lack capabilities in detecting larger objects.

In this paper, we propose an effective *parallel* FP fusion design to tackle this difficult problem of *object detection considering all object scales*. This is done by creating multiple bi-fusion paths to keep tracks of features that are suitable to detect objects of all sizes (including tiny and large objects). Each bi-fusion path keeps track of size-dependent features to represent objects at a specific scale. Assume that there are N prediction maps (where $N = 3$ for YOLOv4), we propose to execute the N different concurrent fusion paths to generate N fused feature maps for the N prediction maps. We use n to index the bi-fusion modules (thus $n \leq N$). Let L be the level of the top layer in the FP. As shown in Fig. 1, the n^{th} bi-fusion module will *bi-fuse* feature maps from the $(L-n+1)^{th}$ layer to the $(L-n-N+2)^{th}$ layer in the backbone. The s^{th} output will be fed into the s^{th} prediction map for object detection, which will integrate feature maps from the s^{th} layer of all bi-fusion modules. Noticeably, the 1^{st} bi-fusion module in our model corresponds to the sole bi-fusion module in SoTA bi-directional methods [25], [15], [14].

B. Concatenation and Re-organization for Feature Bi-Fusion

In Fig. 2, each bi-fusion module consists of three **concatenation and re-organization (CORE)** blocks and two skip connections. Details of the bi-fusion module are shown in Fig. 2(a). The **CORE** design in Fig. 2(b) brings a major advantage that feature fusion can be recursively applied in both top-down and bottom-up fashions to: (1) concatenate semantic features from top layers (top-down), and (2) re-organize spatially rich localization features from bottom layers (bottom-up). To avoid using too many dithering operations (*i.e.*, point-wise convolutions) and to avoid computationally expensive operations (*i.e.*, pooling and addition), we adopt an 1×1 depth-wise convolution in the CORE module. This enables effective fusion of pathways coming from deeper and shallower layers in each layer of the FP. Our 1×1 depth-wise convolution in CORE is very different from most of SoTA bi-directional methods [25], [15], [14], [16], where feature fusion is carried out by concatenating all feature maps. Their simple concatenations result in a large feature map proportional to the total feature size. In contrast, our 1×1 conv filter in CORE is automatically learned, such that features can be fused more effectively via a feature map of fixed size.

In each layer of the used backbone, CORE fuses features of each layer with its two adjacent (immediately shallower and deeper) layers. In other words, feature *bi-fusion* is performed

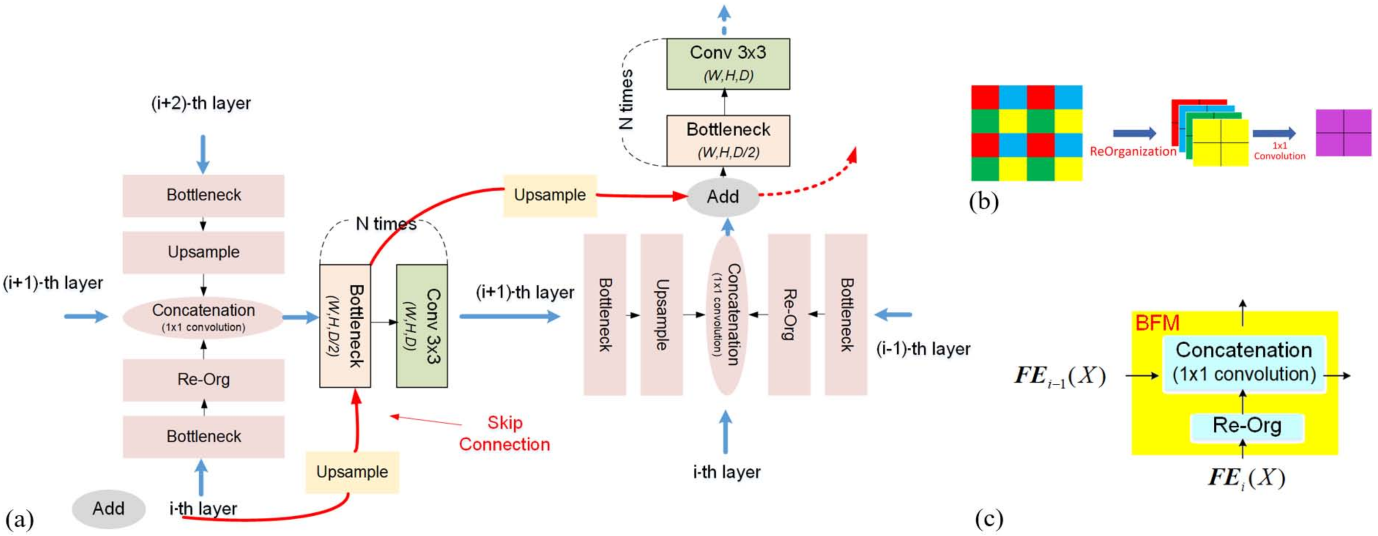


Fig. 3. (a) Details of the proposed **Re-CORE** architecture. (b) The **Re-Org** block for feature re-organization. (c) The **bottom-up fusion module (BFM)**.

in the feature pyramid of CORE. In the bottom-up fusion with the shallower layer, similar to YOLOv2 [4], a **Re-Org** block from Fig. 3(b) is adopted in Fig. 2(b) to re-organize the feature map into 4 channels. However, instead of using a concatenation operation, the 1×1 convolution filter is then performed to fuse all feature maps as the output.

C. Residual Bi-Fusion Feature Pyramid

We further adopt the residual concept inspired from ResNet [5] to the CORE block in our design, and created a new **Residual CORE (Re-CORE)** block. Re-CORE enables the fusion of four adjacent scales (namely, the *shallow*, *current*, *deep*, and *deeper* layers) for better detection of small objects. Specifically, by recursively injecting the output of the $(i+1)^{th}$ CORE module to the i^{th} CORE module, the Bi-Fusion FP becomes a fully-featured **Residual Bi-Fusion FPN** as in Fig. 3(a). Fig. 2(c) depicts the connection between the Re-CORE and Convolution modules, in which F^i and ΔF^i denote the outputs of the i-th Re-CORE and Convolution modules, respectively.

Fig. 3(a) shows the detailed Re-CORE architecture. The Re-CORE module performs bi-fusion to integrate features from the four input layers with residual design. The output of the previous Re-CORE module becomes the input of the current Re-CORE module via a *skip connection*, which is depicted as a red line in Fig. 2(c) and Fig. 3(a), respectively. Features from the i^{th}, $(i-1)^{th}$, and $(i+1)^{th}$ layers are fused by an 1×1 convolution and then added to an up-sampled version of the skip connection to produce a new feature map. This map is then fed into a convolution block to produce the final output of this Re-CORE module.

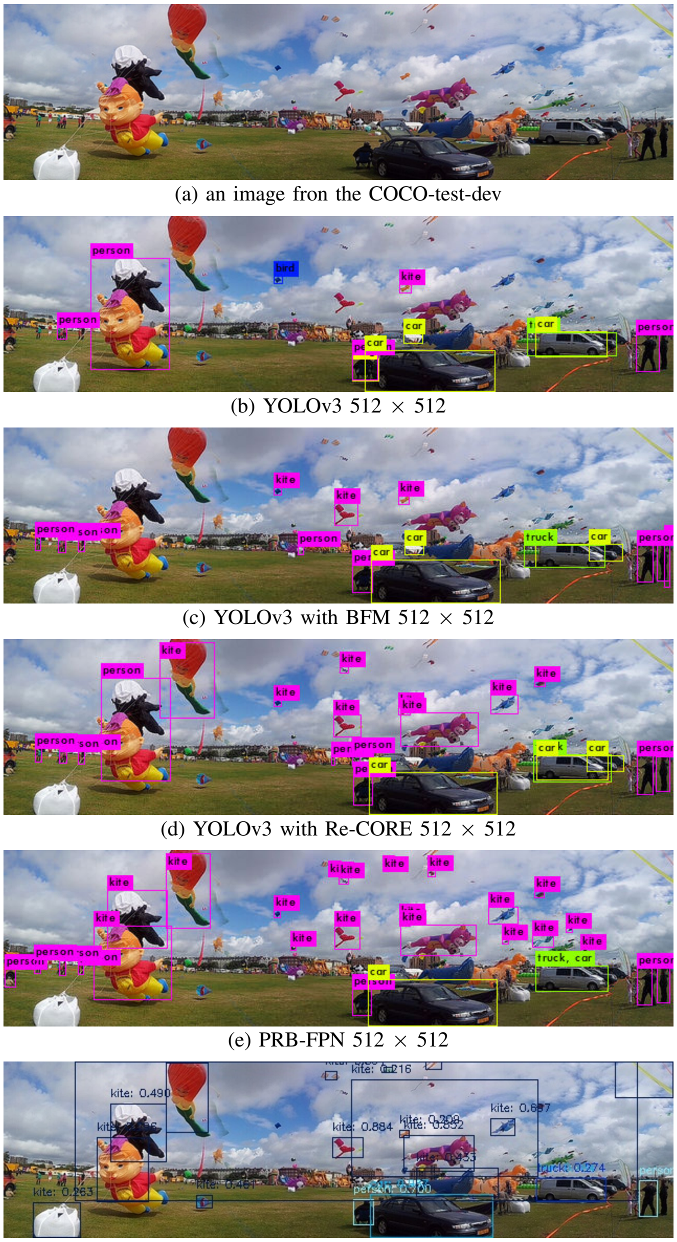
Working with popular backbones: Similar to ResNet [5], the residual nature of our Re-CORE module enables easy training and integration of the FP with a wide range of backbones that works particularly well for small object detection. Instead of learning un-referenced features, Re-CORE obtains better accuracy from the largely increased feature depths when compared with traditional FPs [1], [2], [4], [3]. Note that SoTA FPs [1], [27], [15] often learn redundant features and perform poorly on small object detection.

The Re-CORE module provides a new effective fusion approach for collecting localization information from bottom layers that can improve the accuracy of small object detection. In comparison, the naive approach in [44] detects small objects by generating high-resolution images as inputs to the detection module, which comes with a cost of large computational burden. Another approach for small object detection is to leverage contextual information, by sending semantic features from a top-down way via a FP as in YOLOv3 [2]. However, in these methods without the use of residual property, the learning will include un-referenced features and thus bound the number of FP layers that can actually contribute to object detection. In comparison to the FP proposed in [7], our Re-CORE module can capture richer semantic features from deeper layers that can directly improve small object detection.

In summary, our residual design and bi-directional fusion make the Re-CORE module suitable for detecting small and even tiny objects without notable computation overheads.

D. Bottom-up Feature Fusion

As aforementioned, the top winning method in LPIRC 2019 [7] improved detection on large and medium-sized objects, but not able to keep up the performance for small objects. To address this issue, we propose the adding of a **bottom-up fusion module (BFM)** to the PRB-FPN network to further improve the localization of both small and large objects. Fig. 3(c) depicts the proposed BFM architecture. Instead of using convolution with stride 2 (adopted in PANet [10], Bi-FPN [11], or YOLOv4 [13]), the BFM adopts a Re-Org block to split C channels of feature map into $4C$ channels to better preserve spatial information and generate robust semantic features via 1×1 convolution, which improves small object detection. As for the bidirectional FPN-BPN work [16], convolutions with stride 2 are used for down-sampling, while



(a) an image fron the COCO-test-dev

(b) YOLOv3 512 × 512

(c) YOLOv3 with BFM 512 × 512

(d) YOLOv3 with Re-CORE 512 × 512

(e) PRB-FPN 512 × 512

(f) M2Det [28] 512 × 512

Fig. 4. Small object detection results on the MS COCO test set.

de-convolutions are adopted for up-sampling. However, this design results in lower accuracy for small object detection due to the stride 2 operator, and the use of de-convolution leads to low efficiency in object detection.

In summary, in our design: (1) Section III-A describes the use of parallel bi-fusion paths that run concurrently for effective detection of both small and large objects; (2) Section III-C describes the adding of the Re-CORE module for improving the detection of small objects; and finally, (3) the BFM in Section III-D can bring specific local information from a bottom-up pathway to localize the objects more accurately. The BFM pathway works particularly well for detecting both large and mid-sized objects. Experimental results in this regard are shown in Table III.

TABLE I
ABLATION STUDY OF BFM AMONG DIFFERENT BACKBONES.

Backbone	BFM	FPS	AP	AP_{50}	AP_{75}	AP_s	AP_M	AP_L
DarkNet53		**28.9**	28.6	50.7	29.6	15.5	30.4	35.3
512x512	✓	28.4	**34.9**	**57.2**	**37.7**	**18.6**	**37.1**	**45.3**
Pelee		**85.8**	26.7	49.9	26.2	13.5	27.8	33.5
512x512	✓	84.5	**28.3**	**51.8**	**28.4**	**14.0**	**30.1**	**35.6**
VGG16		**31.8**	34.1	58.3	35.8	17.9	35.9	44.1
512x512	✓	31.4	**34.6**	**58.6**	**36.7**	**18.6**	**36.5**	**44.3**
DenseNet201		**30.5**	30.1	54.5	32.5	15.7	33.8	40.8
512x512	✓	39.4	**31.5**	**54.7**	**33.3**	**15.9**	**33.9**	**41.1**

IV. EXPERIMENTAL RESULTS

We evaluate the PRB-FPN against SoTA object detection methods on MS COCO benchmark [8] and UAVDT [45] using machines with nVidia Titan X GPU and V100. Accuracy is evaluated in the metric of Average Precision (AP). Computational efficiency is evaluated in the processing frames per second (FPS).

Backbones. Our pipeline is not limited to any feature extraction backbone. We evaluated the following backbones: PeLee [17], MobileNet-V2 [46], DarkNet-53 [2], VGG16 [22], `ResNet-50` [5], DenseNet [20], CSPnet [47].

A. Implementation Details and Evaluation Configures

Implementation details: For performance evaluations on MS COCO dataset, the default hyper-parameters are set as follows. Total training steps are $500,500$ with the step decay learning rate 0.001. The learning rate is further multiplied by a factor 0.01 at the $400,000$ steps and $450,000$ steps, respectively. Momentum and weight decay rate are set to be 0.9 and 0.0005, respectively. All various PRB models were trained on a single V100 with batch size 64, and mini-batch size 16, 8, or 4 depended on the used model size for fitting the limitation of the available GPU RAM.

Evaluation details: We next evaluate the newly introduced designs of PRB-FPN in terms of how each design effectively fuse both deep and shallow feature layers in parallel for fast and accurate one-shot object detection. The first major design in the parallel structure of PRB-FPN is the new *residual concatenation and re-organization* (ReCORE) module proposed for effective and efficient data fusion. The second major design is the *bottom-up fusion module* (BFM) added after ReCORE in PRB-FPN as shown in Fig. 2, which can further improve the localization of both small and large objects. To evaluate the effect of each module, accuracy improvement based on BFM is first evaluated in § IV-B. The effect of BFM with ReCORE module for accuracy improvements is evaluated in § IV-C. Evaluation of the RB-FPN module against the original RPN is provided in § IV-D. Finally, comparisons between PRB-FPN and other SoTA methods are provided in § IV-E.

B. BFM Accuracy Improvements

Since the BFM in PRB-FPN is designed to detect both small and large objects, we evaluate the effectiveness of BFM on object detection based on the MS COCO dataset across four backbones, namely PeLee [17], DarkNet-53 [2], VGG16 [22],

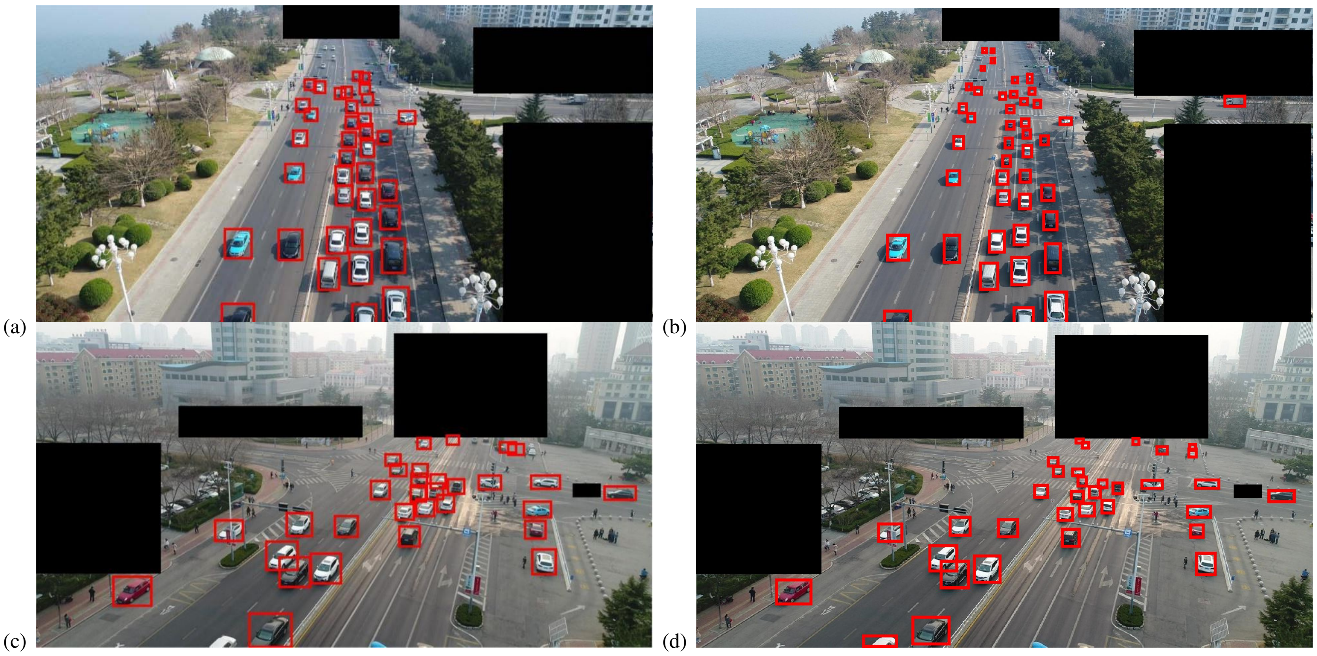


Fig. 5. Small object detection results on the UAVDT17 benchmark [45]. (a) and (c): LRFNet [6]. (b) and (d): The proposed PRB-FPN. Black boxes indicate don't-care regions that come with the original UAVDT17 dataset.

TABLE II
COMPARISONS BETWEEN OUR BFM AND OTHER SOTA BI-DIRECTIONAL FUSION METHODS.

Method	Backbone	Input size	FPS	AP	AP_{50}	AP_{75}	AP_s	AP_M	AP_L
GBFPN-SSD [14]	VGG16	512×512	-	-	33	-	-	-	-
FPN-BPN [16]	VGG16	320×320	**32.4**	29.6	48.4	32.3	9.6	32.5	44.3
FPN-BPN [16]	VGG16	512×512	18.9	33.1	53.1	36.3	15.7	**37**	44.2
EfficientDet-D0 [11]	EfficientNet [49]	512×512	-	34.6	53.0	37.1	-	-	-
NAS-FPN [12]	ResNet-50	1024×1024	-	44.2	-	-	-	-	-
BFM [Ours]	VGG16	512×512	31.4	34.6	58.6	36.7	18.6	36.5	44.3
RB-FPN [Ours]	ResNet-50	512×512	32.1	**44.3**	**65.1**	**48.2**	**25.1**	**47.3**	**56.8**

TABLE III
ABLATION STUDY OF RE-CORE AND BFM; RB DENOTES THE PROPOSED RESIDUAL BI-FUSION DESIGN AS IN FIG.3(A).

Method	Backbone	Re-CORE	BFM	FPS	AP	AP_{50}	AP_{75}	AP_s	AP_M	AP_L
Pelee*	Pelee*			**85.8**	26.7	49.9	26.2	13.5	27.8	33.5
Pelee with BFM	Pelee		✓	84.5	**28.3**	**51.8**	**28.4**	**14.0**	**30.1**	**35.6**
Pelee with RB-FPN	Pelee	✓	✓	84.2	**29.5**	**52.9**	**30.2**	**14.9**	**33.1**	**36.7**
Yolov3†	Darknet53†			**28.9**	32	56.5	33	17.4	34	41.4
Yolov3-SPP†	Darknet53†			28.7	35.3	59.2	37.4	16.9	37.1	48
Yolov3‡	Darknet53‡			**28.9**	28.6	50.7	29.6	15.5	30.4	35.3
Yolov3 with Re-CORE	Darknet53	✓		27.6	36	59.5	38.2	18.9	37.3	47.1
Yolov3 with BFM	Darknet53		✓	28.4	34.9	57.2	37.7	18.6	37.1	45.3
Yolov3 with RB-FPN	Darknet53	✓	✓	**27.2**	**36.8**	**59.7**	**39.6**	**19**	**39.5**	**48**
Yolov4	CSPDarknet53			**31**	43	64.9	46.5	24.3	46.1	55.2
Yolov4 with Re-CORE	CSPDarknet53	✓		28.5	44.8	66.5	47.3	26.9	46.3	55.8
Yolov4 with BFM	CSPDarknet53		✓	30.5	43.7	65.3	47.1	24.5	48.2	55.3
Yolov4 with RB-FPN	CSPDarknet53	✓	✓	27.3	**45.1**	**67.2**	**48.2**	**27.1**	**48.5**	**57**

* The input size for all backbones is 512x512.
* Trained and tested by ourselves according to the paper.
† Test results with weights provided in the YOLOv3 website.
‡ Trained and tested by ourselves according to the instruction.

and DenseNet [20]), to evaluate the generalization capability of BFM. Table I shows this BFM ablation study results. Observe that the BFM computational load is very light and can be ignored for all backbones. Also observe the generalizability of BFM in maintaining high AP across these backbones for detecting different object sizes. Table I also shows another

TABLE IV
IMPROVEMENTS BY PARALLEL AND RESIDUAL FPNS ON UAVDT [45] BENCHMARK.

UAVDT-Benchmark-TestSet				
Methods	Backbone	input size	AP	FPS
Faster-RCNN	VGG-16	1024x540	22.32	2.8
R-FCN	ResNet-50	1024x540	34.35	4.7
SSD	VGG-16	512x512	33.62	41.56
RON**	VGG-16	512x512	21.59	11.1
RetinaNet	ResNet-101-FPN	512x512	33.95	25
LRFNet	VGG-16	512x512	37.81	91
SpotNet	Hourglass-104	512x512	52.8	-
CenterNet	Hourglass-104	512x512	51.18	-
BFM	MobileNet-V2	512x512	29.7	113
Re-CORE	MobileNet-V2	512x512	34.2	110
PRB-FPN	MobileNet-V2	512x512	**65.47**	75
Yolov4 with BFM	CSPDarknet-53	512x512	**64.52**	30.5
Yolov4 with Re-CORE	CSPDarknet-53	512x512	**65.41**	26.3
Yolov4 with PRB-FPN	CSPDarknet-53	512x512	**76.55**	19.2

TABLE V
ABLATION STUDY OF THE NUMBER OF FEATURE PYRAMIDAL LAYERS FOR PRB WITH RESNET50 AND RB WITH CSPDARKNET53 ON UAVDT [45] BENCHMARK.

Number of FP layers			Method	Backbone	FPS	mAP
3	4	5				
✓			PRB	ResNet-50	**31.26**	70.71
	✓		PRB	ResNet-50	27.15	72.32
		✓	PRB	ResNet-50	22.30	74.19
✓			PRB	CSPDarknet-53	19.2	76.55
	✓		PRB	CSPDarknet-53	12.2	77.82
		✓	PRB	CSPDarknet-53	4.7	**79.21**

important observation that BFM can improve the detection accuracy of a shallower backbone more than deeper backbone. Specifically, the improvements on AP50 with BFM for DarkNet [2], Pelee [17], and DenseNet [20] are 6.5%, 3.5%,

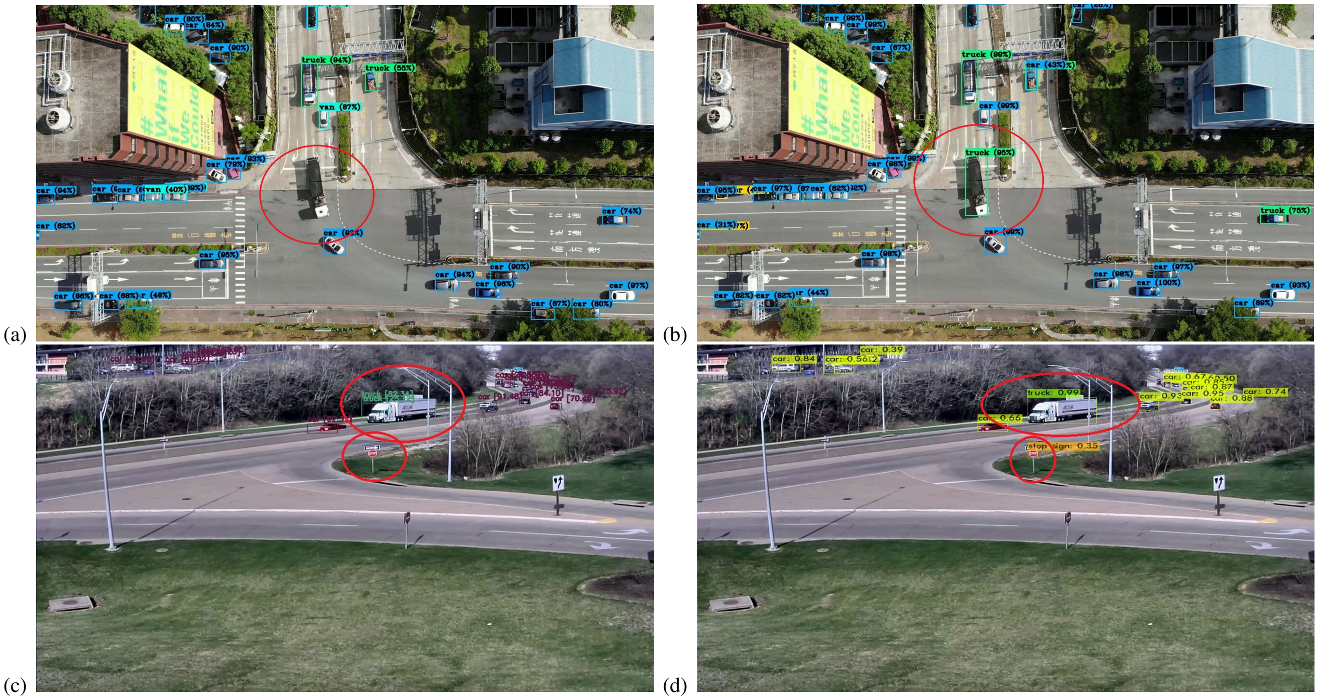


Fig. 6. Comparisons of object detection between YOLOv4 and our PRB-FPN.(a) and (b) are the results of our home pictures taken from aerial cameras in Suao, Taiwan; (c) and (d) are the results on the AI City Challenge [48]. (a) : YOLOv4 512 × 512, (b) : PRB-FPN w/o 512 × 512. (c) : YOLOv4 512 × 512, (d) : PRB-FPN w/o 512 × 512.

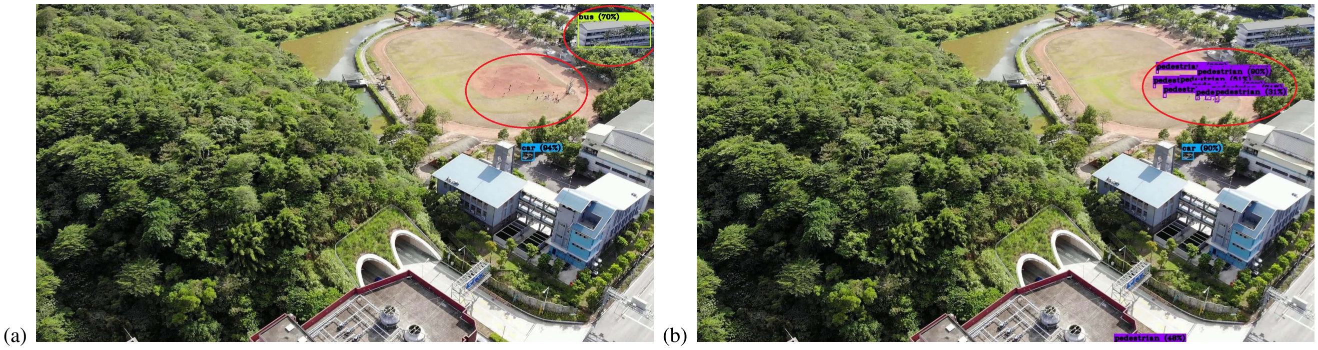


Fig. 7. Comparisons of object detection between YOLOv4 and our PRB-FPN.(a) and (b) are the results of our home pictures taken from aerial cameras in Suao, Taiwan. (a) : YOLOv4 512 × 512, (b) : PRB-FPN 512 × 512.

and 0.2%, respectively. This indicates that BFM improves the detection of large objects better than smaller objects. Thus, BFM can provide a good solution for applications that demands the detection of arbitrary-sized objects.

In addition to the ablation study of our BFM method among different backbones, Table II shows the comparisons against other SoTA bifusion methods in terms of accuracy and efficiency. When the backbones `ResNet-50` and EfficientNet are adopted, our BFM method outperforms EfficientDet-D0 [11] and NAS-FPN [12]. As for the bidirectional FPN-BPN [16], their convolutions with stride 2 for down-sampling result in lower accuracy in small object detection. In addition, their de-convolution for upsampling results in lower efficiency for object detection.

C. BFM with Re-CORE Accuracy Improvements

We evaluate the effectiveness of BFM with Re-CORE for Residual Bi-Fusion object detection. Table III shows the ablation study of the PRB-FPN *vs.* YOLOv3 and YOLOv4 with or without BFM Re-CORE. As a result, PRB-FPN outperforms YOLOv3 in all categories. Note that the frame rates with or without BFM are very similar. For input size 512 × 512, YOLOv3 with BFM also outperforms YOLOv3 alone on all categories. BFM improves the detection of small objects significantly, with an increasing trend as the input image size increases. On the contrary, improvements on the large objects have a decreasing trend as the input size increases.

Fig. 4 shows the ablation study comparisons of object detectors regarding the effects of BFM and Re-CORE modules on a selected image from `COCO-test-dev`. Fig. 4(b) shows detections obtained by YOLOv3. Fig. 4(c) and (d) show detections of YOLOv3 with BF and Re-CORE modules, respectively. Fig. 4(e) shows detections of the proposed PRB-FPN. In comparison, Fig. 4(f) shows detections obtained by M2Det [28]. Observe clearly that the proposed PRB-FRN outperforms YOLOv3, YOLOv4 and M2Det.

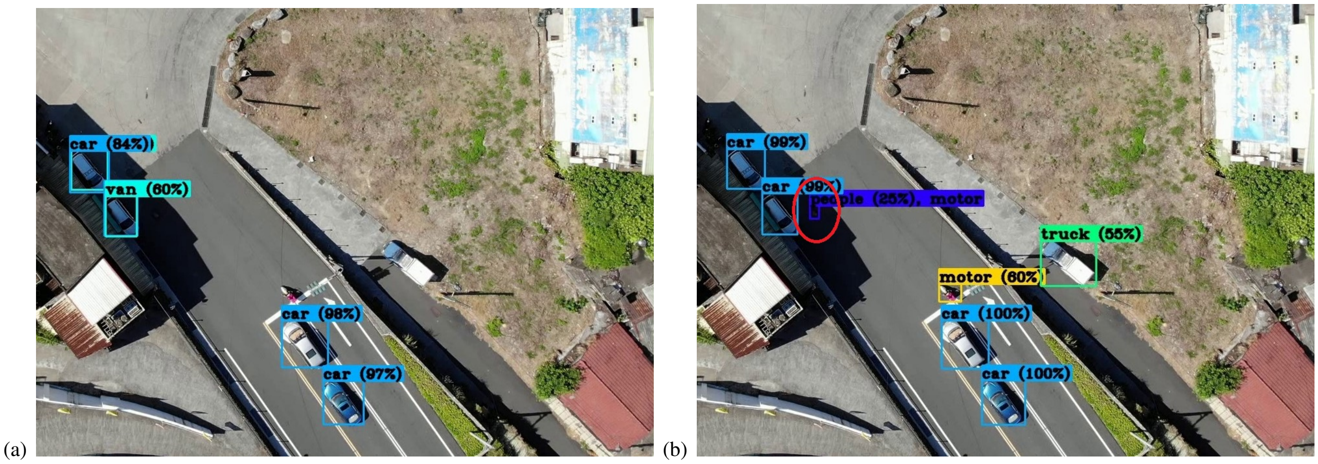


Fig. 8. Visualization of a failure case of PRB-FPN when compared with (a) YOLOv4 512 × 512. (b) shows the result of PRB-FPN 512 × 512, where a false negative detection is shown in a red circle.

TABLE VI
COMPARISONS ON THE MS COCO TEST-DEV SET WITH SOTA MODELS ON NVIDIA VOLTA V100.

Method	Backbone	Input size	FPS	AP	AP50	AP75	APS	APM	APL
YOLOv4 [13]	CSPDarknet-53 [47]	512x512	83	43	64.9	46.5	24.3	46.1	55.2
EfficientDet-D0 [11]	Efficient-B0 [49]	512x512	97.0	33.8	52.2	35.8	12	38.3	51.2
EfficientDet-D1 [11]	Efficient-B1 [49]	640x640	74.0	39.6	58.6	42.3	17.9	44.3	56
EfficientDet-D2 [11]	Efficient-B2 [49]	768x768	57.0	43	62.3	46.2	22.5	47	58.4
EfficientDet-D3 [11]	Efficient-B3 [49]	896x896	36.0	47.5	66.2	51.5	27.9	51.4	62.0
SM-NAS: E2 [50]		800x600	25.3	40	58.2	43.4	21.1	42.4	51.7
SM-NAS: E3 [50]		800x600	19.7	42.8	61.2	46.5	23.5	45.5	55.6
SM-NAS: E5 [50]		1333x800	9.3	45.9	64.6	49.6	27.1	49.0	58.0
NAS-FPN [12]	ResNet-50 [5]	640	24.4	39.9					
NAS-FPN [12]	ReNet-50 [5]	1024	12.7	44.2					
ATSS [51]	ResNet-101 [5]	800x	17.5	43.6	62.1	47.4	26.1	47	53.6
ATSS [51]	ReNet-101 [5]	800x	13.7	46.3	64.7	50.4	27.7	49.8	58.4
RB-FPN [Ours]	CSPDarknet-53 [47]	512x512	76.9	45.1	67.2	48.2	27.1	48.5	57
PRB-FPN [Ours]	CSPDarknet-53 [47]	800x800	37.5	**48.9**	**69.5**	**55.9**	**30.8**	**55.9**	**60.2**

Fig. 5 shows the comparisons of object detection against LRFNet [6] on a 1024 × 540 image from the UAVDT17 benchmark [45]. Note that the black masks in Fig. 5 come with the original images in UAVDT for privacy protection. LRFNet fails to detect the tiny far-away vehicles from the camera view, while PRB-FPN can successfully detect most of them.

D. PRB-FPN vs the Original FPN

We compare the performance the proposed PRB-FPN *vs.* the original FPN on the UAVDT [45] benchmark. Performance evaluation on the MS COCO dataset is omitted, as it contains very few samples of small objects. Table IV shows the performance comparisons with and without the proposal *parallel* or the *residual bi-fusion* modules. We adopted two backbones, namely MobileNet-V2 [46] and `CSPDarknet-53` [47] in this evaluation. `CSPDarknet-53` was created in our previous framework and is now adopted in YOLOv4. The baseline of FPN is the single bi-fusion module adopted in SoTA bi-fusion methods [25], [15], [14].

When the MobileNet-V2 backbone is used, accuracy of the baseline method (*i.e.* single bi-fusion module) is 29.7%. In comparison, as the Re-CORE module is added, the accuracy improves from 29.7% to 34.2%. However, the score is still lower than LRFNet [6], SpotNet [66], and CenterNet [67], since MobileNet-V2 is a very lightweight network. Finally, after the *parallel* FP bi-fusion design is included, the accuracy improves significantly from 34.3% to 65.47%, which outperform all comparison methods. Note that our PRB-FPN achieves double the amount of accuracy and triple amount of efficiency over RetinaNet [24].

Evaluation with the `CSPDarknet-53` backbone is shown in the last three rows of Table IV. Accuracy improvement is significant from 64.52 % to 76.55 %. These evaluations suggest that our *parallel* FP bi-fusion design is general for accuracy improvement. Table V shows the effects on accuracy and FPS of the number of feature pyramid layers. Two backbones, *i.e.*, `ResNet-50` and `CSPDarknet-53` were adopted in this evaluation. Clearly, with more FP layers, higher accuracy can be obtained for both backbones. Also the use of more FP layers results in lower FPS.

Fig. 6 shows visual comparisons of object detection between YOLOv4 and PRB-FPN. YOLOv4 cannot detect both large and small objects well enough. The enlargement of input image to detect small objects in YOLOv4 often fails in detecting large objects and undesired inefficiency. In Fig. 6(a), the cargo truck was missed by YOLOv4 but successfully detected by our PRB-FPN. In addition, YOLOv4 often detects

TABLE VII
COMPARISONS ON THE MS COCO TEST-DEV SET WITH SOTA MODELS ON NVIDIA GEFORCE TITAN X.

Method	Backbone	Input size	FPS	AP	AP50	AP75	APS	APM	APL
two-stage:									
Faster R-CNN w/ FPN [52]	VGGNet-16 [22]	1000 × 600	7.0	21.9	42.7	-	-	-	-
Faster R-CNN w/ FPN [52]	ResNet-101 [5]	1000 × 600	6.0	36.2	59.1	39.0	18.2	39.0	48.2
OHEM++ [53]	VGGNet-16 [22]	1000 × 600	7.0	25.5	45.9	26.1	7.4	27.7	40.3
R-FCN [54]	ResNet-101 [5]	1000 × 600	9.0	29.9	51.9	-	10.8	32.8	45.0
CoupleNet [55]	ResNet-101 [5]	1000 × 600	8.2	34.4	54.8	37.2	13.4	38.1	50.8
Cascade R-CNN [56]	ResNet-101 [5]	1280 × 800	7	42.8	62.1	46.3	23.7	45.5	55.2
Mask-RCNN [57]	ResNeXt-101 [5]	∼1280x800	3.3	39.8	62.3	43.4	22.1	43.2	51.2
Deformable R-FCN [58]	ResNet-101 [5]	1000 × 600	8	34.5	55	-	14	37.7	50.3
Deformable R-FCN [58]	Inc-Res-v2 [58]	1000 × 600	-	37.5	58	40.8	19.4	40.1	52.5
Fitness-NMS [59]	ResNet-101 [5]	1024 × 1024	5	41.8	60.9	44.9	21.5	45.0	57.5
SNIP [60]	DPN-98 [60]	-	-	**45.7**	**67.3**	**51.1**	**29.3**	**48.8**	**57.1**
one-stage low resolution:									
SSD [3]	VGGNet-16 [22]	300x300	43	25.1	43.1	25.8	6.6	25.9	41.4
RON [61]	VGGNet-16 [22]	384x384	15	27.4	49.5	27.1	-	-	-
DSSD [62]	ResNet-101 [5]	321x321	9.5	28.0	46.1	29.2	7.4	28.1	47.6
RFBNet [63]	VGGNet-16 [22]	300x300	66.7	30.3	49.3	31.8	11.8	31.9	45.9
YOLOv3 [2]	DarkNet-53 [2]	416x416	35	31.0	55.3	32.3	15.2	33.2	42.8
PFPNet-R [15]	VGG-16 [22]	320x320	33	31.8	52.9	33.6	12	35.3	46.1
RetinaNet [24]	ResNet-101 [5]	∼ 640 × 400	12.3	31.9	49.5	34.1	11.6	35.8	48.5
RefineDet [19]	VGGNet-16 [22]	320x320	38.7	29.4	49.2	31.3	10.0	32.0	44.4
RefineDet [19]	ResNet-101 [5]	320x320	-	38.6	59.9	41.7	21.1	41.7	52.3
M2Det [28]	VGGNet-16 [22]	320x320	33.4	33.5	52.4	35.6	14.4	37.6	47.6
M2Det [28]	ResNet-101 [5]	320x320	21.7	34.3	53.5	36.5	14.8	38.8	47.9
LRFNet [6]	VGGNet-16 [22]	300x300	76.9	32.0	51.5	33.8	12.6	34.9	47.0
LRFNet [6]	ResNet-101 [5]	300x300	52.6	34.3	54.1	36.6	13.2	38.2	50.7
one-stage high resolution:									
LRFNet [6]	VGGNet-16 [22]	512x512	38	36.2	56.6	38.7	19	39.9	48.8
LRFNet [6]	ResNet-101 [5]	512x512	31	37.3	58.5	39.7	19.7	42.8	50.1
EFIP [64]	VGGNet-16 [22]	512x512	34	34.6	55.8	36.8	18.3	38.2	47.1
RFBNet [63]	VGGNet-16 [22]	512x512	33	33.8	54.2	35.9	16.2	37.1	47.4
RFBNet-E [63]	VGGNet-16 [22]	512x512	30	34.4	55.7	36.4	17.6	37	47.6
SSD [3]	ResNet101 [5]	513x513	31.3	31.2	50.4	33.3	10.2	34.5	49.8
SSD [3]	VGGNet-16 [22]	512x512	22	28.8	48.5	30.3	10.9	31.8	43.5
DSSD [62]	ResNet101 [5]	513x513	5.5	33.2	53.3	35.2	13.0	35.4	51.1
YOLOv2 [4]	DarkNet-19 [4]	544x544	40	21.6	44	19.2	5	22.4	35.5
YOLOv4 [13]	CSPDarknet-53 [47]	512x512	31	43	64.9	46.5	24.3	46.1	55.2
YOLOv3-SPP [2]	DarkNet-53 [2]	608x608	19.8	36.2	60.6	38.2	20.6	37.4	46.1
YOLOv3-SPP [2]	DarkNet-53 [2]	608x608	19.8	36.2	60.6	38.2	20.6	37.4	46.1
RefineDet[19]	VGGNet-16 [22]	512x512	22.3	33	54.5	35.5	16.3	36.3	44.3
CornerNet [65]	Hourglass [65]	512x512	4.4	40.5	57.8	45.3	20.8	44.8	56.7
M2Det [28]	VGGNet-16 [22]	512x512	18	37.6	56.6	40.5	18.4	43.4	51.2
M2Det [28]	ResNet-101 [5]	512x512	15.8	38.8	59.4	41.7	20.5	43.9	53.4
RetinaNet [24]	ResNet-50 [5]	∼832x500	13.9	32.5	50.9	34.8	13.9	35.8	46.7
RetinaNet [24]	ResNet-101 [5]	∼832x500	11	34.4	55.7	36.8	14.7	37.1	47.4
RetinaNet+AP-Loss [24]	ResNet-101 [5]	512x512	11	37.4	58.6	40.5	17.3	40.8	51.9
ACoupleNet [36]	ResNet-101 [5]	600x1000	-	35.4	55.7	37.6	13.2	38.6	52.5
SAFNet [37]	ResNet-101 [5]	768x768	-	39.2	60.6	42.3	20.2	44.2	52.6
Cascade R-CNN [56]	ResNet-101 [5]	∼1280x800	7	42.8	62.1	46.3	23.7	45.5	55.2
FoveaBox [39]	ResNeXt-101 [5]	800x800	-	42.3	62.9	45.4	25.3	46.8	55.0
AB+FSAF [34]	ResNet-101 [5]	800	5.6	40.9	61.5	44	24	44.2	51.3
AB+FSAF [34]	ResNeXt-101 [5]	800	2.8	42.9	63.8	46.3	26.6	46.2	52.7
RB-FPN [Ours]	ResNet-50 [5]	512x512	32.1	44.3	65.1	48.2	25.1	47.3	56.8
PRB-FPN [Ours]	ResNet-50 [5]	800x800	15.9	46.1	67.3	49.9	28.5	49.3	59.4
RB-FPN [Ours]	CSPDarknet-53 [47]	512x512	27.3	**45.1**	**67.2**	**48.2**	**27.1**	**48.5**	**57**
PRB-FPN [Ours]	CSPDarknet-53 [47]	800x800	11.6	**48.9**	**69.5**	**55.9**	**30.8**	**55.9**	**60.2**

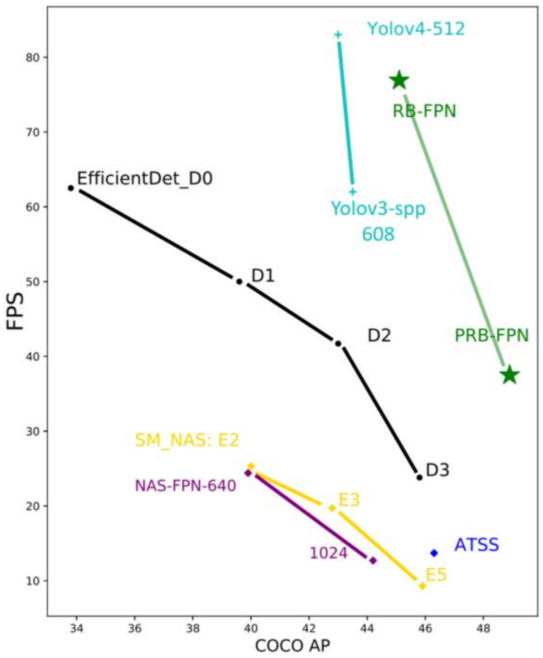


Fig. 9. AP vs. inference time on MS COCO detection.

a large object as several small ones. As shown Fig. 6(c), a truck was detected as two cars, the PRB-FPN can detect it successfully in Fig. 6(d). Note that the stop sign in Fig. 6(c) was missed by YOLOv4. Without the enlargement, YOLOv4 will further miss-detect or incorrectly classify small objects. For example, the small persons on the playground (highlighted in a red circle) in Fig. 7(a) were missed by YOLO 4, while PRB-FPN can successfully detect them in Fig. 7(b). Also, in Fig. 7(a), a small building was wrongly detected as a bus by YOLOv4. High detection rate and high recall rate for small objects are the major characteristics of our PRB model. Fig. 8(b) shows a failure case of our method. A false detection of a pedestrian (shown with a red circle) occurred due to the dark background. In comparison, YOLOv4 detection results in Fig. 8(a) is visually better without a false detection, however this is due to the weakness of YOLOv4 in identifying small objects. Such weakness of YOLOv4 can explain the miss detection of small vehicles and trucks in Fig. 8(a).

E. Comparisons with State-of-The-Art Models

Tables VI and VII compare the PRB-FPN with and without parallelization design against other SoTA object detectors in terms of accuracy and efficiency. Here experiments are conducted on two backbones of `CSPDarknet-53` [47] and `ResNet-50` [5] for the performance comparison of PRB-FPN. To make fair comparisons, we did not evaluate the performance of anchor-free methods as their efficiency scores were not reported. Inference time is calculated as the average of execution time of the network with Non-Maximum Suppression (NMS) from 999 random images.

Fig. 9 plots the inference time vs. *mean Average Precision small* (APs) [8] for many evaluated models, where the PRB-FPN was tested on nVidia Titan X. Observe that PRB-FPN (green curve in Fig. 9) achieves outstanding speed-accuracy performance compared to other SoTA models. We highlight two advantages of the PRB-FPN: (1) the parallel bi-fusion design for multi-scale feature extraction can detect both small and large objects at the same time with higher accuracy , and (2) the fusion module can effectively fuse both deep and shallow feature layers in parallel for fast and accurate one-shot object detection, specially for small objects. Observe that PRB-FPN outperforms the other SoTA one-stage object detectors (YOLOv4, YOLOv3, EfficientDet, ATSS, SM_NAS, and NAS-FPN), when taking both the accuracy and speed into account.

V. Conclusions

We present a new PRB-FPN model that can effectively fuse deep and shallow pyramid features for fast and accurate object detection. Our novel bi-directional residual FP design enables easy training and integration with popular backbones. The proposed bottom-up fusion improves the detection of both small and large objects. Extensive evaluations show that PRB-FPN outperforms other bi-directional methods and SoTA one-stage methods, in terms of accuracy and speed.

Future work includes the development of anchor-free methods that can avoid handcrafted anchors, which might further improves detection accuracy. Finally, Network Architecture Search (NAS) can potentially be adopted to find the better architecture, considering both the backbone and FP structures.

VI. Acknowledgement

The authors sincerely appreciate Mr. Yuwei Chen for proofreading and improving the English writing of this paper. We thank to National Center for High-performance Computing (NCHC) for providing computational and storage resources.

References

[1] T.-Y. Lin *et al.*, "Feature pyramid networks for object detection," in *Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

[2] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv*, 2018.

[3] W. Liu *et al.*, "SSD: Single shot multibox detector," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 21–37.

[4] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517–6525.

[5] K. He *et al.*, "Deep residual learning for image recognition," in *Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] T. Wang *et al.*, "Learning rich features at high-speed for single-shot object detection," in *IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[7] S. Alyamkin *et al.*, "Low-power computer vision: Status, challenges, and opportunities," *IEEE Journa on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 2, pp. 411–421, 2019.

[8] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision (ECCV)*, 2014.

[9] P. Sermanet *et al.*, "OverFeat: Integrated recognition, localization and detection using convolutional networks." in *International Conference on Learning Representations (ICLR)*, Jan. 2014.

[10] S. Liu *et al.*, "Path aggregation network for instance segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[11] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[12] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[13] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," in *arXiv*, 2020.

[14] S. Woo, S. Hwang, H.-D. Jang, and I. S. Kweon, "Gated bidirectional feature pyramid network for accurate one-shot detection," *Machine Vision and Applications*, vol. 30, pp. 543–555, 2019.

[15] S.-W. Kim *et al.*, "Parallel feature pyramid network for object detection," in *European Conference on Computer Vision (ECCV)*, September 2018.

[16] X. Wu *et al.*, "Single-shot bidirectional pyramid networks for high-quality object detection," *Neurocomputing*, vol. 401, pp. 1–9, 2020.

[17] W. R. J., L. Xiang, and L. C. X., "Pelee: A real-time object detection system on mobile devices," in *Neural Information Processing Systems*, 2018.

[18] M. Everingham *et al.*, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, p. 303–338, Jun. 2010.

[19] S. Zhang *et al.*, "Single-shot refinement neural network for object detection," in *Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4203–4212.

[20] H. Gao *et al.*, "Densely connected convolutional networks," in *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21] A. Khosla, N. Jayadevaprakash, B. Yao, and F. fei Li, "Fine-grained visual categorization," in *Computer Vision and Pattern Recognition Workshop*, 2011.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[24] T.-Y. Lin *et al.*, "Focal loss for dense object detection," in *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[25] D. Scherer, A. C. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition." in *ICANN (3)*, ser. Lecture Notes in Computer Science, vol. 6354. Springer, 2010, pp. 92–101.

[26] R. Zhang, "Making convolutional networks shift-invariant again," in *International Conference on Machine Learning (ICML)*, 2019.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition." in *European Conference on Computer Vision (ECCV)*, 2014.

[28] Z. Qijie *et al.*, "M2Det: A single-shot object detector based on multi-level feature pyramid network," *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[29] Z. Fu *et al.*, "Previewer for multi-scale object detector," *ACM international conference on Multimedia*, 2018.

[30] G.-J. Qi, "Hierarchically gated deep networks for semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2267–2275.

[31] D. Cao, Z. Chen, and L. Gao, "An improved object detection algorithm based on multi-scaled and deformable convolutional neural networks," *Human-centric Computing and Information Sciences*, vol. 10, pp. 1–22, 2020.

[32] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *European Conference on Computer Vision (ECCV)*, Sep. 2018.

[33] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *International Conference on Computer Vision (ICCV)*, October 2019.

[34] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," in *Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[35] H. Lee, S. Eum, and H. Kwon, "ME R-CNN: Multi-expert R-CNN for object detection," *IEEE Transactions on Image Processing*, vol. 29, pp. 1030–1044, 2020.

[36] Y. Zhu *et al.*, "Attention couplenet: Fully convolutional attention coupling network for object detection," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 113–126, 2019.

[37] Z. Jin *et al.*, "SAFNet: A semi-anchor-free network with enhanced feature pyramid for object detection," *IEEE Transactions on Image Processing*, vol. 29, pp. 9445–9457, 2020.

[38] J. Cao, Y. Pang, J. Han, and X. Li, "Hierarchical shot detector," in *International Conference on Computer Vision (ICCV)*, 2019.

[39] T. Kong *et al.*, "FoveaBox: Beyound anchor-based object detection," *IEEE Transactions on Image Processing*, vol. 29, pp. 7389–7398, 2020.

[40] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1019–1034, 2015.

[41] X. Shu, G.-J. Qi, J. Tang, and J. Wang, "Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation," in *Proceedings of the 23rd ACM International Conference on Multimedia*, 2015, p. 35–44.

[42] J. Tang, X. Shu, Z. Li, G.-J. Qi, and J. Wang, "Generalized deep transfer networks for knowledge propagation in heterogeneous domains," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 4s, 2016.

[43] X. Bu *et al.*, "Gaia: A transfer learning system of object detection that fits your needs," in *Computer Vision and Pattern Recognition (CVPR)*, 2021.

[44] P. Hu and D. Ramanan, "Finding tiny faces," in *Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[45] D. Du *et al.*, "The unmanned aerial vehicle benchmark: Object detection and tracking," in *European Conference on Computer Vision (ECCV)*, September 2018.

[46] M. Sandler *et al.*, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[47] C.-Y. Wang *et al.*, "CSPNet: A new backbone that can enhance learning capability of cnn," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 1571–1580.

[48] M. Naphade *et al.*, "The 4th AI City Challenge," in *Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020, p. 2665–2674.

[49] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning (ICML)*, 2019, pp. 6105–6114.

[50] L. Yao, H. Xu, W. Zhang, X. Liang, and Z. Li, "SM-NAS: Structural-to-modular neural architecture search for object detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 12 661–12 668, Apr. 2020.

[51] S. Zhang *et al.*, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[52] S. Ren *et al.*, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[53] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[54] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Neural Information Processing Systems*, vol. 29, 2016, pp. 379–387.

[55] Y. Zhu *et al.*, "CoupleNet: Coupling global structure with local parts for object detection," in *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[56] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[57] K. He *et al.*, "Mask R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[58] J. Dai *et al.*, "Deformable convolutional networks," in *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[59] L. Tychsen-Smith and L. Petersson, "Improving object localization with fitness NMS and bounded IoU loss," in *Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[60] B. Singh and L. S. Davis, "An analysis of scale invariance in object detection SNIP," in *Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[61] T. Kong *et al.*, "RON: Reverse connection with objectness prior networks for object detection," in *Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[62] C. Fu *et al.*, "DSSD : Deconvolutional single shot detector," *Arxiv*, 2017.

[63] S. Liu, D. Huang, and a. Wang, "Receptive field block net for accurate and fast object detection," in *European Conference on Computer Vision (ECCV)*, September 2018.

[64] Y. Pang *et al.*, "Efficient featurized image pyramid network for single shot detector," in *Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[65] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *European Conference on Computer Vision (ECCV)*, September 2018.

[66] H. Perreault, G. Bilodeau, N. Saunier, and M. Héritier, "SpotNet: Self-attention multi-task network for object detection," in *Conference on Computer and Robot Vision (CRV)*, 2020, pp. 230–237.

[67] K. Duan *et al.*, "CenterNet: Keypoint triplets for object detection," in *International Conference on Computer Vision (ICCV)*, October 2019.

Ping-Yang Chen is a Ph.D. student at the Department of computer science, Institute of Computer Science and Engineering, National Yang Ming Chiao Tung University (NYCU). He is a Ph.D. student with the Computer Science Department from 2019-present. He served as a computer vision advisor with the low-power embedded system for ELAN MICROELECTRONICS CORP from 2019-2020. His research interests include low-power computer vision, image processing, pattern recognition, and deep learning. He is a student member of IEEE.

Ming-Ching Chang is an Assistant Professor at the Department of Computer Science, College of Engineering and Applied Sciences (CEAS), University at Albany, State University of New York (SUNY). He was with the Department of Electrical and Computer Engineering from 2016 to 2018. He was an Adjunct Professor with the Computer Science Department from 2012-2016. During 2008-2016, he was a Computer Scientist at GE Global Research Center. He received his Ph.D. degree in the Laboratory for Engineering Man/Machine Systems (LEMS), School of Engineering, Brown University in 2008. He was an Assistant Researcher at the Mechanical Industry Research Labs, Industrial Technology Research Institute (ITRI) at Taiwan from 1996 to 1998. He received his M.S. degree in Computer Science and Information Engineering (CSIE) in 1998 and B.S. degree in Civil Engineering in 1996, both from National Taiwan University.

Dr. Chang's expertise includes video analytics, computer vision, image processing, and artificial intelligence. His research projects are funded by GE Global Research, IARPA, DARPA, NIJ, VA, and UAlbany. He is the recipient of the IEEE Advanced Video and Signal-based Surveillance (AVSS) 2011 Best Paper Award - Runner-Up, the IEEE Workshop on the Applications of Computer Vision (WACV) 2012 Best Student Paper Award, the GE Belief - Stay Lean and Go Fast Management Award in 2015, and the IEEE Smart World NVIDIA AI City Challenge 2017 Honorary Mention Award. Dr. Chang serves as Co-Chair of the annual AI City Challenge CVPR 2018-2021 Workshop, Co-Chair of the IEEE Lower Power Computer Vision (LPCV) Annual Contest and Workshop 2019-2021, Program Chair of the IEEE Advanced Video and Signal-based Surveillance (AVSS) 2019, Co-Chair of the IWT4S 2017-2019, Area Chair of IEEE ICIP (2017, 2019-2021), ICME (2021), and TPC chairs lead of MIPR (2022). He has authored more than 85 peer-reviewed journal and conference publications, 7 US patents and 15 disclosures. He is a senior member of IEEE.

Jun-Wei Hsieh put text here...

Yong-Sheng Chen (Member, IEEE) received the B.S. degree in computer and information science from National Chiao Tung University, Hsinchu, Taiwan, in 1993, and the M.S. and Ph.D. degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1995 and 2001, respectively. He is currently a Professor with the Department of Computer Science and the Director of the Computer Vision Research Center, National Yang Ming Chiao Tung University, Hsinchu, Taiwan. His research interests include biomedical signal processing, functional brain mapping, and computer vision. He is a senior member of IEEE. not sure yet