BMNet: A Reconstructed Network for Lightweight Object Detection via Branch Merging

Hefei Ling
lhefei@hust.edu.cn
Li Zhang
li_zhang@hust.edu.cn
Yangyang Qin
qinyangyang@hust.edu.cn
Yuxuan Shi
shiyx@hust.edu.cn
Lei Wu
leiwu@hust.edu.cn
Jiazhong Chen
jzchen@hust.edu.cn
Baiyan Zhang
zhangbyxy@hust.edu.cn

School of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, CN

Abstract

Object detection has made great progress in recent years along with the rapid development of deep learning. However, most current object detection networks cannot be used in the devices with limited computation power and memory resource, such as electronic chips, mobile phones, etc. To achieve an object detection network for resource-constrained scenario, this paper proposes a reconstructed Network for lightweight object detection via Branch Merging (BMNet). BMNet introduces an innovative and efficient architecture named 2-way Merging Lightweight Dense Block (2-way MLDB), which merges the duplicate parts of two branches in a dense block of the backbone network to obtain multi-receptive field features with fewer parameters and computations. In addition, to alleviate the decrease of accuracy caused by drastically reduced parameter size, BMNet builds an FPN-like SSD based on an Attention Prediction Block (APB) structure. Through extensive experiments on two classic benchmarks (PASCAL VOC 2007 and MS COCO), we demonstrate that BMNet is superior to the most advanced lightweight object detection solutions such as Tiny SSD, MobileNet-SSD, MobileNetv2-SSD and Pelee in terms of parameter size, FLOPs and accuracy. Concretely, BMNet achieves 73.48% of mAP on PASCAL VOC 2007 dataset with only 1.49 M parameters and 1.51 B FLOPs, which is the latest result with relatively low resource requirements and without pre-training to date.

1 Introduction

Object detection is one of the basic and important issues in the field of computer vision, since it combines object classification and localization within one scene, which is the basis for more complex and higher-level visual tasks such as segmentation, scene understanding, target tracking, etc. There exist two problems in traditional object detection methods which consist of three steps: region selection, feature extraction and classification regression. One is that the strategy of region selection is inferior and time-consuming; the other is that the manually extracted feature is less robust. With the rising of deep learning, a large number of CNN-based object detection methods were proposed, such as R-CNN [5], YOLO [20], SSD [16], and their variants [3, 4, 13, 18, 19, 21, 29], which have significantly improved the accuracy of object detection.

Although these networks have achieved very high accuracy, most of them require a lot of resources because of their high computation complexity and a large number of parameters. Huang *et al*. [9] have conducted extensive research on the trade-off between speed and accuracy, but resource limitation (consisting of computation limitation and memory limitation) has received little attention. Some terminal devices, such as smart chips and mobile phones, are difficult to store and run large models because of the resource scarcity. Therefore, most current object detection solutions are not suitable for this scenario.

In order to alleviate this limitation, many attempts have been made to design a lightweight object detection network. For example, Tiny SSD [26], the simplified version of SSD, compresses the SSD parameter size to 2.3 M and achieves 61.3% accuracy on PASCAL VOC 2007 dataset [2]. YOLOv3 also provides a light version called YOLOv3-tiny, which successfully reduces the parameter size of YOLOv3 to smaller than 10 M. Some detection frameworks based on lightweight classification network, such as MobileNet [7], MobileNetv2 [22], SqueezeNet [11], etc., also greatly reduce memory and computation requirements. For instance, MobileNet-SSD requires only 5.5 M parameters and 1.14 B FLOPs calculation on the PASCAL VOC 2007 dataset. Although these networks reduce the computation and memory resources to a certain extent, they also weaken the expression ability of features due to the sharp decrease of the number of parameters, and the detection accuracy is necessarily reduced. For example, On PASCAL VOC 2007 dataset, the accuracy of MobileNet-SSD (68%) is 9.2% lower than that of SSD (77.2%). Therefore, the existing lightweight object detection networks still do not achieve satisfactory performance with limited resource requirements, and it is necessary to develop a better lightweight object detection network.

We propose BMNet, designed to improve object detection accuracy while meeting strict constraints on memory and computation budget. The main problem we have to solve is how to extract features that are more suitable for the detection task using fewer parameters and calculations. The backbone of the proposed framework is inspired by some of the key design principals of Pelee [25] and the depthwise separable convolution structure. Pelee adopts a 2-way dense connection structure to acquire features of different receptive fields, which not only enriches the diversity of extracted features but also helps to detect small objects. BMNet incorporates an efficient depthwise separable convolution structure into 2-way dense connection structure, and introduces a novel 2-way Merging Lightweight Dense Block (2-way MLDB) to construct backbone network. This design greatly reduces the number of parameters while achieving the same rich features as the 2-way dense connection structure in Pelee. In order to alleviate the lack of precision caused by too few parameters in BMNet, we build an FPN-like SSD based on Attention Prediction Block (APB). APB is added after each feature map used for detection to obtain more discriminative features on each detection

scale, which can significantly improve the detection accuracy with a few extra parameters and less computation cost, especially with small objects.

We conduct extensive experiments to verify the effectiveness of BMNet on two classic datasets (PASCAL VOC 2007 and MS COCO), and compare our BMNet with other advanced lightweight object detectors such as Tiny SSD, MobileNet-SSD, MobileNetv2-SSD and Pelee. The results show that our BMNet has much better performance considering the memory and computation cost.

Our main contributions are summarized as follows:

1. We propose a 2-way MLDB, a novel and efficient network architecture that incorporates an efficient depthwise separable convolution structure into 2-way dense connection block structure and merges the same parts of two branches in the dense block for ultra-efficient computer vision applications.

2. We design an APB based FPN-like SSD for detection task to obtain the discriminative feature on each scale, which is a lightweight version of FPN [15] with an attention prediction block after each feature for detection.

3. We design the efficient object detection network BMNet based on the proposed 2-way MLDB and APB based FPN-like SSD. We show that our BMNet can train from scratch and achieve state-of-the-art performance on two standard benchmarks (PASCAL VOC 2007 and MS COCO) with a smaller number of parameters and less computation cost.

2 Related Work

Classic Object Detection Networks. Various of CNN based methods have been emerged in the field of general object detection in recent years. They could be divided into two major categories: region proposal based methods and proposal-free methods.

Typical region proposal based methods include R-CNN [5], Fast R-CNN [4], Faster R-CNN [21] and R-FCN [1]. R-CNN first utilizes selective search [23] to generate potential bounding boxes that may contain objects to be detected in the entire image, then evaluates the boxes with a classifier, and finally improves the bounding boxes by post-processing to eliminate duplicate detection. Fast R-CNN combines the feature extractor, classifier and regression of R-CNN, so that the training can be finished in one time and the consuming time will be decreased. Faster R-CNN introduces Region Proposal Network (RPN) instead of selective-search to enable the entire network to be end-to-end. R-FCN removes fully-connected layers and proposes a position-sensitive score map for final detection to improve speed and accuracy of the network. Region proposal based methods excessively pursuit high accuracy and ignore computation cost and speed. By using anchor mechanism, proposal-free methods like SSD [16] and YOLO [20] solve the problem of slow speed in region proposal based methods, and they can directly produce the class probability and position coordinate value of the objects. Proposal-free methods can balance speed and accuracy but ignore parameter size, which are usually faster than region proposal methods with less accuracy, while still not available for resource-constrained scenarios.

Lightweight Object Detection Networks. Recently, many works have been devoted to designing lightweight object detection networks for resource-constrained scenarios. Some networks use well-designed structures to reduce the number of parameters and calculation consumption, while ensuring detection accuracy. For example, Pelee [25] introduces a 2-way dense layer structure based backbone to reduce computation cost while maintaining detection accuracy for mobile applications. SqueezeDet [28] utilizes a backbone network

based on a variant of SqueezeNet [11] structure to achieve an object detection network with lower computation cost, higher accuracy and higher speed.

Meanwhile, depthwise separable convolution is widely used in lightweight image classification networks[6, 7, 17, 22] with a few parameters and efficient computation power, which has also led to a number of lightweight object detection networks based on SSD framework such as MobileNet-SSD and MobileNetv2-SSD. MobileNet-SSD compresses the parameter size to 5.5 M while achieving 68% mAP on PASCAL VOC 2007 dataset.

Nevertheless, the performance of lightweight object detection networks still has a lot of room for improvement. There is still a large accuracy gap between the lightweight object detection network and the corresponding conventional detection network, especially for small objects. For example, Tiny YOLOv3 achieves 33.1% mAP on COCO dataset [14], while YOLOv3 can reach 51.5% mAP under the same setting. Tiny-DSOD [26] reaches 4.3% APs (small object detection accuracy) on COCO dataset, while DSOD achieves 9.4% APs under the same setting. This observation inspires us that there might be a higher-accuracy lightweight object detection network structure with less resource consumption.

Attention mechanism. Attention mechanism has been widely used in various computer vision tasks based on deep learning in recent years. Many works have shown that the attention mechanism allows the network to focus on the most informative areas rather than the entire image, thereby improving the performance of the network [8, 24, 27]. Wang *et al*. [24] propose Residual Attention Network, whose stacked network structure based on attention residual learning can easily optimize and learn the deeper network model. Hu *et al*. [8] introduce Squeeze-and-Excitation network, which improves accuracy by modeling correlations among feature channels. Sanghyun Woo *et al*. [27] propose Convolutional Block Attention Module (CBAM), it refines the attention-based features into channel and spatial modules, and enables significant performance improvements while maintaining small overhead.

Attention mechanism helps the network choose better intermediate features and improve the performance with few extra parameters and computations, which is exactly what we need to design a lightweight network. Therefore, our BMNet adopts a parallel channel and spatial attention module to improve detection accuracy while maintaining small overhead.

3 BMNet

We intend to design a relatively efficient object detection network for resource-constrained scenarios. The overview of BMNet proposed in this paper is shown in Figure 1. Our BMNet method is a multi-scale proposal-free detection framework, which is based on the Single Shot Detector (SSD) [16] framework. It consists of two main parts: the backbone part and the front-end part. Next, we will introduce these two parts in Section 3.1 and 3.2.

3.1 Multi-Receptive fields Dense Block Based Backbone

Inspired by DenseNet [10], we construct a DenseNet-like backbone since it is easier to train from scratch, and can fully utilize the features thanks to the dense connection structure. In addition, many works have shown that the large receptive field of features have a good detection effect on large objects, and the small receptive field of features are benefit for small objects' detection. Therefore, the backbone part of the BMNet uses a variant of the 2-way dense block like Pelee [25] to obtain the multi-receptive field features. Taking the resource

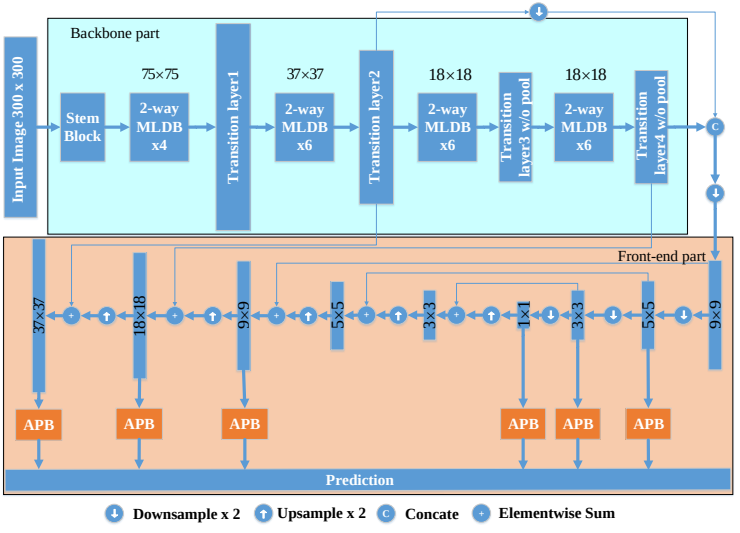


Figure 1: The network structure of BMNet. The input image size is 300×300. We use proposed 2-way MLDB to build the backbone part, which outputs a feature with a dimension of 18×18. And then, we perform a series of downsample, bilinear interpolation upsample and elementwise sum to obtain fusion features with different dimensions. Finally, we add attention prediction block after each different dimension features for detection.

constraints into consideration, we try to utilize the depthwise convolution and dilation convolution into the 2-way dense block, which is named as 2-way Lightweight Dense Block (2-way LDB).

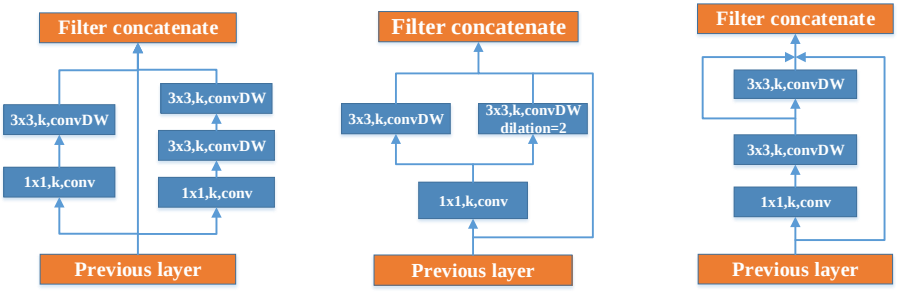
We propose three types of 2-way LDB units, i.e., 2-way LDB-a, 2-way LDB-b and 2-way MLDB, as shown in Figure 2. Considering the limitation of parameters and computation capacity, we adopt different strategies in each of the three units to obtain features for multi receptive fields.

2-way LDB-a unit in Figure 2(a) is inspired by the 2-way dense layer in Pelee [25]. It adopts two completely separate ways to get different scales of receptive fields. One way uses one 3×3 depthwise convolution to capture small-size objects, and the other utilizes two staked 3×3 depthwise convolutions to learn semantic features for large objects.

Dilation convolution is also widely used to reduce the number of convolutions while maintaining large receptive field, which is also a way to reduce resource consumption. Therefore, we propose 2-way LDB-b unit, as shown in Figure 2(b), which utilizes a 3×3 dilation depthwise convolution to obtain large receptive field feature and a 3×3 standard depthwise convolution to get small receptive field feature. The two branches share the 1×1 convolution to reduce parameters.

In order to obtain multi-receptive field features, most multi-way structures use completely independent branches regardless of the duplication among the branches, when stacking these structures in the network, the duplicate parts will bring many parameters and much computation cost. For instance, there are duplicate parts in the two branches of the 2-way LDB-a, we can effectively maintain the feature expression ability with fewer parameters and

less computation cost by merging these parts. With this consideration, we design another type of 2-way LDB unit as shown in Figure 2(c), which merges the duplicate parts in the two branches of the 2-way LDB-a and obtains the same rich features as 2-way LDB-a on the basis of reducing a branch structure, we call it as 2-way Merging Lightweight Dense Block (2-way MLDB).



(a) 2-way LDB-a (b) 2-way LDB-b (c) 2-way MLDB

Figure 2: Illustrations of the 2-way Lightweight Dense Block (LDB). (a) incorporates depth-wise separable convolution to reduce the parameters in Pelee's 2-way dense layer. (b) introduces dilation convolution to build multi-receptive fields block. (c) merges the duplicate parts in (a) to further reduce the parameters while maintaining rich feature representation.

Through the experiment in Section 4.2, we confirm that 2-way MLDB consumes fewer resources and has a relatively high accuracy compared with other two units, and it is more suitable for lightweight object detector. Therefore, we choose 2-way MLDB as the basic unit to build our backbone network.

Table 1 shows our backbone network, which consists of a stem block and a feature extractor. The stem block is the same as Tiny-DSOD [12], which implements the first downsample of the input image spatial dimensions and the increase in the number of channels, and ensures strong feature expression ability without incurring too much extra calculation. Four dense stages of the feature extractor are followed by translation blocks. The last layer in the first two translation blocks is average pool layer with stride 2. Each dense stage contains several 2-way MLDBs. Convolution layers in 2-way MLDB are followed by a BatchNorm layer and a ReLU layer, which make it easier to train. We also gradually increase the number of inter-channels in the 2-way MLDB of four dense stages to reduce computation cost, because the shallower large spatial size requires more computations.

3.2 FPN-like SSD based on Attention Prediction Block

FPN-like SSD. The plain front-end of SSD is limited when using low-level features to predict small objects, i.e., SSD is less effective for detecting small objects due to the lack of high-level semantic features. To overcome this problem, we use the feature pyramid method to combine high-level features and low-level features. Many works [3, 15] prove that feature pyramid method can greatly improve the detection accuracy of small objects by combine high-level features and low-level features. Most of them choose deconvolution when performing feature fusion upsample. Although effective, it also brings huge parameter size and computation cost. We use the method in Tiny-DSOD [12] to build detection front-end, which utilizes a simple bilinear interpolation to upsample features for feature fusion, and uses all

Module name		Output size (Input 3×300×300)	Component
Stem Block	Convolution	64×150×150	3×3 conv, stride 2
	Convolution	64×150×150	1×1 conv, stride 1
	Depthwise Convolution	64×150×150	3×3 convDW, stride 1
	Convolution	128×150×150	1×1 conv, stride 1
	Depthwise Convolution	128×150×150	3×3 convDW, stride 1
	Pooling	128×75×75	2×2 max pool, stride 2
Extractor	Stage 0	384×75×75	2-way MLDB **x 4**
	Translation Block 0	128×37×37	1×1 conv, stride 1 2×2 max pool, stride 2
	Stage 1	704×37×37	2-way MLDB **x 6**
	Translation Block 1	128×18×18	1×1 conv, stride 1 2×2 max pool, stride 2
	Stage 2	896×18×18	2-way MLDB **x 6**
	Translation Block 2	256×18×18	1×1 conv, stride 1
	Stage 3	1216×18×18	2-way MLDB **x 6**
	Translation Block 3	64×18×18	1×1 conv, stride 1

Table 1: Overview of BMNet backbone architecture. In the "Component" column, the symbol "x" after block name indicates that block repeats number times given after the symbol.

the fused features for detection.

Noting that the original intention of feature fusion is to enrich shallow features to detect more small objects. Since the deep features already contain rich global semantic information, it may incur some noise when introducing feature fusion. We also find in experiments that the deeper feature fusion structure does not help the performance improvement, on the contrary, when the training is insufficient, it may incur the performance decline. Therefore, we select the former three shallow fusion features and the latter three deep features for detection, we call it as FPN-like SSD, as shown in Figure 1.

Attention Prediction Block. In order to obtain more discriminative features for specific detection layer, we introduce the attention mechanism in our BMNet. For each feature map used for detection, we build an Attention Prediction Block (APB) before conducting prediction, whose location is described in Figure 1, it is more efficient for obtaining discriminative detection features than other locations. The structure of APB is inspired by CBAM [27], but we use a series of convolutional layers to obtain the attention map of channel and spatial respectively, and use channel and spatial attention map in parallel to reduce the inference time. For an input feature F, we first calculate the attention map according to two independent dimensions: channel ($M_c(F)$) and spatial ($M_s(F)$), which utilize the combined characteristics of average pool and max pool, this is more efficient than using each independently. And then, we obtain the attention map through softmax and sigmoid respectively. Finally, we multiply the two attention maps by element-wise to the input feature map for adaptive feature refinement ($R(F)$) at the same time. In short, the attention and refined feature are

computed as:

$$M_c(F) = Softmax(Convs(AvgPool(F) + MaxPool(F))) \quad (1)$$
$$M_s(F) = Sigmoid(Convs(AvgPool(F); MaxPool(F))) \quad (2)$$
$$R(F) = F \otimes M_c(F) \otimes M_s(F) \quad (3)$$

4 Experiments

4.1 Implementation Details

We implement our work basing on the PyTorch framework. All of our models are trained from scratch with SGD solver on NVIDIA Titan Xp GPU. Most of our training settings follow SSD [16] (e.g., data augmentation, scale, aspect ratios for default boxes and loss function), while learning rate schedule is slightly changed to adapt to training from scratch. We conduct experiments on the widely used PASCAL VOC and MS COCO datasets that have 20 and 80 object categories respectively. Object detection performance is measured by mean Average Precision (mAP).

Row	LDB-a	LDB-b	MLDB	APB	Params	FLOPs	mAP (%)
(1)					5.43 M	1.21 B	70.9
(2)	✓				1.96 M	1.91 B	73.02
(3)		✓			1.39 M	1.45 B	71.85
(4)			✓		1.47 M	1.50 B	72.31
(5)	✓			✓	2.13M	1.99B	74.33
(6)		✓		✓	1.40M	1.46B	72.32
(7)			✓	✓	1.49 M	1.51 B	**73.48**

Table 2: Ablation Study on PASCAL VOC 2007 test set. We use an FPN-like SSD front-end in each detection network. Row (1) indicates that the network uses the original 2-way dense block in Pelee. LDB-a, LDB-b and MLDB in the header represent the 2-way LDB-a, 2-way LDB-b and 2-way MLDB mentioned above, respectively (for simplicity). The tick "✓" means the corresponding configuration is adopted in the object detection network (row-wise), otherwise not.

4.2 Ablation study on PASCAL VOC 2007

We first investigate the design of three types of 2-way LDB in our multi-receptive fields dense block based backbone. We train our model on the union of PASCAL VOC 2007 and PASCAL VOC 2012 trainval dataset called PASCAL VOC 07+12 trainval dataset, and test on the PASCAL VOC 2007 test set. The batch size is set to 32. The number of BMNet training epochs is 800, we set the initial learning rate to 0.01, then it decreases by a factor of 10 every 200 epochs. The results of the experiment are summarized in Table 2, which show that 2-way MLDB based backbone is better than other units under similar resource cost. Compared with row (1), who uses the original 2-way dense block in Pelee, 2-way MLDB based model achieves 1.43% higher mAP with only 1/4 parameters. Comparing the 2-way

LDB-b and 2-way MLDB in row (3) and (4), whose parameter sizes are smaller than 1.5 M, 2-way MLDB gains 0.46% mAP higher than 2-way LDB-b with only 0.05 B FLOPs rise.

We further investigate the effectiveness of our APB. Comparing row (2) and row (5) , row (3) and row (6), row (4) and row (7) in Table 2 respectively, it is obvious that after adding the APB module, models with different 2-way LDB modules have improved significantly in accuracy, while the FLOPs and parameters have not increased too much. In particular, comparing the row (4) and row (7), APB brings 1.17% mAP gains, introducing only 0.02 M parameters and 0.01 B FLOPs. It is worthwhile to build an attention block after each feature map used for detection.

4.3 Benchmark Results on PASCAL VOC 2007

The training strategy is exactly the same as the ablation study part. As shown in Table 3, the accuracy of our BMNet is 73.48%, which is better than other lightweight detectors with relatively small parameter size and low computation cost. For example, BMNet reduces the parameter size to about 1/4 and gains 2.58% accuracy boost with only 0.3 B FLOPs rise compare with Pelee. Note that our BMNet is trained from scratch. However, the inference speed of BMNet is not as good as other detectors, which is due to the lack of optimization of depthwise separable convolution in PyTorch. BMNet achieves 77.05% mAP taking the model trained on COCO trainval35k as described in Section 4.4 and fine-tuning it on the 07+12 dataset, which is the best result of the lightweight model to date.

Method	Input size	data	FPS	Params	FLOPs	mAP (%)
Tiny SSD [26]	300×300	07+12	-	2.30 M	0.57 B	61.3
MobileNet-SSD	300×300	07+12	59	5.77 M	1.15 B	68.00
Pelee* [25]	304×304	07+12	35	5.43 M	1.21 B	70.9
BMNet (ours)	300×300	07+12	30	1.49 M	1.51 B	**73.48**
MobileNet-SSD	300×300	07+12+COCO	-	5.77 M	1.15 B	72.7
Pelee*	304×304	07+12+COCO	35	5.43 M	1.21 B	76.4
BMNet (ours)	300×300	07+12+COCO	30	1.49 M	1.51 B	**77.05**

Table 3: Results on PASCAL VOC 2007. Data 07+12 means training on the PASCAL VOC 07+12 trainval dataset, data 07+12+COCO means first to train on COCO trainval35k, then fine-tune on 07+12. "*" means the results are reimplemented on the framework of PyTorch.

4.4 Benchmark Results on COCO

Finally, we evaluate our method on the MS COCO dataset. For fair comparison, we follow the common train settings, train our network on trainval35k dataset consisting of 80k training images and 35k validation images, and evaluate on a set of 5k validation images called minival. We also report the final results on a set of 20k test images (test-dev). The batch size is set to 128. For stable training, we introduce the warmup learning strategy to train the network from scratch, the learning rate first gradually increases from 10^{-6} to 10^{-2} in the first 5 epochs, and then divided by 10 every 66 epochs. The total number of training epochs is 350. Other training settings are the same as training COCO in SSD.

The test results are summarized in Table 4. BMNet achieves 39.1% mAP on the test-dev dataset in metric of AP@IOU [0.5], which outperforms MobileNet-SSD, MobileNetv2-SSDLite and Pelee with fewer parameters. Meanwhile, BMNet has a detection accuracy of 6.5% with small objects, which is almost twice that of Pelee, while the parameter size is 1/2 of Pelee and the FLOPs are equivalent to Pelee. The detection performance on small objects of BMNet is even comparable to heavyweight detectors like SSD and YOLOv2.

Method	Input size	FLOPs	Params	AP (%), IOU		APs (%) , Area S
				0.5:0.95	0.5	
SSD [16]	300×300	34.36 B	34.30 M	25.1	43.1	6.6
YOLOv2 [18]	416×416	17.50 B	67.43 M	21.6	44.0	5.0
MobileNet-SSD	300×300	1.20 B	6.80 M	18.8	-	-
MobileNetv2-SSDLite	300×300	0.80 B	4.30 M	22.1	-	-
Pelee [25]	304×304	1.29 B	5.98 M	22.4	38.3	3.7
BMNet (ours)	300×300	1.81 B	2.70 M	**22.0**	**39.1**	**6.5**

Table 4: COCO test-dev 2018 detection results.

5 Conclusion and Future Work

This paper proposes a lightweight object detection method, named BMNet. We achieve a better object detector with fewer parameters and less computation cost through an innovative 2-way MLDB and an APB based FPN-like SSD. We validate the effectiveness of the 2-way MLDB and detectors presented in this paper through extensive ablation studies. We also compare BMNet with state-of-the-art lightweight object detection models such as MobileNet-SSD, MobileNetv2-SSD and Pelee on two classic benchmarks (PASCAL VOC 2007 and MS COCO). The results show that BMNet outperforms each of the benchmarks in terms of all three indicators (accuracy, FLOPs and parameter size). In particular, BMNet achieves 73.48% mAP on PASCAL VOC 2007 with only 1.49 M parameters and 1.51 B FLOPs of computation cost. This is the latest technical result with such low resource requirements so far. BMNet is only used for lightweight object detection currently, but its lightweight network design approach can also be extended to lightweight object tracking in the field of automatic driving in the future.

References

[1] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 379–387. Curran Associates, Inc., 2016.

[2] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[3] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[4] Ross Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[5] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[6] M. G. Hluchyj and M. J. Karol. Shuffle net: an application of generalized perfect shuffles to multihop lightwave networks. *Journal of Lightwave Technology*, 9(10): 1386–1397, Oct 1991. ISSN 0733-8724. doi: 10.1109/50.90937.

[7] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[8] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[9] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[10] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.

[11] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[12] Yuxi Li, Jiuwei Li, Weiyao Lin, and Jianguo Li. Tiny-dsod: Lightweight object detection for resource-restricted usages. *arXiv preprint arXiv:1807.11013*, 2018.

[13] Zuoxin Li and Fuqiang Zhou. Fssd: feature fusion single shot multibox detector. *arXiv preprint arXiv:1712.00960*, 2017.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[17] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[18] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[19] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[20] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[22] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[23] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104 (2):154–171, 2013.

[24] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[25] Robert J. Wang, Xiang Li, and Charles X. Ling. Pelee: A real-time object detection system on mobile devices. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1963–1972. Curran Associates, Inc., 2018.

[26] A. Womg, M. J. Shafiee, F. Li, and B. Chwyl. Tiny ssd: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 95–101, May 2018. doi: 10.1109/CRV.2018.00023.

[27] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[28] Bichen Wu, Forrest Iandola, Peter H Jin, and Kurt Keutzer. Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 129–137, 2017.

[29] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4203–4212, 2018.