A keypoint-based object detection method with wide dual-path backbone network and attention modules

Zhong Qu¹ | Run Zhang¹ | Kang-hua Bao²

¹ The College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, No.2, Chongwen Road, Nan'an District, Chongqing, People's Republic of China
² The Institute of Information Technology, Southwest Computer Co., Ltd., No.1, Guangdian Road, Nan'an District, Chongqing, People's Republic of China

Correspondence
Zhong Qu, the College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, No.2, Chongwen Road, Nan'an District, Chongqing, People's Republic of China, 400065.
Email: quzhong@cqupt.edu.cn

Funding Information
This work was supported by National Natural Science Foundation of China under Grant 61701060.

Abstract
Keypoint-based object detection is one of the most efficient and speedy methods at present, yet its performance is often worse than the anchor-based method. Without prior settings in the keypoint-based method, the huge search space of the keypoints results in high recall but low precision. In this paper, the wide dual-path backbone network is introduced as a feature extractor to extract richer original information, which has fewer parameters and better classification performance. Then, the attention fusion module is designed to effectively fuse the dual-path with the consideration of the respective advantages of the residual-path and the densely connected path. In order to provide more accurate pixel-level information for keypoint prediction, the upsampling dual-attention module is proposed to recover the spatial size of the feature map, which integrates multi-scale of channel-wise and spatial attention. Compared with other state-of-the-art detectors, this method has achieved accuracy-efficiency results with fewer parameters, lower FLOPs, and smaller model size. Experimental results show that the proposed wide dual-path backbone network has achieved 4.98% top1-error on the CIFAR-10 classification dataset. On the PASCAL VOC object detection dataset, this model has achieved an accuracy-efficiency tradeoff result of 78.3% mAP at the speed of 41 FPS.

1 INTRODUCTION

Object detection, as one of the most fundamental and challenging tasks in computer vision, aims to localize all possible objects in images and provide their class-label confidences, has been rapidly advanced with the development of convolutional neural networks. One of the most common components of state-of-the-art detectors is anchor boxes, which are predefined and placed with various sizes and aspect ratios, and regressed to ground-truth places.

Anchor-based detectors often densely place pre-designed anchor boxes on feature maps to ensure sufficient overlap with most of ground-truth boxes. Liu et al. proposed single shot multiBox detector (SSD) [1] to place more than 8k pre-designed anchor boxes, while more than 100k in RetinaNet [2], on feature maps to overlap with ground-truth boxes. Ren et al. [3] set 3 scales and 3 width and height ratios of anchors at each point in Faster R-CNN. Redmon et al. [4] found another way to learn the anchor settings through the K-Means cluster method in YOLOv2. Yu et al. [5] designed the anchor cascade framework to improve anchor quality. Kong et al. [6] proposed FoveaBox to directly predict bounding boxes of objects. Li et al. [7] introduced the pose anchor boxes to utilize the prior knowledge. Nevertheless, only a few anchors will overlap with ground-truth and the hyper-parameters of anchors differ detection performance, which will create tremendous computational wastage and an imbalance between positive and negative anchors.

Keypoint-based object detection is a sort of approach that generates object bounding boxes by detecting their keypoints without any predefined anchor boxes, as shown in Figure 1. Law et al. [8] proposed CornerNet to find top-left and bottom-right bounding box corners as a pair of keypoints and predict embedding vectors to group a pair of corners belong to the same object. Duan et al. [9] developed CenterNet-Triplets based on CornerNet to predict both corner points and centre points of objects. Zhou et al. [10] introduced ExtremeNet to detect the top-most, left-most, bottom-most, right-most, and centre points of all objects; and group these extreme points...
into objects with a geometry-based approach. Zhou et al. [11] presented CenterNet to detect objects by a single point as their bounding box centre. It uses a fully convolutional network [12], to generate heatmaps of all categories for the centres of all instances in an image, and then directly predicts their heights and widths. Besides, the keypoint-based method can extend to other tasks with minor efforts, such as object tracking [13–15], pose estimation [16–18] and 3D object detection [19–21].

The performance of detectors is tremendously affected by the backbone network, which results in the backbone network structure design has been one of the most important research. Karen et al. [22] provided a simple and practical backbone network structure design guidance in VGGNet, which stacks blocks with the same shape. ResNet [23] proposed by He et al. and DenseNet [24] proposed by Gao et al. are the two most popular backbone networks in recent years. ResNet adds the output directly to the input while DenseNet concatenates the output to the input. Chen et al. [25] presented DPN by combining the advantages of ResNet and DenseNet, which remarkably reduced computational complexity and improved performance. Recently, more and more lightweight networks [26–28] are proposed to satisfy the need for real-time or mobile detection. Depth-wise convolution [29] is wildly used among them to effectively reduce the number of parameters.

In recent years, the attention mechanism has become one of the important methods to improve the performance of the network. As shown in Figure 2, the attention mechanism instructs the network where to focus along with enhancing the expression of features in key-areas. Hu et al. [30] proposed SENet to model the importance of channels. Cao et al. [31] proposed GCNet via combining the advantages of the non-local global context modelling capability of NLNet [32] and the computation-saving ability of SENet. In the convolutional bottleneck attention module (CBAM) [33], the authors exploited both spatial attention and channel attention via two lightweight attention modules. Li et al. [34] developed SKNet to generate the attention map across two network branches. Some networks [35] intensify the
feature expression by fusing the features of the adjacent stage but ignore diverse representations of different features.

In this paper, we present DPANet for object detection, based on the encoding-decoding framework, shown in Figure 3. In DPAnet, it first extracts features from the input image via the wide dual-path backbone network at the decoding stage. Then the AFM is used to generate the channel-wise attention map and fuse the dual-path of the backbone network for the preparation of the decoding stage. At the decoding stage, three UDM are stacked to gradually recover the feature map with a combination of the low-level features from the encoding stage. At last, three convolution predictors are assigned to predict the results. Our main contributions are as follows:

1. In this paper, we propose a wide dual-path backbone network by introducing depth-wise convolution, which has fewer parameters and achieves better performance on the classification dataset.
2. We build the lightweight AFM to effectively develop and fuse the respective advantages of residual-path and densely connected path. Then, three UDM are stacked to generate precise pixel-level information for spatial size recovering.
3. Combining the wide dual-path backbone network with AFM and UDM, we build a novel keypoint-based dual-path attention network for object detection. Experiments show that our model has achieved accuracy-efficiency tradeoff results on the PASCAL VOC object detection dataset.

The rest of this paper is organized as follows. In Section II, we briefly introduce the related works about three main parts of this paper. Section III describes the wide dual-path backbone network in detail. Attention modules are explained in Section IV. Section V illustrates the dual-path attention network architecture. Section VI gives the experimental comparison results of the proposed model with the state-of-the-art models and ablation studies. Section VII draws the conclusion of this paper.

2 Related Work

2.1 Keypoint-based object detection

Keypoint-based detectors are usually more computationally efficient than anchor-based detectors while having a competitive performance of accuracy. Both CornerNet and ExtremeNet have constructed robust keypoint-based detectors. However, the grouping stage significantly reduces its time efficiency. CenterNet directly predicts the center point and height and width per object without grouping stage or post-processing. This makes it very fast and effective while competitive with other detectors. Besides, its method is general and can be extended to other tasks with minor effort.

2.2 Backbone network architecture

The performance of the backbone network relies on well-designed network architecture. ResNet, which uses skip connections to push the depth of the backbone network to hundreds of layers, has greatly promoted the development of backbone network architecture. DenseNet, which also uses skip connections, concatenates the input to the output instead of adding. In DPN, the authors provide a deeper understanding of relations between residual networks and densely connected networks and...
QU ET AL. present a novel dual path network architecture that has achieved better accuracy but lower computing complexity.

2.3 Attention mechanism

The performance of the backbone network has been significantly improved by the attention mechanism. SENet designs the SEBlock to get channel-wise attention by explicitly modelling the interdependence between channels. To achieve better results on various computer vision tasks, GCNet designs GCBlock to combine the non-local global context modelling capability of NLNet with the computation-saving ability of SENet. Nevertheless, spatial attention is neglected in these approaches. In CBAM, the authors construct two effective attention modules to exploit both spatial attention and channel attention. Furthermore, they also verify that the use of both spatial and channel-wise attention is superior to simply use channel-wise attention.

3 WIDE DUAL-PATH BACKBONE NETWORK

Our proposed backbone network consists of an input stem and four subsequent stages, which have been shown in Appendix A. At the input stem, most of the networks use a $7 \times 7$ kernel size convolution with a stride of 2 to get large receptive field, which is more expensive than $3 \times 3$ kernel size convolution. In our proposed backbone network, we replace the original $7 \times 7$ kernel size convolution with three $3 \times 3$ kernel size convolutions and set the stride to 1. This strategy resolves the input image size not to drop too fast at the input stem, and more information can be obtained with the same receptive field. Then, we use various settings of channels to achieve different accuracy-efficiency tradeoff. Other models, such as InceptionV2 [36] and ShuffleNet, have likewise implemented this strategy to keep the same receptive field while having better accuracy but fewer parameters.

Group convolution is first proposed in AlexNet [37] to solve the problem of insufficient GPU memory. The authors found that group convolution could dramatically reduce the number of parameters and make the model not overfit easily. Depthwise convolution, first introduced in ref. [29], is a kind of special grouping convolution with the same number of groups, input channels, and output channels. It tremendously decreases the number of parameters and lessens computational complexity compared to the standard convolution operation. Hence, we replace the standard $3 \times 3$ convolution in each stacked block with $3 \times 3$ depth-wise convolution to lessen the overall parameters and computational complexity of the entire backbone network.

At the downsample block, DPN used a $1 \times 1$ convolution with a stride of 2 to downsample the feature map as shown in Figure 4. That method will result in the loss of the input spatial information according to Ref. [38]. To minimize spatial information loss, we add a $2 \times 2$ average pooling with the stride of 2 before $1 \times 1$ convolution and changed the stride of $1 \times 1$ convolution from 2 to 1. The performance will be improved in this way while with little additional computational cost. The detailed calculation process of wide dual-path block is given in Appendix B.

Based on the DPN structure, we redefine the numbers of stacked blocks in each stage and the depth of the whole backbone network is consistent with ResNet-50. However, insufficient numbers of stacked blocks will severely affect the performance of the densely connected path. To tackle this issue, we widen the width of each dual-path block, which we call it the wide dual-path block. In detail, we increase the channels of convolution kernel at each wide dual-path block and the width increment of the densely connected path. We customize different scales of channel width using Wide_DPN-50_1.0 $\times$ as the baseline. Wide_DPN-50_1.5 $\times$ means scaling the number of channels in Wide_DPN-50_1.0 $\times$ by 1.5 times and Wide_DPN-50_2.0 $\times$ by 2 times.

4 ATTENTION MODULES

4.1 Attention fusion module

At the end of the encoding stage, it is necessary to fuse the dual-path of the backbone network into one merged path since the dual-path is not available at the decoding stage. As discussed in ref. [25], the residual-path is capable of reusing features, while the densely connected path is adept in discovering new or redundant features. In essence, fusing the dual-path is aggregating encoded channel information of each path, and full use of channel information is critical to the final performance. Besides, the output feature map of the last encoding stage contains rich semantic information. The importance of the channels in the dual-path should be distinguished carefully when fusing them.
Instead of using element-wise summation or concatenating straightly to fuse the dual-path, we take respective specialities of the residual-path and the densely connected path into consideration. While striving to improve the expression of reused features in the residual-path, we also raise the importance of new-finding features and suppress redundant features in the densely connected path. To achieve this goal, we take the channel-wise attention mechanism to model the channel relationship before fusing the dual-path. The multi-path attention mechanism has proved great success in SKNet, which generates attention maps across two network branches. Inspired by SKNet, we design a novel attention module, attention fusion module (AFM) as illustrated in Figure 5, to provide guidance by channel-wise attention for fusing the dual-path.

AFM is fed by the feature maps of the dual-path as the input and made of two parts—model, fuse and two branches—residual branch and densely connected branch. For the model part, it is mainly to generate the channel-wise attention of the dual-path. Firstly, we use an element-wise summation to connect the dual-path for computation saving. Then we aggregate global contextual information via global average pooling, and the channel relationship is generated by sending global information into two fully connected (FC) layers. For more desirable efficiency, the first FC layer reduces the channel dimension to $1/r$ ($r = 16$ is the setting in our experiment), and the second FC layer raises the channel dimension back to its original size. At last, a softmax operator is applied to generate the channel-wise attention maps after FC layers.

As for the fuse part, it uses the channel-wise attention maps generated by the model part to weight the original feature maps of the two branches separately via element-wise product. The final fused feature map is produced by merging the weighted feature maps of the two branches. Details have been shown in Appendix C.

5 | UPSAMPLE DUAL-ATTENTION MODULE

The size of the feature map continuously becomes smaller at the encoding stage, yielding more abstract pixel-level spatial information. For keypoint-based detectors, rough pixel-level information is not conducive to accurately predicting keypoints. Thus, it is essential to recover the feature map to a certain resolution before making predictions. Some encoder-decoder networks \cite{39, 40} use transposed convolution layers to gradually recover the resolution while others fuse different scales of feature maps to recover spatial information which are proven to be effective. Ref. \cite{41–43} have shown that the attention mechanism can effectively enhance expressions of valuable information and suppress worthless information.

We attempt to extract attention from multi-scale features through a lightweight attention module, and then apply attention to the up-sampled features. The up-sampled features generated in this way have richer pixel-level spatial information and semantic information, which can narrow the search space of keypoints and facilitate accurate prediction. The channel attention module in CABM is used to generate the channel-wise attention map: it first performs global average pooling and global max pooling on feature maps to aggregate channel information; then the two results of the previous operation are forwarded to a multi-layer perceptron (MLP) to compute channel-wise attention map.

As for spatial attention, we present a novel method to generate spatial attention maps. It consists of two branches: (1) Perform global average pooling and global max pooling then concatenate the results and use a $1 \times 1$ convolution to aggregate spatial information; (2) Parallel convolution is performed by different kernels on low-level features, which are concatenated to obtain multi-scale spatial information. We use an element-wise summation to fuse the spatial information of the two branches and perform a sigmoid operation to get the final spatial attention map. The detailed generation of the spatial attention is given in Appendix D.

Combining the two methods above, our proposed upsample dual-attention module (UDM), as shown in Figure 6, aggregates multi-scale spatial attention and channel-wise attention into upsampled features. It generates channel-wise attention from the high-level feature map with stronger semantic information, and spatial attention from the low-level feature map with rich spatial information. Unlike most attention methods that apply the attention map directly on the feature map, we adopt the strategy of applying channel-wise attention first and then spatial attention, which is simple and effective, and high-level channel-wise attention can provide the guidance information for the low-level spatial attention. Both elementwise summation and concatenation operation are used to apply the attention map on the upsampled feature map as shown in Figure 7.

5.1 | Dual-path attention network

Based on the encoding-decoding framework, we present dual-path attention network (DPANet) for object detection, shown in Figure 2. The network structure is made of three parts: the encoding stage, the decoding stage, and prediction.

The backbone network has a great influence on the final performance of the object detector, and most of the detection time is spent on it. Besides, more and more detectors are transferred to mobile terminals with less hardware capability, which has a high limitation on model size and computational complexity.
Our proposed wide dual-path network series have powerful performance on the feature extraction with smaller size and less computational complexity. So we use our proposed wide dual-path backbone network as the backbone network for feature extraction at the encoding stage, which consists of a stem block and four stages of stacking multiple wide dual-path blocks. We perform downsampling by the first block of each stage, and the input will be downsampled to $32 \times$ in the encoding stage. The numbers of stacked blocks are $\{3, 4, 6, 3\}$ at each stage, and the total weighted layers are 52.

For the keypoint-based object detection, pixel-level information is critical for the prediction. Feature information has been encoded by the backbone network at the encoding stage, and need to be decoded at the decoding stage. Our proposed UDM can recover precise pixel-level information via fusing the multi-scale attention map, but the dual-path is not available in the UDM. The AFM is introduced to solve this problem. Therefore, the decoding stage contains one AFM and three UDMs. We fist use the AFM to generate channel-wise attention and fuse of the dual-path from the encoding stage. Then, three UDMs are stacked to upsample the feature map and generate the multi-scale attention map. More specifically, the low-level feature of the first UDM’s comes from the output of stage 3, the second one comes from the output of stage 2, and the third one from stage 1. Finally, the decoding stage will recover the resolution to a quarter of the input.

In the prediction part, three convolution predictors are assigned to predict results: keypoint predictor for predicting keypoint heatmaps, width and height predictor for predicting...
6 | EXPERIMENTAL RESULTS AND ANALYSIS

We evaluate our wide dual-path backbone network on the CIFAR-10 classification dataset and report the top-1 errors. Then we test our dual-path attention network on PASCAL VOC and MS COCO [44] object detection dataset, and report mean average precision (mAP). All the results are the averages of 3 runs by default. Experiments are conducted under the Windows10 system and Mxnet frameworks using the i9 9900k CPU, and the RTX 2080ti GPU.

6.1 | CIFAR-10 classification

CIFAR-10 dataset consists of 10 classes of nature images and the train and test sets contain 50k images and 10k images respectively with 32 × 32 pixels each. Random flip and crop are used as data augmentation to avoid over-fitting. We adopt the cosine learning rate decay strategy [38] to control training process. And we use SGD to optimize the loss with a weight decay of 0.0005 and a momentum of 0.9.

Table 1 shows the results of the CIFAR-10 validation set with different backbone networks. Wide_DPN-50_2.0x achieves the best accuracy among compared backbone networks, with a 4.98% top-1 error. In the meantime, our Wide_DPN-50_1.5x outperforms DenseNet-121, while Wide_DPN-50_1.0x with fewer params outperforms ResNet-50. Our best accurate backbone network, Wide_DPN-50_2.0x, has 12% fewer params comparing with 92 layers DPN. And all of our Wide_DPN-50 series outperform ResNet-50.

We also evaluate the differences in three modifications on the basic dual-path block, the results are shown in Table 2. From the results, we can see these changes did not bring too many changes of parameters and complexity. On the contrary, they can bring some extend of improvement: Replacing the 7 × 7 convolution with three 3 × 3 ones reduces about 0.6% error while replacing normal 3 × 3 convolution with 3 × 3 depthwise convolution brings another 0.3% improvement; adding an avgpool can bring about 0.5% improvement. The total improvement of an error on the basic wide dual-path network is more than 1%.

6.2 | PASCAL VOC object detection

PASCAL VOC is one of the most famous object detection datasets. Our models are trained on the VOC 2007 and 2012 train val sets (contain 16,551 images), and tested on the VOC 2007 test set (contain 4952 images). For a fair comparison, we use the same objective, optimizer, and data augmentation of CenterNet to train our DPANet. Noting that we did not pretrain our backbone network on ImageNet and directly train our network on VOC dataset from scratch. To get considerable results in comparison with other pre-trained models, we increased the number of training epochs. All layers are randomly initialized by Xavier [45] method. Learning rate decays at every 60 epochs and the initial learning rate is 0.003. Other models and codes are downloaded from Mxnet model-zoo to test FPS.

Table 3 and Figure 8 show our experimental results and test results on VOC 2007 test set. Relationship between the performance and the model size is shown in Figure 9, compared with state-of-the-art models. Our DPANet_2.0x achieves the best accuracy (79.9% mAP at 11 FPS). However, considering with DPANet_1.0 x and DPANet_1.5x, this improvement cost too much time. Our DPANet_1.0x achieves an accuracy-efficiency trade-off result (78.3% mAP at 41 FPS). Compared with CenterNet with Resnet-50 backbone, our three versions of DPANet outperform it by 1.2%, 2.3%, and 2.8% respectively in mAP.

Compared the FPS, mAP, and model size of three versions of DPANet, we find that although DPANet_2.0x can achieve a better mAP, its speed is too slow (11 FPS). Compared with DPANet_1.5x, it only improves mAP by 0.5%, but the speed is 53% slower than DPANet_1.5x, and its model size is 36% larger than DPANet_1.5x. Therefore, simply widening the network by increasing the number of channels can get better accuracy but cannot achieve an accuracy-efficiency tradeoff. We also evaluate our model using DPN-92 as baseline, which achieves 79.1% mAP at 26 FPS. The results suggest that our 50 layers of wide dual-path backbone 1.5 x can achieve the performance of 92 layers of DPN with faster speed and smaller model size.

For those keypoint-based detectors with the Hourglass network as backbone network, although their performance is relatively good (more than 81% mAP), the speed is very slow.
| Method          | Backbone     | Input size     | FPS | Model size  | mAP (%) |
|-----------------|--------------|----------------|-----|-------------|---------|
| Two-stage:      |              |                |     |             |         |
| Faster R-CNN [3]| Resnet-50    | 600 \times 1000|     | 130 MB      | 78.3    |
| Faster R-CNN [3]| Resnet-101   | 600 \times 1000|     | 203 MB      | 79.5    |
| R-FCN [46]      | Resnet-50    | 600 \times 1000|     | 135 MB      | 77.2    |
| R-FCN [46]      | Resnet-101   | 600 \times 1000|     | 208 MB      | 78.4    |
| One stage:      |              |                |     |             |         |
| YOLOv3 [47]     | Darknet-53   | 320 \times 320  | 49  | 235 MB      | 79.3    |
| YOLOv4 [48]     | CSPDarknet-53| 512 \times 512  | 53  | 289 MB      | 82.4    |
| SSD [1]         | VGG-16       | 300 \times 300  | 46  | 101 MB      | 77.6    |
| SSD [1]         | VGG-16       | 512 \times 512  | 31  | 140 MB      | 78.4    |
| CornerNet [8]   | Hourglass-104| 511 \times 511  | 3   | —           | 81.2    |
| CenterNet-Triplets [9] | Hourglass-104 | 511 \times 511 | 2   | —           | 83.6    |
| ExtremeNet [10] | Hourglass-104| 511 \times 511  | 2   | —           | 81.1    |
| CenterNet [8]   | Resnet-50    | 384 \times 384  | 55  | 117 MB      | 77.1    |
| CenterNet [8]   | Resnet-101   | 384 \times 384  | 32  | 189 MB      | 78.9    |
| Ours:           |              |                |     |             |         |
| DPANet          | DPN-92       | 384 \times 384  | 26  | 165 MB      | 79.1    |
| DPANet_1.0x     | Wide_DPN-50_1.0x | 384 \times 384 | 41  | 76 MB      | 78.3    |
| DPANet_1.5x     | Wide_DPN-50_1.5x | 384 \times 384 | 29  | 128 MB     | 79.4    |
| DPANet_2.0x     | Wide_DPN-50_2.0x | 384 \times 384 | 11  | 175 MB     | 79.9    |

**FIGURE 8** Test results of DPANet on VOC object detection dataset
The relationship between performance and model size of DPANet and state-of-arts (lower than 5 FPS). Two main reasons are responsible for their slow speed: (1) Backbone network. The Hourglass network is a powerful backbone network that can extract accurate pixel information to predict keypoints. However, due to its complex network structure, it contains more parameters and needs more time. (2) Post-processing methods. For ExtremeNet and CenterNet-Triplets, they are essentially the upgraded methods of CornerNet. After predicting multiple keypoints of objects, they need to apply additional processes to assemble the keypoints of the same object. So it takes extra time to deal with the predictions of the keypoints.

As for our proposed model, we use our wide dual-path network as backbone, which is configured to obtain different information extraction capabilities through different network width settings. We also merge the multi-scale features by using our proposed attention module AFM and UDM to provide more accurate pixel information. In addition, after predicting the centre point of an object instance, our model directly predicts the height and width of the object. It does not need to spend more time to assemble the keypoint information of the object, so our model is faster.

6.3 Ablation study on VOC object detection

6.3.1 Effectiveness and generality of UDM

We compare the effects of different attention applying methods: element-wise summation, concatenation, element-wise summation then concatenation. Table 4 shows that using both element-wise summation and concatenation attention map on the upsampled feature map achieves better performance while loses negligible speed. This is similar to the idea of the dual-path network; using both element-wise summation and concatenation to apply attention map on upsampled feature map, which will double the number of output channels but it can be beneficial to discover more feature details. Therefore, the performance is better but speed is relatively slower.

| Attention applying method      | mAP%  | FPS  | Model size |
|--------------------------------|-------|------|------------|
| DPANet_1.0x                    |       |      |            |
| +Summation                     | 78.22 | 41.7 | 76 MB      |
| +Concatenation                 | 78.15 | 42.2 | 76 MB      |
| +Summation and Concatenation   | 78.31 | 41.4 | 76 MB      |

6.3.2 Channel increment of Wide_DPN-50

It can be seen from Table 3: although DPANet_1.0x has achieved better accuracy and has a smaller model size, its speed is slower than CenterNet with ResNet-50 backbone. In this part, we try to explore the factors that affect the speed of DPANet. From Table 8, it can be seen that except for DPANet_2.0x, the

| TABLE 4 | The results of different attention applying methods on accuracy and speed |
|---------|---------------------|
| Attention applying method | mAP%  | FPS  | Model size |
| DPANet_1.0x                    |       |      |            |
| +Summation                     | 78.22 | 41.7 | 76 MB      |
| +Concatenation                 | 78.15 | 42.2 | 76 MB      |
| +Summation and Concatenation   | 78.31 | 41.4 | 76 MB      |

6.3.1 Effectiveness and generality of UDM

We study the contribution of AFM and UDM to the performance of DPANet. Table 5 shows that using AFM and UDM can improve mAP by 0.5% and 1.7% respectively. If we use the two modules together, the improvement will be better than using the two modules separately, and the speed will be faster. The reason is that AFM can effectively gather channel information of the dual-path and reduce the number of channels, which is better than using $1 \times 1$ convolution to change channels.

To test generality of UDM, we apply it in CenterNet. Table 6 shows the result of CenterNet with UDM: it improves 1.4% mAP and loses 19 FPS on speed. Since the output channels of the last layer in Resnet-50 is very large (2048), the speed loses a lot and the model size also increases. If we could use a lightweight module like AFM to aggregate global channel information and reduce the number of channels before UDM, the speed loss can be lowered and the performance can be better. We believe that UDM can be used in any decoder-encoder structure model, and improve their performance to some extent without losing too much speed.

| TABLE 5 | Ablation study on DPANet. Noting that we replacing UDM with normal transposed convolution layers on basic DPANet_1.0x |
|---------|-------------------------------------------------|
| DPANet  | mAP (%) | FPS  | Model size |
| DPANet_1.0x (basic) | 76.2 | 37   | 68 MB      |
| +AFM    | 76.7    | 37   | 65 MB      |
| +UDM    | 77.9    | 28   | 80 MB      |
| +AFM, UDM | 78.3 | 41   | 76 MB      |

6.3.2 Channel increment of Wide_DPN-50

It can be seen from Table 3: although DPANet_1.0x has achieved better accuracy and has a smaller model size, its speed is slower than CenterNet with ResNet-50 backbone. In this part, we try to explore the factors that affect the speed of DPANet. From Table 8, it can be seen that except for DPANet_2.0x, the

| TABLE 6 | CenterNet with Resnet-50 backbone achieves better results when normal upsample layers replaced with UDM |
|---------|-------------------------------------------------|
| CenterNet | mAP (%) | FPS  | Model size |
| Resnet-50 | 77.1    | 55   | 117 MB     |
| +UDM      | 78.5    | 36   | 197 MB     |
TABLE 7  Keypoint-based object detection comparison on the MS COCO validation dataset

| Method                  | Backbone   | Input size | FPS | AP  | AP50 | AP75 | AP5 | AP25 | AP49 | AP49 |
|-------------------------|------------|------------|-----|-----|------|------|-----|------|------|------|
| CornerNet [8]           | Hourglass-104 | 511 × 511  | 3.6 | 40.5 | 58.2 | 45.1 | 20.9 | 44.8 | 56.7 |
| ExtremeNet [10]         | Hourglass-104 | 511 × 511  | 2.7 | 40.2 | 55.3 | 43.2 | 20.4 | 43.2 | 53.1 |
| CenterNet-Triplets [9]  | Hourglass-104 | 511 × 511  | 2.5 | 44.7 | 61.9 | 47.5 | 25.2 | 47.0 | 57.1 |
| CenterNet [8]           | Resnet-101  | 512 × 512   | 21.3| 37.9 | 56.8 | 40.4 | 17.9 | 41.4 | 49.2 |
| DPANet                  | DPN-92      | 512 × 512   | 19.4| 38.5 | 57.9 | 42.1 | 18.8 | 42.9 | 51.1 |
| DPANet_1.0              | Wide_DPN-50_1.0X | 512 × 512  | 11.5| 2.8G | 167.4 MB | 37 |
| DPANet_1.5              | Wide_DPN-50_1.5X | 512 × 512  | 22.6| 5.5G | 228.6 MB | 25 |
| DPANet_2.0              | Wide_DPN-50_2.0X | 512 × 512  | 31.8| 8.3G | 283.1 MB | 9 |

TABLE 8  Comparison of parameters, FLOPs, and memory reads and writes between Resnet-50 and Wide_DPN-50 series. Noting the input size is 384 × 384

| Method                  | #params | FLOPs | Memory | FPS |
|-------------------------|---------|-------|--------|-----|
| CenterNet:              | 25.5 M  | 12.1G | 322.3 MB | 55 |
| Resnet-50               |         |       |        |     |
| DPANet(basic):          |         |       |        |     |
| Wide_DPN-50_1.0X        | 11.5 M  | 2.8G  | 167.4 MB | 37 |
| Wide_DPN-50_1.5X        | 22.6 M  | 5.5G  | 228.6 MB | 25 |
| Wide_DPN-50_2.0X        | 31.8 M  | 8.3G  | 283.1 MB | 9 |

TABLE 9  Results of different channel increments in the densely connected path

| DPANet_1.0X | #params | FLOPs | Memory | FPS | mAP (%) |
|-------------|---------|-------|--------|-----|---------|
| A: {16,32,32,64} | 9.6 M  | 2.5G  | 156 MB | 44.6 | 77.6 |
| B: {32,64,64,128} | 11.5 M | 2.8G  | 167.4 MB | 41.5 | 78.3 |
| C: {64,128,128,256} | 15.6 M | 3.6G  | 190.3 MB | 33.2 | 78.5 |

6.4 | MS COCO object detection

We also evaluate our method on the MS COCO dataset (2017 train-val), which contains 80 categories and more than 1.5 million object instances, and make comparison with keypoint-based methods. Our models are trained on the 2017 train dataset, and tested on the 2017 validation dataset. Other keypoint-based models are downloaded pretrained models from original publication and tested on the validation-set with our local machine. We report average precision (APs) at different IOU thresholds, and APs for different object sizes.

Table 7 presents the comparison with the state-of-the-art keypoint-based detectors, and Figure 10 shows detection results on MS COCO 2017 validation dataset. CenterNet-Triplets with Hourglass-104 backbone network outperforms all other keypoint-based detectors with 44.7% AP and 2.9 FPS. However, its speed is nearly 15 times slower than our method and 8 times slower than CenterNet with Resnet-101 backbone network.

As for our DPANet series, we fix the input resolution to 512 × 512, and our DPANet_1.5X outperforms the baseline with 0.6% AP and 2.7 FPS improvement. Given that the baseline uses a 92 layers DPN network, our proposed wide dual-path backbone network achieves better accuracy-efficiency tradeoff. Our DPANet_2.0X achieves the best performance with 39.3% AP at 11.4 FPS, and DPANet_1.0X achieves the fast speed with 36.8% AP at 33.5 FPS.

7 | CONCLUSION

In this paper, we focus on improving the detection of the objects’ centre point. We mainly improve it from two aspects: proposing a Wide_DPN-50 and introducing two attention modules (AFM and UDM). Our proposed DPANet series, which is a combination of our proposed Wide_DPN-50, AFM, and UDM, have achieved better performance in PASCAL VOC object detection datasets. We have also proved that our proposed attention module, UDM, can be also used in other models.

In the future, we will continue to improve the performance of the keypoint-based detector from two aspects: (1) Adopt a lightweight network structure to build a more powerful backbone network for accelerating inference speed. (2) Combine
attention mechanism with multi-keypoint methods to improve the accuracy of object detection.

ACKNOWLEDGMENTS
The authors wish to thank the associate editors and anonymous reviewers for their valuable comments and suggestions on this paper.

ORCID
Run Zhang https://orcid.org/0000-0003-4199-0760

REFERENCES
1. W. Liu, et al.: SSD: Single shot multiBox detector. In: Proceedings of the European Conference on Computer Vision. Amsterdam, pp. 21–37 (2016)
2. T. Lin, et al.: Focal loss for dense object detection. IEEE Trans. Pattern Anal. Mach. Intell. 42(2), 318–327 (2020)
3. S. Ren, et al.: Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39(6), 1137–1149 (2017)
4. J. Redmon, A. Farhadi: Yolo9000: Better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, pp. 6517–6525 (2017)
5. B. Yu, D. Tao: Anchor cascade for efficient face detection. IEEE Trans. Trans. Image Process. 28(5), 2490–2501 (2018)
6. T. Kong, et al.: Foveabox: Beyond anchor-based object detection. IEEE Trans. Image Process. 29, 7389–7398 (2020)
7. Y. Li, et al.: Pose anchor: A single-stage hand keypoint detection network. IEEE Trans. Circuits Syst. Video Technol. 30(7), 2104–2113 (2020)
8. H. Law, J. Deng: Cornernet: Detecting objects as paired keypoints. Int. J. Comput. Vis. 128(3), 642–656 (2019)
9. K. Duan, et al.: Centernet: Keypoint triplets for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. Seoul, pp. 6568–6577 (2019)
10. X. Zhou, J. Zhuo, P. Krahenbuhl: Bottom-up object detection by grouping extreme and center points. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. California, pp. 850–859 (2019)
11. X. Zhou, D. Wang, P. Krahenbuhl: Objects as points. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. California, arXiv:1904.07850 (2019)
12. J. Long, E. Shelhamer, T. Darrell: SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39(12), 2481–2495 (2017)
13. X. Li, et al.: Multi-task structure-aware context modeling for robust keypoint-based object tracking. IEEE Trans. Pattern Anal. Mach. Intell. 41(4), 915–927 (2019)
14. L. Liqiang, C. Jianzhong: End-to-end learning interpolation for object tracking in low frame-rate video. IET Image Process. 14(6), 1066–1072 (2020)
15. K. Jun, et al.: Collaborative model tracking with robust occlusion handling. IET Image Process. 14(9), 1701–1709 (2020)
16. P. Karl, R. Leonardo, R. Eduardo: Real-time pose detection and tracking of hundreds of objects. IEEE Trans. Circuits Syst. Video Technol. 26(12), 2200–2214 (2015)
17. T. Lin, et al.: Pose detection in complex classroom environment based on improved Faster R-CNN. IET Image Process. 13(3), 451–457 (2019)
18. C. Zhichao, L. Yueshu, R. Fujii: Homography-based traffic sign localization and pose estimation from image sequence. IET Image Process. 13(14), 2829–2839 (2019)
19. E. Mehryar, E. Adrian: Nasal patches and curves for expression robust 3d face recognition. IEEE Trans. Pattern Anal. Mach. Intell 23(4), 729–734 (2013)
20. B. Gaurang, et al.: Deep3DScan: Deep residual network and morphological descriptor based framework for lung cancer classification and 3D segmentation. IET Image Process. 14(7), 1240–1247 (2020)
21. S. S. Prakash, S. Ulli, A. Samit: 3D Features for human action recognition with semi-supervised learning. IET Image Process. 13(6), 983–990 (2019)
22. A.Z. Karen Simonyan: A very deep convolutional networks for largescale image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Columbus, pp. 2333–9721 (2014)
23. K. He, et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Seattle, pp. 770–778 (2016)
24. G. Huang, et al.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, pp. 2261–2269 (2017)
25. Y. Chen, et al.: Dual path networks. In: Proceedings of Advances in Neural Information Processing Systems. California, pp. 4467–4475 (2017)

26. X. Zhang, et al.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, pp. 6848–6856 (2018)

27. A. P. Sundara Sobitha Raj, S. K. Vajravelu: Ddla: Dual deep learning architecture for classification of plant species. IET Image Process. 13(12), 2176–2182 (2020)

28. H. Xun, et al.: Mobilenet-ssd microscope using adaptive error correction algorithm: Real-time detection of license plates on mobile devices. IET Intell. Transp. Syst. 14(2), 110–118 (2020)

29. F. Chollet: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, pp. 1800–1807 (2017)

30. J. Hu, et al.: Squeeze-and-excitation networks. IEEE Trans. Pattern Anal. Mach. Intell. 48(8), 2011–2023 (2019)

31. Cao, Y., et al.: GCNet: Non-Local Networks Meet Squeeze-excitation Networks and Beyond. In: Proceedings of the IEEE International Conference on Computer Vision. Seoul, pp. 1971–1980 (2019)

32. X. Wang, et al.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, pp. 7794–7803 (2018)

33. S. Zhou, et al.: Highresolution encoder–decoder networks for low-contrast medical image segmentation. IEEE Trans. Image Process. 29, 461–475 (2019)

34. T. He, et al.: Bag of tricks for image classification with convolutional neural networks. In: Proceedings of Advances in Neural Information Processing Systems. Barcelona, pp. 2818–2826 (2016)

35. Y. Zhu, et al.: Attention couplenet: Fully convolutional attention coupling network for object detection. IEEE Trans. Image Process. 28(1), 113–126 (2019)

36. S. Liu, D. Huang, Y. Wang: Pay attention to them: Deep reinforcement learning-based cascade object detection. IEEE Trans. Neural Networks Learn. Syst. 31(7), 2544–2556 (2020)

37. J. Redmon, A. Farhadi: Yolov3: An incremental improvement. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, arXiv:1804.02767 (2018)

38. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.: YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934

How to cite this article: Qu Z, Zhang R, Bao K-h. A keypoint-based object detection method with wide dual-path backbone network and attention modules. IET Image Process. 2021;15:1800–1813. https://doi.org/10.1049/ipr2.12152
APPENDIX A: Backbone architecture

TABLE 10  Architecture comparison of ResNet and our proposed Wide_DPN-50 series. The symbol (+k) denotes the increment on the densely connected path.

| Stage  | Output size | ResNet-50 | Wide DPN-50 1.0x | Wide DPN-50 1.5x | Wide DPN-50 2.0x |
|--------|-------------|-----------|------------------|------------------|------------------|
| Stem   | 256 × 256   | 7 × 7, 64, stride 2 | 3 × 3, 32, stride 1 | 3 × 3, 48, stride 1 | 3 × 3, 64, stride 1 |
| 128 × 128 | 3 × 3 maxpool, stride 2 | 1 × 1, 64 | 3 × 3, 32, 128 | 1 × 1, 128, 192 | 1 × 1, 265, 32 |
| Stage 2 | 64 × 64 | 3 × 3, 128 | 1 × 1, 128 | 3 × 3, 256 | 1 × 1, 320 |
| Stage 3 | 32 × 32 | 1 × 1, 256 | 3 × 3, 384 | 3 × 3, 512 | 1 × 1, 640 |
| Stage 4 | 16 × 16 | 1 × 1, 1024 | 1 × 1, 512 | 1 × 1, 768 | 1 × 1, 1280 |
| 1 × 1 | Average pool, 10-d fc, softmax |

APPENDIX B: Algorithm of dual-path block forward propagation

Algorithm 1. The forward propagation of wide dual-path block

Input: residual, dense, chanIn;
Output: residual, dense;
(1) inData = Concat(residual, dense);
(2) if downsample: true then
(3) \{inData = Avgpool(inData);
(4) inData = Conv1 × 1(inData);
(5) dataResidual = Split(inData, residual.channel);
(6) dataDense = Split(inData, dense.channel);\}
(7) else if downsample: false then
(8) \{dataResidual = residual;
(9) dataDense = dense;\}
(10) end if
(11) conv = Conv1 × 1(inData);
(12) conv = DWConv3 × 3(conv);
(13) conv = Conv1 × 1(conv);
(14) convResidual = Split(conv, residual.channel);
(15) convDense = Split(conv, dense.channel + chanIn);
(16) residual = convResidual + residual;
(17) dense = Concat(convDense, dense);

APPENDIX C: Algorithm of AFM

Algorithm 2. Attention guided fusion of the dual-path

Input: residual, dense;
Output: mergedPath;
(1) \texttt{att} = residual + dense;
(2) \texttt{att} = GlobalAvgPool(\texttt{att});
(3) \texttt{att} = FC(\texttt{att});
(4) \texttt{att} = Softmax(\texttt{att});
(5) \texttt{attResidual} = Split(\texttt{att}, residual.channel);
(6) \texttt{attResidual} = \texttt{attResidual} × residual;
(7) \texttt{attDense} = Split(\texttt{att}, dense.channel);
(8) \texttt{attDense} = \texttt{attDense} × dense;
(9) \texttt{mergedPath} = \texttt{attResidual} + \texttt{attDense};
APPENDIX D: Algorithm of spatial attention generation

Algorithm 3. Generation of the spatial attention

|   | Input: lowFeature; |
|---|-------------------|
|   | Output: spatialAtt; |
| 1 | \( \text{avg} = \text{Avgpool}(\text{lowFeature}) \); |
| 2 | \( \text{max} = \text{Maxpool}(\text{lowFeature}) \); |
| 3 | \( \text{att} = \text{Concat}(\text{avg}, \text{max}) \); |
| 4 | \( \text{att} = \text{Conv1} \times 1(\text{att}) \); |
| 5 | \( \text{conv1} = \text{Conv1} \times 1(\text{lowFeature}) \); |
| 6 | \( \text{conv3} = \text{Conv3} \times 3(\text{lowFeature}) \); |
| 7 | \( \text{conv5} = \text{Conv5} \times 5(\text{lowFeature}) \); |
| 8 | \( \text{conv7} = \text{Conv7} \times 7(\text{lowFeature}) \); |
| 9 | \( \text{spatial} = \text{Concat}(\text{conv1}, \text{conv3}, \text{conv5}, \text{conv7}) \); |
|10| \( \text{spatialAtt} = \text{Sigmoid}(\text{att} + \text{spatial}) \); |