A new multi-scale backbone network for object detection based on asymmetric convolutions

Xianghua Ma and Zhenkun Yang
School of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shanghai, China

Abstract
Real-time object detection on mobile platforms is a crucial but challenging computer vision task. However, it is widely recognized that although the lightweight object detectors have a high detection speed, the detection accuracy is relatively low. In order to improve detecting accuracy, it is beneficial to extract complete multi-scale image features in visual cognitive tasks. Asymmetric convolutions have a useful quality, that is, they have different aspect ratios, which can be used to exact image features of objects, especially objects with multi-scale characteristics. In this paper, we exploit three different asymmetric convolutions in parallel and propose a new multi-scale asymmetric convolution unit, namely MAC block to enhance multi-scale representation ability of CNNs. In addition, MAC block can adaptively merge the features with different scales by allocating learnable weighted parameters to three different asymmetric convolution branches. The proposed MAC blocks can be inserted into the state-of-the-art backbone such as ResNet-50 to form a new multi-scale backbone network of object detectors. To evaluate the performance of MAC block, we conduct experiments on CIFAR-100, PASCAL VOC 2007, PASCAL VOC 2012 and MS COCO 2014 datasets. Experimental results show that the detection precision can be greatly improved while a fast detection speed is guaranteed as well.

Keywords
Object detection, asymmetric convolutions, multi-scale representation, deep learning, backbone network

Corresponding author:
Xianghua Ma, School of Electrical and Electronic Engineering, Shanghai Institute of Technology, 100 Haiquan Road, Shanghai 201418, China.
Email: xhuam@sit.edu.cn

Creative Commons Non Commercial CC BY-NC: This article is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 License (https://creativecommons.org/licenses/nc/4.0/) which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
Introduction

Object detection is a hot topic of computer vision and digital image processing, which is widely used in robot navigation, intelligent video monitoring, industrial detection and many other fields. In 2019, Wu et al.\textsuperscript{1} proposed a novel object detection framework using spatial-frequency channel feature. In Wu et al.,\textsuperscript{2} a novel and efficient framework (FRIFB) was presented for geospatial object detection. Experiments demonstrated the superiority and effectiveness of the FRIFB compared to previous state-of-the-art methods.

In recent years, Convolutional Neural Networks (CNNs) are increasingly applied to a variety of vision problems, such as object detection\textsuperscript{3–5} and image classification.\textsuperscript{6,7} Thanks to the huge development of deep neural networks, a large number of object detection methods have emerged. CNN-based object detectors are commonly classified into two-stage detectors and one-stage detectors. The two-stage detector such as Fast R-CNN\textsuperscript{8} and R-FCN\textsuperscript{9} has achieved great success in recent years. Fast R-CNN\textsuperscript{8} employed several innovations to improve training and testing speed while also increasing detection accuracy. R-FCN\textsuperscript{9} proposed position-sensitive score maps to address a dilemma between translation-invariance in image classification and translation-variance in object detection. Compared with the two-stage approach, the one-stage approach, such as YOLO\textsuperscript{10}, SSD\textsuperscript{11} and YOLOv3\textsuperscript{12} have a better real-time performance while maintaining better detection accuracy. In Redmon et al.,\textsuperscript{10} training and detection were placed in a network, and object detection was solved as a regression problem. SSD\textsuperscript{11} predicted objects at multiple layers of the feature hierarchy without combining features or scores. YOLOv3\textsuperscript{12} improved the detection ability of small objects and reduced the overfitting of the network. In 2020, Yang et al.\textsuperscript{13} proposed a new lightweight one-stage generic object detector to improve the detection accuracy of object detectors while maintaining a high detection speed.

Previous research Asymmetric convolutions have an obvious advantage on reducing the computational cost of CNNs. Kim et al.\textsuperscript{14} proposed a macro unit, which employed asymmetric convolutions to reduce heavy computations and the number of model parameters. Jiang and Jin\textsuperscript{15} designed a multiple asymmetric convolutional layer of different scales to extract nonlinear features of multi-granular n-gram phrases. Denton et al.\textsuperscript{16} exploited the redundancy present within the convolutional filters to derive approximations based on Singular Value Decomposition manner, thereby significantly reducing the required computation. Jaderberg et al.\textsuperscript{17} found that the set of horizontal and vertical filters can be learnt by explicitly minimizing the reconstruction error of the original filters. Jin et al.\textsuperscript{18} applied structural constraints in 1D separated filter learning to achieve significant reduction in parameters. Szegedy et al.\textsuperscript{19} have shown that a $7 \times 7$ convolution can be replaced by a $1 \times 7$ convolution. In practice, the authors found that using this factorization did not work well on earlier layers, but it gives good results with a grid size of 12–20. Ding et al.\textsuperscript{20} proposed a new deep neural network (ACNet), which employed asymmetric convolutions to explicitly enhance the representational power of a standard square-kernel layer.
According to the above analysis, these researches on object detection methods based on asymmetric convolutions can reduce the computation cost significantly and improve the detection speed by network compression and acceleration. However, the amount of network parameters is reduced during the compression and acceleration process of the convolution network; therefore, the detection precision of the object detectors is sacrificed. It is a big challenge for object detection to achieve a fast detection speed while guaranteeing a high detection precision.

Actually, image features of objects usually present at multi-scale in natural scenes. The vision pattern of objects may appear with different sizes and aspect ratios in a single image. In order to improve detecting precision, it is beneficial to extract complete multi-scale image features in visual cognitive tasks. Asymmetric convolutions have another useful quality, that is, they have different aspect ratios, which can be used to exact image features of objects, especially objects with multi-scale characteristics. In this paper, we propose a new multi-scale building block for object detectors, namely “multi-scale asymmetric convolution” (MAC) block based on the different aspect ratios of asymmetric convolutions to enhance multi-scale representation ability of CNNs. By using the MAC block, the multi-scale problem of different sizes and aspect ratios in the image can be handled more effectively.

Our main contributions are summarized as follows:

1. We exploit three different asymmetric convolutions in parallel and propose a new multi-scale asymmetric convolution unit, namely MAC block to enhance multi-scale representation ability of CNNs.

2. The proposed MAC block can adaptively merge the features with different scales by allocating learnable weighted parameters to different asymmetric convolution branches.

The rest of this paper is organized as follows. The proposed MAC block and its mathematical formulation are described in section 2. The new multi-scale backbone network, MAC-ResNet-50, is given in section 3. Experiments and analysis are presented in section 4, and conclusions are drawn in section 5.

MAC block

MAC module

Considering the image features of objects usually appear at multiple scales in an image, we propose to construct MAC block to enhance the multi-scale representation ability of backbone network for object detectors. As shown in Figure 2, a MAC block has three parallel scales with kernel sizes of $1 \times 3$ and $3 \times 1$, $1 \times 3$, $3 \times 1$, respectively, which take the same input. The first scale: Asymmetric convolution with kernel sizes of $1 \times 3$ and $3 \times 1$, respectively; The second scale: Asymmetric convolution with a kernel size of $1 \times 3$; The third scale: Asymmetric convolution with a kernel size of $3 \times 1$. We fuse the features with three parallel scales. However, the features with three parallel scales usually contribute to the
output unequally. Hence, as shown in Figure 1, a learnable weight is added to the MAC block. Suppose $W_i$ represents the weight for three parallel scales, where $i \in \{1, 2, 3\}$.

To illustrate the MAC block more intuitively, we use a sliding window to provide the specific convolution process of the MAC block as shown in Figure 2. We only depict the sliding window at the top-left and bottom-right corners of the input feature map. The yellow part in the Figure 2 is the convolution kernel, and the rectangular frame surrounded by the red line represents the sliding window.

For the first scale of the MAC block, this scale consists of asymmetric convolution kernels of two sizes of $1 \times 3$ and $3 \times 1$, which can be equivalent to a $3 \times 3$ square convolution kernel. In this way, the network reduces the amount of parameters while obtaining the same effect of the $3 \times 3$ square convolution kernel. For the second scale of the MAC block, the aspect ratio of the $1 \times 3$ asymmetric convolution kernel is 1–3. When convolution operation is performed, $1 \times 3$ asymmetric convolution is more effective for processing objects with small aspect ratio. For the third scale of MAC block, the aspect ratio of $3 \times 1$ asymmetric convolution kernel is 3 to 1. When performing convolution operation, $3 \times 1$ asymmetric convolution is more beneficial for processing objects with large aspect ratio. Image features of different scales can be extracted by summarizing the feature maps output of the three scales. The three different asymmetric convolution scales of the MAC block make it more robust when dealing with multi-scale feature extraction of objects.
Figure 2. The convolution process of MAC block.
The first scale: scale surrounded by red dotted frame; the second scale: scale surrounded by blue dotted frame; the third scale: scale surrounded by green dotted frame. $W_1^c$, $W_2^c$ and $W_3^c$ represent the weights of the corresponding $c$-th output feature map channel in the three parallel scales, respectively.

Mathematical formulation

As shown in Figure 1, for a scale in a MAC block, $I \in \mathbb{R}^{M \times N \times D}$ denotes the input of convolutional layer. $F = [g_1, g_2, ..., g_c]$ denotes the learned set of convolution kernels, where $g_c$ refers to the parameters of the corresponding $c$-th convolution kernel. $U \in \mathbb{R}^{H \times W \times C}$ denotes the output of convolutional layer. $H$, $W$, and $C$ denote the height, width, and number of channels of the feature map, respectively. For the $c$-th filter at such a layer, the corresponding output feature map channel is

$$V^c = g_c * I = \sum_{s=1}^{D} g_c^s * Y^s.$$  \hspace{1cm} (1)

Here * denotes convolution, $g_c \in \mathbb{R}^{K \times K \times D}$, $U = [V^1, V^2, ..., V^C]$, $g_c = [g_{c1}, g_{c2}, ..., g_{cD}]$ and $I = [Y^1, Y^2, ..., Y^D]$. $g_c^s$ is a 2D spatial kernel representing a single channel of $g_c$. $V^c$ denotes the $c$-th output feature map channel of $U$. $Y^s$ denotes the $s$-th input feature map channel of $I$.

To reduce overfitting and accelerate the training process, we adopt a batch normalization operation after the convolutional layer. As a common practice, in order to enhance the representation ability, we perform a linear scaling transformation after a batch normalization layer. Compared to (1), the output of the feature map channel then becomes
\[ V^c = (g_c^*I - \mu_c) \frac{\gamma_c}{\sigma_c} + \beta_c = \left( \sum_{y=1}^{D} g_c^*Y^y - \mu_c \right) \frac{\gamma_c}{\sigma_c} + \beta_c. \]  

(2)

Where \( \mu_c \) and \( \sigma_c \) are the values of channel-wise mean and standard deviation of batch normalization, \( \gamma_c \) and \( \beta_c \) are the learned scaling factor and bias term, respectively.

Finally, we fuse the features with three parallel scales. The corresponding fused output feature map channel can be calculated as

\[ V_{fusion}^c = \sum_{i=1}^{3} W_i^c, V_i^c. \]  

(3)

Where \( V_{fusion}^c \) denotes the corresponding c-th fused output feature map channel of three parallel scales. \( W_i^c \) and \( V_i^c \) represent the corresponding weight and the corresponding c-th output feature map channel for three parallel scales, where \( i \in \{1, 2, 3\} \).

**MAC-ResNet-50**

The backbone network used to extract image features is an important part of object detectors based on CNNs. By analyzing the mathematical model of the MAC block, it illustrates the feasibility of employing asymmetric convolutions in parallel to enhance multi-scale representation ability of CNNs. In this section, we will introduce a new backbone network, called MAC-ResNet-50, by integrating the MAC block into an existing backbone network.

MAC blocks can be integrated into a state-of-the-art architecture by simply replacing every \( 3 \times 3 \) layer such as residual modules. By making this change to the residual module, we can obtain a MAC-ResNet module. Figure 3 depicts the schema of the original residual module and a MAC-ResNet module.

The new backbone network, called MAC-ResNet-50, is formed by stacking a set of repeated MAC-ResNet blocks, which are termed “MAC-ResNet units”. For a concrete example of MAC-ResNet-50 backbone architecture is presented in Table 1. Each MAC-ResNet unit consists of a sequence of \( 1 \times 1 \) convolution, MAC block, and further \( 1 \times 1 \) convolution. MAC-ResNet-50 uses \{3, 4, 6, 3\} MAC-ResNet units. The first column shows the size of the output feature map at each stage of MAC-ResNet-50. The second column shows the architecture of ResNet-50 with a \( 32 \times 4d \) template. The third column shows the architecture of MAC-ResNet-50 based on ResNet-50 with a \( 32 \times 4d \) template. Filter sizes, feature dimensionalities, and strides of a MAC-ResNet module are shown inside the brackets; the number of stacked blocks for each stage is shown outside the brackets.

**Experiments**

In this section, we first conduct experiments to study the effectiveness of MAC block on the CIFAR-100 dataset. Besides, we adopt YOLOv3 as our detection
Figure 3. (a) The schema of the original residual module and (b) the schema of a MAC-ResNet module (right).

Table 1. MAC-ResNet-50 backbone network architecture based on the ResNet-50.

| Output | ResNet-50                                      | MAC-ResNet-50                                      |
|--------|-----------------------------------------------|---------------------------------------------------|
| 112 × 112 | Conv, 7 × 7, 64, stride 2               |                                                   |
| 56 × 56   | Max pool, 3 × 3, stride 2                      |                                                   |
| 56 × 56   | [1 × 1, 128]                                  | [1 × 1, 128]                                      |
|          | 3 × 3, 128 × 3                               | MAC, 128 × 3                                      |
|          | 1 × 1, 256                                   | 1 × 1, 256                                        |
| 28 × 28   | [1 × 1, 256]                                  |                                                   |
|          | 1 × 1, 256                                   |                                                   |
|          | 3 × 3, 256 × 4                               |                                                   |
|          | 1 × 1, 512                                   |                                                   |
| 14 × 14   | [1 × 1, 512]                                  |                                                   |
|          | 1 × 1, 512                                   |                                                   |
|          | 3 × 3, 512 × 6                               |                                                   |
| 7 × 7     | [1 × 1, 1024]                                 |                                                   |
|          | 1 × 1, 1024                                  |                                                   |
|          | 3 × 3, 1024 × 3                              |                                                   |
|          | 1 × 1, 2048                                  |                                                   |
| 1 × 1     | Global average, 1000-d fc, softmax            |                                                   |
method and MAC-ResNet-50 as our backbone network to form our object detector MACNet. We compare MACNet with other detection methods with the parameters of the Mean Average Precision (mAP) and the number of Frames per Second (FPS) on PASCAL VOC 2007, PASCAL VOC 2012, and COCO 2014 datasets. We implement the proposed models using the Pytorch framework. All experiments are performed on AMD 2700 CPU and GTX 1080Ti (11GB) GPU and use the Pytorch framework. We use a weight decay of 0.005 and a momentum of 0.9. The resulting model is fine-tuned using SGD.

CIFAR

In order to study the effectiveness of the MAC block, we conduct experiments on the CIFAR-100 dataset and evaluate single crop top-1 error rate. For the CIFAR-100 dataset, every model is trained 200 epochs and starts from a learning rate of 0.1 and divided by 10 every 60 epochs. The CIFAR-100 dataset consists of 60k 32 × 32 color images drawn from 100 classes, which contain 50k training images and 10k testing images. During the training process, the image is flipped horizontally, filled with four pixels on each side, and then randomly 32 × 32 cropped. We use the implementation of ResNet-18, ResNet-50, and ResNet-101, as the representatives for the residual model architecture.

We compare the single-crop top-1 error rate of each baseline and our backbones on CIFAR-100. As shown in Table 2, MAC-ResNet-50 achieves significant performance gains over the ResNet-50 with 0.72. Compared with the ResNet-18, MAC-ResNet-18 has an improvement of 0.44 in terms of top-1 error rate. Remarkably, MAC-ResNet-50 achieves 20.43 top-1 errors, although ResNet-101 is 28.59% larger in parameter. The results show that MAC blocks are consistent in improving the performance of state-of-the-art CNNs. Testing curves comparisons for different architectures are shown in Figure 4. The plot in Figure 4 shows that compared to ResNetXt-29 with 23.68 M parameters, the MAC-ResNet-50 with 30.49 M trainable parameters is able to achieve higher accuracy.

| Model            | #P   | Top-1 err. (%) |
|------------------|------|---------------|
| ResNet-18        | 11.22 M | 24.05         |
| ResNet-50        | 23.68 M | 21.55         |
| ResNet-101       | 42.70 M | 21.07         |
| MAC-ResNet-18    | 14.73 M | 23.61         |
| MAC-ResNet-50    | 30.49 M | 20.43         |
| MAC-ResNet-101   | 54.46 M | 19.84         |

#P denotes the number of parameters.
VOC 2007

For VOC 2007 task, we use VOC 2007 trainval for training and VOC 2007 test for testing. MACNet and YOLOv3 are trained 200 epochs and start from a learning rate of 0.001 and divided by 10 every 60 epochs. The corresponding mAP is obtained by calculating the AP of each category when IOU = 0.5. As can be seen from the results in Table 3 and Figure 5, Faster DPM\textsuperscript{25} improves detection accuracy compared to DPM,\textsuperscript{25} but it sacrifices 2 times real-time performance. DPM has lower detection accuracy than neural network approaches. R-CNN minus R replaces Selective Search with static bounding box proposals.\textsuperscript{26} R-CNN minus R has higher detection accuracy than Fastest DPM, but it still falls short of real-time. Faster R-CNN uses RPN to generate region proposals, which has greatly improved the detection accuracy. The detection accuracy and detection speed of Faster R-CNN reach 69.9 mAP and 7 FPS, respectively. Compared with Faster R-CNN, R-FCN has a 5.8 improvement on mAP. MACNet has a mAP of 75.7, which is 4.9 higher than YOLOv3’s 70.8. As for detection speed, MACNet has the similar detection speed with YOLOv3.

Table 3. PASCAL VOC 2007 test detection results. Training data key: VOC 2007 trainval. “impl.”: the model we implemented.

| Method              | Train | mAP  | FPS |
|---------------------|-------|------|-----|
| 30 Hz DPM\textsuperscript{25} | 2007  | 26.1 | 30  |
| Faster DPM\textsuperscript{25} | 2007  | 30.4 | 15  |
| R-CNN Minus R\textsuperscript{26} | 2007  | 53.5 | 6   |
| Faster R-CNN\textsuperscript{27} | 2007  | 69.9 | 7   |
| YOLOv3 (impl.)      | 2007  | 70.8 | 18  |
| MACNet (ours)       | 2007  | 75.7 | 16  |
For VOC 2012 task, we follow the same experimental setting of VOC 2007, but use 2007 + 2012 consisting of VOC 2007 trainval and VOC 2012 trainval for training. As can be seen from the data in Table 4 and Figure 6, the MACNet has a 6.1% improvement in detection precision over the YOLOv3. Compared to the ZF model based Faster R-CNN, the two-stage approach, Faster R-CNN VGG-16 and Faster R-CNN ResNet, has a certain improvement in precision, but has a reduced detection speed. Compared with the two-stage approach, the one-stage approach greatly improves the detection speed. YOLO achieves good detection precision, and detection speeds up to 45 FPS. Because the SSD draws on the regression idea of YOLO and the anchor boxes mechanism of Faster R-CNN, its detection speed and precision are better realized. Compared with other methods, MACNet has a detection precision of 81.7 mAP while maintaining good real-time detection.
As can be seen from the data in Table 5, on the VOC 2012 test, the detection precision of SSD is lower than R-CNN, Fast R-CNN, and Faster R-CNN. However, compared with the two-stage object detection method, YOLOv3 improves the detection precision to 73.4 mAP. Besides, compared with other methods, MACNet achieves the detection precision of 77.8 mAP. In addition, as can be seen from Table 5, MACNet has a better detection effect for objects such as bus and people with different aspect ratios and large differences in sizes such as airplane and bottle.

**COCO 2014**

For the MS COCO 2014 dataset, it contains 80 k images for training, 40 k for validation and 20 k for testing. We use the 80 k training set plus a 35 k val subset for training, minival (a 5 k val subset) for validation. MACNet and YOLOv3 are trained 140 epochs and start from a learning rate of 0.001 and divided by 10 every 40 epochs. Table 6 shows the AP of the YOLOv3 and MACNet models. The detection results of the YOLOv3 and MACNet models are shown in Table 5. The mAP is calculated by averaging the AP on the IOU [0.5, 0.95]. MACNet achieves a mAP of 30.0%, which outperforms the YOLOv3 by 2.1 points while maintaining good real-time detection. The detection effect of the MACNet model on the COCO 2014 val is shown in Figure 7.

**Conclusions**

In this paper, we propose a new building block for CNNs, namely “multi-scale asymmetric convolution” (MAC) block, by exploiting asymmetric convolutions in parallel to enhance multi-scale representation ability of CNNs. MAC block can merge the features with different scales by allocating learnable weighted parameters to three different asymmetric convolution branches. In addition, MAC blocks can be inserted into an existing backbone such as ResNet-50 to form a new backbone
| Method                  | Train   | mAP  | Aero | Bike | Bird | Boat | Bottle | Bus | Car | Cat | Chair | Cow | Table | Dog | Horse | Mbike | Person | Plant | Sheep | Soft | Train | Tv  |
|-------------------------|---------|------|------|------|------|------|--------|-----|-----|-----|-------|-----|-------|-----|-------|-------|--------|-------|-------|------|-------|-----|
| R-CNN5                  | 2007 + 2012 | 49.6 | 68.1 | 63.8 | 29.4 | 27.9 | 56.6   | 56.6| 57.0| 65.9| 26.5  | 48.7| 39.5  | 66.2| 57.3  | 65.4  | 53.2  | 26.2 | 54.5 | 38.1 | 50.6 | 51.6|
| Fast R-CNN8             | 2007 + 2012 | 68.4 | 82.3 | 78.4 | 70.8 | 52.3 | 38.7   | 77.8| 71.6| 89.3| 44.2  | 73.0| 55.0  | 87.5| 80.5  | 80.8  | 72.0  | 35.1 | 68.3 | 65.7 | 80.4 | 64.2|
| Faster R-CNN27          | 2007 + 2012 | 70.4 | 84.9 | 79.8 | 74.3 | 53.9 | 49.8   | 77.5| 75.9| 88.5| 45.6  | 77.1| 55.3  | 86.9| 81.7  | 80.9  | 79.6  | 40.1 | 72.6 | 60.9 | 81.2 | 61.5|
| YOL12                   | 2007 + 2012 | 57.9 | 77.0 | 67.2 | 57.7 | 38.3 | 22.7   | 68.3| 55.9| 81.4| 36.2  | 60.8| 48.5  | 77.2| 72.3  | 71.3  | 63.5  | 28.9 | 52.2 | 54.8 | 73.9 | 50.8|
| SSD (299 × 299)11       | 2007 + 2012 | 54.4 | 71.2 | 61.7 | 49.4 | 38.6 | 25.9   | 69.5| 52.0| 75.7| 34.1  | 52.3| 53.8  | 69.2| 59.2  | 68.0  | 63.8  | 25.8 | 44.4 | 53.3 | 70.7 | 50.1|
| SSD (443 × 443)11       | 2007 + 2012 | 63.3 | 78.6 | 68.5 | 60.2 | 46.4 | 35.6   | 74.7| 65.5| 81.8| 46.6  | 61.0| 57.6  | 75.8| 69.4  | 75.7  | 72.1  | 38.2 | 62.7 | 60.2 | 75.9 | 58.9|
| YOLOv3 (impl.)          | 2007 + 2012 | 73.4 | 86.3 | 82.0 | 74.8 | 59.2 | 51.8   | 79.8| 76.5| 90.6| 52.1  | 78.2| 58.5  | 89.3| 82.5  | 83.4  | 81.3  | 49.1 | 77.2 | 62.4 | 83.8 | 68.7|
| MACNet (ours)           | 2007 + 2012 | 77.8 | 90.1 | 84.9 | 76.3 | 64.2 | 54.4   | 85.5| 82.7| 91.6| 64.8  | 85.9| 67.7  | 91.0| 87.6  | 88.3  | 85.2  | 50.6 | 81.4 | 73.2 | 88.3 | 76.9|
Experimental results on different datasets show that the object detector using MAC-ResNet-50 has a higher detection precision while satisfying good real-time detection requirement comparing with some state-of-art detectors.

### Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

### Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by Shanghai Science and Technology Development Foundation under grant number 1951103402.

### ORCID iD

Xianghua Ma [https://orcid.org/0000-0002-6377-6629](https://orcid.org/0000-0002-6377-6629)
References

1. Wu X, Hong D, Tian J, et al. ORSIm detector: a novel object detection framework in optical remote sensing imagery using spatial-frequency channel features. IEEE Trans Geosci Remote Sens 2019; 57(7): 5146–5158.

2. Wu X, Hong D, Chanussot J, et al. Fourier-based rotation-invariant feature boosting: an efficient framework for geospatial object detection. IEEE Geosci Remote Sens Lett 2020; 17(2): 302–306.

3. Mnih V, Heess N and Graves A. Recurrent models of visual attention. In: Proceedings of the neural information processing systems (NIPS), Montréal, QC, Canada, 8–13 December 2014, pp.2204–2212. Montreal, QC, Canada: MIT.

4. He K, Zhang X and Ren S. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 2015; 37: 1904–1916.

5. Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Columbus, OH, USA, 23–28 June 2014, pp.580–587. Columbus, OH, USA: IEEE.

6. Ma X, Yang Z and Yu Z. FSRFNet: feature-selective and spatial receptive fields networks. Appl Sci 2019; 9(2): 1–15.

7. Rasti B, Hong D and Hang R. Feature extraction for hyperspectral imagery: the evolution from shallow to deep: overview and toolbox. IEEE Geosci Remote Sens Magaz 2020; 8(4): 60–88.

8. Girshick R. Fast r-cnn. In: Proceedings of the IEEE conference on computer vision, Boston, MA, USA, 8–10 June 2015, pp.1440–1448. Boston, MA, USA: IEEE.

9. Li Y, He K and Sun J. R-FCN: object detection via region-based fully convolutional networks. In: Proceedings of the advances in neural information processing systems, 2016, pp.379–387. Barcelona, Spain: MIT.

10. Redmon J, Divvala S and Girshick R. You only look once: unified, real-time object detection. In: Proceedings of IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016, pp.779–788. Las Vegas, NV, USA: IEEE.

11. Liu W, Anguelov D, Erhan D, et al. SSD: single shot multibox detector. In: Proceedings of Eur. conference computer vision, Amsterdam, Netherlands: Springer, 2016, pp.21–37. Amsterdam, Netherlands: Springer.

12. Redmon J and Farhadi A. Yolo v3: an incremental improvement. 2018, arXiv:1804.02767.

13. Yang Z, Ma X and An J. Asymmetric convolution networks based on multi-feature fusion for object detection. In: 2020 IEEE 16th international conference on automation science and engineering (CASE), 2020, pp.1355–1360. Hong Kong, China: IEEE.

14. Kim DH, Lee SH and Song BC. MUNet: macro unit-based convolutional neural network for mobile devices. In: IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW), 2018, pp.1749–1757. Salt Lake City, UT, USA: IEEE.

15. Jiang W and Jin Z. Integrating bidirectional LSTM with inception for text classification. In: Iapr Asian conference on pattern recognition, 2017, pp.870–875. Nanjing, China: IEEE.

16. Denton E, Zaremba W and Bruna J. Exploiting linear structure within convolutional networks for efficient evaluation. In: Advances in neural information processing systems, 2014, pp.1269–1277. Montreal, QC, Canada: MIT.

17. Jaderberg M, Vedaldi A and Zisserman A. Speeding up convolutional neural networks with low rank expansions. arXiv preprint arXiv:1405.3866, 2014.
18. Jin J, Dundar A and Culurciello E. Flattened convolutional neural networks for feedforward acceleration. arXiv preprint arXiv:1412.5474, 2014.
19. Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016, pp.2818–2826.
20. Ding X, Guo Y, Ding G, et al. ACNet: strengthening the Kernel skeletons for powerful CNN via asymmetric convolution blocks. In: 2019 IEEE/CVF international conference on computer vision (ICCV), Seoul, Korea (South), 2019, pp.1911–1920. Seoul, South Korea: IEEE.
21. Ioffe S and Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proc ICML, 2015, pp.448–456. Lille, France: ACM.
22. Krizhevsky A and Hinton G. Learning multiple layers of features from tiny images. Technical Report. University of Toronto, Toronto, ON, Canada, 2009.
23. Everingham M, Van Gool L, Williams CKI, et al. The pascal visual object classes (voc) challenge. Int J Comput Vis 2010; 88: 303–338.
24. Lin T-Y, Maire M, Belongie S, et al. Microsoft COCO: common objects in context. In: Proceedings of European conference on computer vision, Springer, 2014, pp.740–755. Zurich, Switzerland: Springer.
25. Sadeghi MA and Forsyth D. 30 Hz object detection with dpm v5. In: Proceedings of ECCV, 2014, pp.65–79. Zurich, Switzerland: Springer.
26. R-cnn minus r. arXiv preprint arXiv:1506.06981, 2015.
27. Ren S, He K and Girshick R. Faster r-cnn: towards real-time object detection with region proposal networks. In: Proceedings of the international conference on neural information processing systems, Cambridge, MA, USA, 7–12 December 2015, pp.91–99. Cambridge, MA, USA: MIT.
28. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, June 2016, pp.770–778. Las Vegas, NV, USA: IEEE.
29. Redmon J and Farhadi A. Yolo9000: Better, faster, stronger. In: Proceedings of IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, July 2017, pp.6517–6525. Honolulu, HI, USA: IEEE.

Author biographies

Xianghua Ma was born in Yantai city, Shandong province, China in 1975. She received the B.S. and M.S. degrees in Electrical engineering from the University of Lanzhou Science and Technology, Gansu, in 1998 and 2002 and the Ph.D. degree in Control Theory and Control Engineering from Shanghai Jiaotong University, Shanghai, in 2006. From 2007 to 2008, she worked in Mechanical Engineering Department, Shanghai Institute of Technology. Since 2009, she has been an Assistant Professor with Electrical Engineering Department. Her research interests include optimal control, system model, and computer vision.

Zhenkun Yang received the B.S. degree in automation from Henan University of Urban Construction, Henan, China, in 2017. He is currently pursuing the M.S. degree with the School of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shanghai, China. His current research interests include computer vision, machine learning, and complex networks.