AKConv: 任意采样形状卷积核与任意参数数量卷积核

张新[1],毅　　宋ngze[1], 婷婷[1,*], 杨德刚[1,2,*], 叶一辰[3], 周杰[1], 和 张黎明 [1]

[1] 计算机与信息科学学院, 重庆师范大学[2] 重庆教育大数据智能感知与应用工程研究中心[3] 电子与信息工程学院, 西南大学 *通讯作者: 邮箱:address:{ttsong,yangdg}@cqnu.edu.cn

摘要. 基于卷积操作的神经网络在深度学习领域取得了显著成果, 但标准卷积操作存在两个固有缺陷。一方面, 卷积操作局限于局部窗口, 无法捕获其他位置的信息, 且采样形状固定。另一方面, 卷积核的大小固定为k × k, 是一个固定的方形形状, 参数数量随大小呈平方增长。显然, 目标的形状和大小在不同数据集和不同位置是多样的。固定采样形状和方形的卷积核无法很好地适应变化的目标。针对上述问题, 本文探索了可变核卷积 (AKConv), 为网络开销和性能之间的权衡提供了更丰富的选项, 该卷积核具有任意数量的参数和任意的采样形状。在AKConv中, 我们通过一种新的坐标生成算法定义了任意大小卷积核的初始位置。为了适应目标的变化, 我们引入偏移量来调整每个位置的样本形状。此外, 我们通过使用相同大小但初始采样形状不同的AKConv 探索了神经网络的影响。
AKConv 通过不规则卷积操作完成高效特征提取过程, 为卷积采样形状提供了更多探索选项。在代表性数据集COCO2017、VOC 7+12 和 VisDrone-DET2021 上进行的对象检测实验完全证明了 AKConv 的优势。AKConv 可作为即插即用的卷积操作来替代标准卷积操作, 以提高网络性能。相关任务的代码可在 https://github.com/CV-ZhangXin/AKConv.

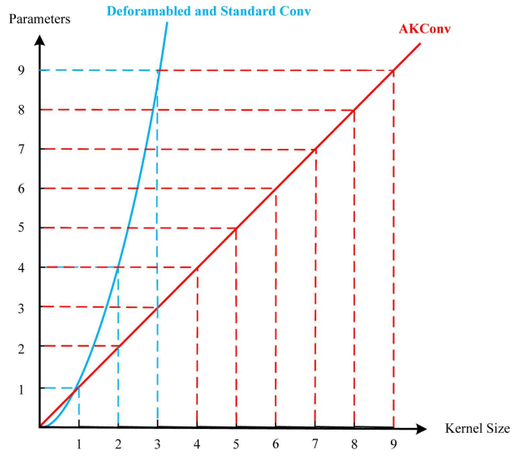

图1. 显然，与可变形卷积和标准卷积相比，AKConv具有更多选项，卷积参数的数量随卷积核大小呈线性增加。注意：为了清晰地描述AKConv的优势，在AKConv和可变形卷积中，我们忽略了学习偏移量的参数数量，因为它远小于参与特征提取的卷积参数数量。

1　引言

卷积神经网络（CNNs），如ResNet [1],DenseNet [2], 和 YOLO [3], 在各种应用中已展现出优异的性能，并在现代社会许多方面引领了技术进步。它已成为自动驾驶汽车中的图像识别 [4] 和医学图像分析 [5] 到智能监控 [6] 和个性化推荐系统 [7]不可或缺。这些成功的网络模型高度依赖卷积操作，该操作高效地提取图像中的局部特征，并确保模型复杂度。

尽管卷积神经网络（CNNS）在分类[8], 目标检测 [9], 语义分割 [10], 等方面取得了许多成功，但它们仍然存在一些局限性。其中最显著的局限性之一是卷积样本形状和大小的选择。标准卷积操作通常依赖于具有固定采样位置的方形核，例如 1 × 1, 3 × 3, 5 × 5 和 7 × 7等。常规核的采样位置是不可变形的，并且无法根据物体形状的变化动态改变。可变形卷积 [11, 12] 通过偏移量来增强网络性能，灵活调整卷积核的采样形状，使其适应目标的变化。

例如，在 [13, 14, 15], 中，他们利用它来对齐特征。赵等人 [16] 通过将其添加到 YOLOv4 [17]中，提高了检测死鱼的有效性。杨等人 [18] 通过将其添加到主干中，改进了 YOLOv8 [19] 用于检测牛。李等人 [20] 将可变形卷积引入深度图像压缩任务 [21, 22] ，以获得内容自适应感受野。

尽管上述研究已经证明了可变形卷积 (Deformable Conv) 的优越性，但它仍然不够灵活。因为卷积核仍然局限于选择核大小，并且标准卷积操作和可变形卷积中的卷积核参数数量随着卷积核大小的增加呈平方增长趋势，这对硬件环境来说并不是一种友好的增长方式。因此，在仔细分析了标准卷积操作和可变形卷积之后，我们提出了可变核卷积 (AKConv)。与标准常规卷积不同，AKConv 是一种新型的卷积操作，它可以使用具有任意参数数量（如 (1, 2, 3, 4, 5, 6, 7...)）的高效卷积核提取特征，而标准卷积和可变形卷积并没有实现这一点。AKConv 可以轻松地用于替换网络中标准卷积操作以改善网络性能。

AKConv: Convolutional Kernel with Arbitrary Sampled Shapes and Arbitrary Number of Parameters

Xin Zhang[1], Yingze Song[1], Tingting Song[1,*], Degang Yang[1,2,*], Yichen Ye[3], Jie Zhou[1], and Liming Zhang[1]

[1] College of Computer and Information Science, Chongqing Normal University
[2] Chongqing Engineering Research Center of Educational Big Data Intelligent Perception and Application
[3] College of Electronic and Information Engineering, Southwest University
*Corresponding author: E-mail:address:{ttsong,yangdg}@cqnu.edu.cn

Abstract. Neural networks based on convolutional operations have achieved remarkable results in the field of deep learning, but there are two inherent flaws in standard convolutional operations. On the one hand, the convolution operation be confined to a local window and cannot capture information from other locations, and its sampled shapes is fixed. On the other hand, the size of the convolutional kernel is fixed to k × k, which is a fixed square shape, and the number of parameters tends to grow squarely with size. It is obvious that the shape and size of targets are various in different datasets and at different locations. Convolutional kernels with fixed sample shapes and squares do not adapt well to changing targets. In response to the above questions, the Alterable Kernel Convolution (AKConv) is explored in this work, which gives the convolution kernel an arbitrary number of parameters and arbitrary sampled shapes to provide richer options for the trade-off between network overhead and performance. In AKConv, we define initial positions for convolutional kernels of arbitrary size by means of a new coordinate generation algorithm. To adapt to changes for targets, we introduce offsets to adjust the shape of the samples at each position. Moreover, we explore the effect of the neural network by using the AKConv with the same size and different initial sampled shapes. AKConv completes the process of efficient feature extraction by irregular convolutional operations and brings more exploration options for convolutional sampling shapes. Object detection experiments on representative datasets COCO2017, VOC 7+12 and VisDrone-DET2021 fully demonstrate the advantages of AKConv. AKConv can be used as a plug-and-play convolutional operation to replace convolutional operations to improve network performance. The code for the relevant tasks can be found at https://github.com/CV-ZhangXin/AKConv.

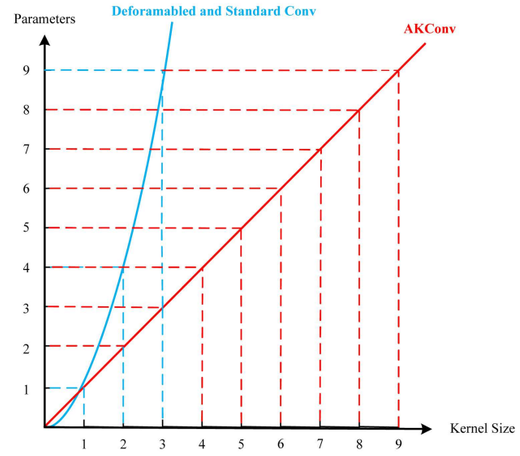

Fig. 1. It is evident that AKConv has more options compared to Deformabled and standard Conv and the number of convolutional parameters shows a linear increase with the convolutional kernel size. Note: In order to clearly describe the advantages of AKConv, in AKConv and Deformable Conv we ignore the number of parameters of the learning offset, since it is much smaller than the number of convolutional parameters involved in feature extraction.

1　Introduction

Convolutional Neural Networks (CNNs), such as ResNet [1], DenseNet [2], and YOLO [3], have demonstrated excellent performance in various applications and have led the technological progress in many aspects of modern society. It has become indispensable from image recognition in self-driving cars [4] and medical image analysis [5] to intelligent surveillance [6] and personalized recommendation systems [7]. These successful network models rely heavily on convolutional operations, which efficiently extract local features in images and ensure model complexity.

Despite the fact that CNNs have achieved many successes in classification [8], object detection [9], semantic segmentation [10], etc., they still have some limitations. One of the most notable limitations concerns the choice of convolutional sample shape and size. Standard convolution operations tend to rely on square kernels with fixed sampling locations, such as 1 × 1, 3 × 3, 5 × 5 and 7 × 7, etc. The sampling position of the regular kernel is not deformable and cannot be dynamically changed in response to changes in the shape of the object. Deformable Conv [11, 12] enhances network performance with offset to flexibly adjust the sampling shape of the convolution kernel, which adapts to the change of the target. For instance, in [13, 14, 15], they utilized it to to align features. Zhao et al. [16] improve the effectively of detection the dead fish by add it in YOLOv4 [17]. Yang et al. [18] improves the YOLOv8 [19] for detecting the cattle by add it in backbone. Li et al. [20] introduced Deformable Conv into deep image compression tasks [21, 22] to obtain content-adaptive receptive-fields.

Although the studies mentioned above have demonstrated the superior benefits of Deformable Conv. It is still not flexible enough. Because the convolution kernel is still limited to select kernel-size, and the number of convolution kernel parameters in standard convolutional operations and Deformable Conv shows a squared growth trend with the increase of the convolution kernel size, which is not a friendly way of growth to the hardware environment. Therefore, after careful analysis of standard convolution operations and Deformable Conv, we propose Alterable Kernel Convolution (AKConv). Unlike standard regular convolution, AKConv is a novel convolutional operations, which can extract features using efficient convolution kernels with any number of parameters such as (1, 2, 3, 4, 5, 6, 7...), which is not implemented by standard convolution and Deformable Convolution. AKConv can easily be used to replace the standard convolutional operations in a

网络以改善网络性能。重要的是，AKConv 允许卷积参数数量线性上升或下降，这对硬件环境有利，并且它可以作为轻量级模型的替代方案来减少模型参数和计算开销。其次，在资源充足的情况下，它在大核上有更多选项来提高网络性能。图 1 显示，常规卷积核使参数数量呈平方增长趋势，而 AKConv 仅呈线性增长趋势。与平方增长趋势相比，AKConv 增长温和，为卷积核的选择提供了更多选项。此外，其思想可以扩展到特定领域。因为，可以根据先验知识为卷积操作创建特殊的采样形状，然后通过偏移量动态和自动地适应目标形状的变化。在 VOC [23], COCO2017 [24], VisDrone-DET2021[25] 等代表性数据集上的目标检测实验完全证明了 AKConv 的优势。总之，我们的贡献如下：

1. 针对不同大小的卷积核，我们提出了一种生成任意大小卷积核初始采样坐标的算法。2. 为了适应目标的不同变化，我们通过获得的偏移量调整不规则卷积核的采样位置。3. 与常规卷积核相比，所提出的AKConv实现了不规则卷积核提取特征的功能，为各种变化的目标提供了具有任意采样形状和大小的卷积核，弥补了常规卷积的不足。

2 相关工作

近年来，许多工作从不同角度考虑和分析标准卷积操作，然后设计了新的卷积操作以提高网络性能。

齐等人。[26] 认为，在所有空间位置共享参数的卷积核会导致不同空间位置建模能力有限，并且无法有效捕获空间长距离关系。其次，为每个输出通道使用不同的卷积核实际上并不高效。因此，为了解决这些缺点，他们提出了反卷积算子，该算子反转卷积操作的特征以提高网络性能。齐等人。 [27] 基于可变形卷积提出了DSConv。从可变形卷积中学习到的偏移量是自由度，导致模型丢失一小部分细长管状结构特征，这对分割细长管状结构任务提出了巨大挑战，因此，他们提出了DSConv。张等人。[28] 从新的视角理解了空间注意力机制，他们断言，空间注意力机制本质上解决了卷积操作的参数共享问题。然而，一些空间注意力机制，如CBAM [29] 和CA [30], 并未完全解决大尺寸卷积参数共享问题。因此，他们提出了RFA-卷积。陈等人。 [31] 提出了动态卷积。与为每一层使用卷积核不同，动态卷积基于注意力动态聚合多个并行卷积核。动态卷积提供了更丰富的特征表示。谭等人。[26] 认为，卷积神经网络中经常被忽视卷积核大小，这可能会影响网络的准确性和效率。其次，仅使用逐层卷积并未充分利用卷积网络的全部潜力。因此，他们提出了混合卷积，该卷积自然地在单个卷积中混合多个核大小以提高网络性能。

卷积动态地基于注意力聚合多个并行卷积核。动态卷积提供了更丰富的特征表示。谭等人。 [32] 认为，卷积核大小在卷积神经网络中经常被忽视，这可能会影响网络的准确性和效率。其次，仅使用逐层卷积并未充分利用卷积网络的全部潜力。因此，他们提出了混合卷积，该卷积自然地在单个卷积中混合多个核大小以提高网络性能。

虽然这些方法提高了卷积操作的性能，但它们仍然局限于常规卷积操作，不允许卷积样本形状的多种变化。相比之下，我们提出的AKConv可以使用具有任意参数数量和采样形状的卷积核高效地提取特征。

3 种方法

3.1 定义初始采样位置

卷积神经网络基于卷积操作，通过常规采样网格在相应位置定位特征。在 [11, 33, 34], 中， 3×3 卷积操作的常规采样网格给出。令 R 表示采样网格，则 R 表示如下：

$$R = \{(-1,-1),(-1,0),...,(0,1),(1,1)\} \tag{1}$$

然而，采样网格是常规的，而 AKConv 目标是不规则形状的卷积核。因此，为了使不规则卷积核具有采样网格，我们创建了一种任意大小卷积的算法，该算法生成了卷积核的初始采样坐标 P_n。首先，我们将采样网格生成为一个常规采样网格，然后为其余采样点创建不规则网格，最后将它们拼接起来生成整体采样网格。伪代码如算法 1。

如图 2 所示，它显示了为任意大小卷积生成了初始采样坐标。常规卷积的采样网格以 (0, 0) 点为中心。而不规则卷积在许多大小上没有中心，为了适应所使用的卷积大小，我们在算法中将左上角 (0, 0) 点设置为采样原点。

在定义了不规则卷积的初始坐标 P_n 之后，位置 P_0 处的相应卷积操作可以定义如下：

$$Conv(P_0) = \sum w \times (P_0 + P_n) \tag{2}$$

这里， w 表示卷积参数。然而，不规则卷积操作无法实现，因为不规则采样坐标无法与相应大小的卷积操作匹配，例如大小为 5、7 和 13 的卷积。巧妙的是，我们提出的 AKConv 实现了这一点。

3.2 可变卷积操作

很明显，标准卷积采样位置是固定的，这导致卷积只能提取当前窗口的局部信息，并且

network to improve network performance. Importantly, AKConv allows the number of convolutional parameters to trend linearly up or down, which is beneficial to hardware environments, and it can be used as an alternative to lightweight models to reduce the number of model parameters and computational overhead. Secondly, it has more options to improve the network performance in large kernels with sufficient resources. Fig. 1 shows that the regular convolutional kernel makes the number of parameters to show a square increasing trend, while AKConv only shows a linear increasing trend. Compared to the square growth trend, AKConv grows gently and provides more options for the choice of convolution kernel. Furthermore, its ideas can be extended to specific areas. Because, the special sampled shapes can be created for convolution operations according to the prior knowledge, and then dynamically and automatically adapt to changes in the target shape via offset. Object detection experiments on representative datasets VOC [23], COCO2017 [24], VisDrone-DET2021 [25] fully demonstrate the advantages of AKConv. In summary, our contributions are as follows:

1. For different sizes of convolutional kernels, we propose an algorithm to generate initial sampled coordinate for convolutional kernels of arbitrary sizes.
2. To adapt to the different variations of the target, we adjust the sampling position of the irregular convolutional kernel by the obtained offsets.
3. Compared to regular convolution kernels, the proposed AKConv realizes the function of irregular convolution kernels to extract features, providing convolution kernels with arbitrary sampling shapes and sizes for a variety of varying targets, which makes up for the shortcomings of regular convolutions.

2 Related works

In recent years, many works have considered and analyzed standard convolutional operations from different perspectives, and then designed novel convolutional operations to improve network performance.

Li et al. [26] argued that convolutional kernels sharing parameters across all spatial locations, which leads to limited modeling capabilities across different spatial locations, and do not effectively capture spatially long-range relationships. Secondly, the approach of using a different convolution kernel for each output channel is actually not efficient. Therefore, to address these shortcomings, they proposed the Involution operator, which inverts the features of the convolutional operation to improve network performance. Qi et al. [27] proposed the DSConv based on Deformable Conv. The offset obtained from learning in Deformable Conv is freedom, leading to the model losing a small percentage of fine structure features, which poses a great challenge for the task of segmenting elongated tubular structures, therefore, they proposed the DSConv. Zhang et al. [28] understood the spatial attention mechanism form a new perspective, they asserted that the spatial attention mechanism essentially solves the problem of parameter sharing of convolutional operations. However, some spatial attention mechanisms, such as CBAM [29] and CA [30], not completely solve the problem of large-size convolutional parameter sharing. Therefore, they proposed RFA-Conv. Chen et al. [31] proposed the Dynamic Conv. Unlike using a convolutional kernel for every layers, the Dynamic Conv dynamically aggregated multiple parallel convolutional kernels based on their attention. The Dynamic Conv provided greater representation of features. Tan et al. [32] argued that kernel size is often neglected in CNNS, which may affect the accuracy and efficiency of the network. Second, using only layer-by-layer convolution does not utilize the full potential of convolutional networks. Therefore, they proposed MixConv, which naturally mixes multiple kernel sizes in a single convolution to improve performance of networks.

Although these methods improve the performance of convolutional operations, they are still limited to regular convolutional operations and do not allow multiple variations of convolutional sample shapes. In contrast, our proposed AKConv can efficiently extract features using a convolutional kernel with arbitrary number of parameters and sample shapes.

3 Methods

3.1 Define the initial sampling position

Convolutional neural networks are based on the convolution operation, which localizes the features at the corresponding locations by means of a regular sampling grid. In [11, 33, 34], the regular sampling grid for the 3×3 convolution operation is given. Let R denote the sampling grid, then R is denoted as follows:

$$R = \{(-1,-1),(-1,0),...,(0,1),(1,1)\} \tag{1}$$

However, the sampling grid is regular, while AKConv targets irregularly shaped convolutional kernels. Therefore, to allow irregular convolutional kernels to have a sampling grid, we create an algorithm for arbitrary size convolution, which generates the initial sampling coordinates of the convolutional kernel P_n. First, we generate the sampling grid as a regular sampling grid, then the irregular grids is created for the remaining sampling points, and finally, we stitch them to generate the overall sampling grid. The pseudo code is as in Algorithm 1.

As shown in Fig. 2, it shown that the initial sampled coordinates is generated for arbitrary size convolution. The sampling grid of the regular convolution is centered at the (0, 0) point. While the irregular convolution has no center at many sizes, to adapt to the size of the convolution used, we set the upper left corner (0, 0) point as the sampling origin in the algorithm.

After defining the initial coordinates P_n for the irregular convolution, the corresponding convolution operation at position P_0 can be defined as follows:

$$Conv(P_0) = \sum w \times (P_0 + P_n) \tag{2}$$

Here, w denotes the convolutional parameter. However, the irregular convolution operations are impossible to realize, because irregular sampling coordinates cannot be matched to the corresponding size convolution operations, e.g., convolution of sizes 5, 7, and 13. Cleverly, our proposed AKConv realizes it.

3.2 Alterable convolutional operation

It is obvious that the standard convolutional sampling position is fixed, which leads to the convolution can only extract the local information of the current window, and can

算法1 用于初始坐标生成的伪代码
类似PyTorch的卷积核。

```
# func get_p_n(num_param, dtype)
# num_param: the kernel size of AKConv
# dtype: the type of data

####### function body ########
# get a base integer to define coordinate
base_int = round(math.sqrt(num_param))
row_number = num_param // base_int
mod_numer = num_param % base_int

# get the sampled coordinate of regular kernels

p_n_x,p_n_y = torch.meshigrid(
torch.meshgrid(0, row_numb)
torch.meshgird(0, base_int))

# flatten the sampled coordinate of regular kernels
p_n_x = torch.flatten(p_n_x)
P_n_y = torch.flatten(p_n_y)

# get the sampled coordinate of irregular kernels
If mod_number > 0:
       mod_p_n_x, mod_p_n_y = torch.meshgird(
       torch.arange(row_number, row_number + 1),
       torch.arange(0, mod_number))

       mod_p_n_x = torch.flatten(mod_p_n_x)
       mod_p_n_y = torch.flatten(mod_p_n_y)
       P_n_x,p_n_y = torch.cat((p_n_x,mod_p_n_x)),torch.cat((
            p_n_y,mod_p_n_y))

# get the completed sampled coordinate
p_n = torch.cat([p_n_x, p_n_y], 0)
p_n = p_n.view(1, 2 * num_param, 1, 1).type(dtype)
return p_n
```

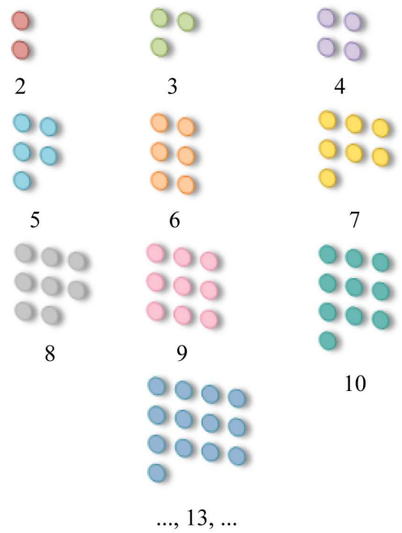

图2. 任意卷积核大小的初始采样坐标是由一个生成算法生成的。它为不规则卷积核大小提供初始采样形状。

无法捕捉其他位置的信息。可变形卷积通过卷积操作学习偏移量，以调整初始规则模式的采样网格。该方法在一定程度上弥补了卷积操作的缺点。然而，标准卷积和可变形卷积都是规则采样网格，不允许使用具有任意数量参数的卷积核。此外，随着卷积核大小的增加，其卷积参数的数量趋于平方级增长，这对硬件环境不友好。因此，我们提出了一种新颖的可变卷积操作（AKConv）。如图3所示，它说明了大小5的AKConv的整体结构。

与可变形卷积类似，在AKConv中，首先通过卷积操作获得对应卷积核的偏移量，其维度为(B, 2N, H, W)，其中N是卷积核大小。以图3 为例，N = 5。然后通过将偏移量与原始坐标相加（$P_0 + P_n$）获得修改后的坐标。最后通过插值和重采样获得对应位置的特征。难以提取不规则卷积核采样位置对应的特征。为解决这个问题，我们经过深入思考后发现有很多解决方法。在可变形卷积 [11] 和RFA卷积 [28] , 中，它们在空间维度上堆叠卷积特征。然后，使用步长为3的卷积操作提取特征。但是，这种方法针对的是方形采样形状。因此，可以将特征堆叠在行或列上，以使用列卷积或行卷积提取对应不规则采样形状的特征。特征提取使用适当大小和步长的卷积核。此外，我们可以将特征转换为四维（C, N, H, W），然后使用步长和卷积大小（N,1,1）的Conv3d提取特征。当然，我们也可以将特征在通道维度上堆叠为（CN, H, W），然后使用 1×1卷积将维度降低到（C, H, W）。所以上述所有方法都可以提取对应不规则采样形状的特征。它只需要重塑特征并使用相应的卷积操作。所以在图3 中，最终的“重塑”和“卷积”代表上述任何一种方法。

对应不规则采样形状的轨迹特征。只需要重塑特征并使用相应的卷积操作。所以在图 3，最终的“重塑”和“卷积”代表上述任何一种方法。

在RFA卷积和可变形卷积之后，我们在列方向上堆叠重采样特征，然后使用大小为(N, 1)的行卷积和步长为(N, 1)。因此，AKConv可以完美地完成不规则卷积特征提取过程。AKConv通过不规则卷积完成特征提取过程，并且可以根据偏移量灵活调整样本形状，为卷积采样形状带来更多探索选项。与标准卷积和可变形卷积不同，它们受到规则卷积核思想的限制。

3.3 扩展AK卷积

我们考虑AKConv的设计是一种新颖的设计，它实现了从不规则和任意采样形状的卷积核中提取特征的功能。即使不使用可变形卷积中的偏移量概念，AKConv仍然可以生成各种卷积核形状。因为，AKConv可以通过初始坐标重新采样，以呈现各种变化。如图4所示，我们设计了各种大小为5的卷积初始采样形状。在图4中，我们仅展示了一些大小为5的示例。然而，AKConv的大小可以是任意的，因此随着大小的增加，AKConv的初始卷积采样形状变得更加丰富，甚至无限。考虑到目标形状在不同数据集中变化，设计与采样形状对应的卷积操作至关重要。AKConv通过根据特定相位域设计相应的卷积操作，完全实现了这一点。它还可以通过添加可学习偏移量，类似于可变形卷积，动态适应对象的变化。对于特定任务，卷积核初始采样位置的设计是一个重要知识。

Algorithm 1 Pseudo-code for initial coordinate generation for convolution kernel in a PyTorch-like.

```
# func get_p_n(num_param, dtype)
# num_param: the kernel size of AKConv
# dtype: the type of data

####### function body ########
# get a base integer to define coordinate
base_int = round(math.sqrt(num_param))
row_number = num_param // base_int
mod_numer = num_param % base_int

# get the sampled coordinate of regular kernels

p_n_x,p_n_y = torch.meshigrid(
torch.meshgrid(0, row_numb)
torch.meshgird(0, base_int))

# flatten the sampled coordinate of regular kernels
p_n_x = torch.flatten(p_n_x)
P_n_y = torch.flatten(p_n_y)

# get the sampled coordinate of irregular kernels
If mod_number > 0:
       mod_p_n_x, mod_p_n_y = torch.meshgird(
       torch.arange(row_number, row_number + 1),
       torch.arange(0, mod_number))

       mod_p_n_x = torch.flatten(mod_p_n_x)
       mod_p_n_y = torch.flatten(mod_p_n_y)
       P_n_x,p_n_y = torch.cat((p_n_x,mod_p_n_x)),torch.cat((
            p_n_y,mod_p_n_y))

# get the completed sampled coordinate
p_n = torch.cat([p_n_x, p_n_y], 0)
p_n = p_n.view(1, 2 * num_param, 1, 1).type(dtype)
return p_n
```

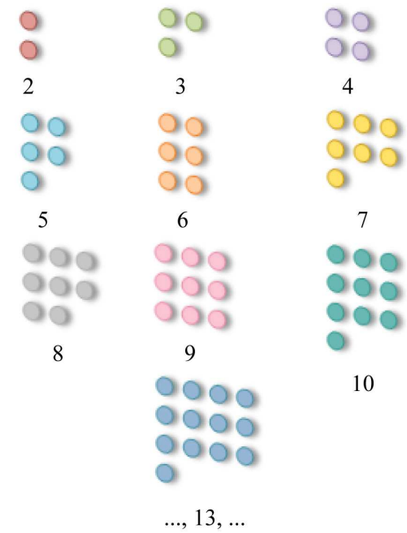

Fig. 2. The initial sampled coordinates for arbitrary convolutional kernel sizes are generated by an generation algorithm. It provides initial sampling shapes for irregular convolution kernel sizes.

not capture the information of other positions. Deformable Conv learns the offsets through convolutional operations to adjust the sampling grid of the initial regular pattern. The approach compensates for the shortcomings of the convolution operation to a certain extent. However, the standard convolution and Deformable Conv are regular sampling grids that not allow convolution kernels with arbitrary number of parameters. Moreover, as the size of the convolution kernel increases their number of convolution parameters tends to increase by a square, which is not friendly for the hardware environment. Therefore, we propose a novel Alterable convolutional operation (AKConv). As shown in Fig. 3, it illustrates the overall structure of an AKConv of size 5.

Similar to Deformable Conv, in AKConv, the offset of the corresponding kernel are first obtained by convolution operations, which has the dimensions (B, 2N, H, W), where N is the convolution kernel size. Take Fig. 3 as an example, N = 5. Then the modified coordinates are obtained by summing offset and original coordinates ($P_0 + P_n$). Finally the features at the corresponding positions are obtained by interpolating and resampling. It is difficult to extract the features corresponding to the sampled positions of the irregular convolution kernel. To solve this problem, we found that there are many ways to solve it after deep thinking. In Deformable Conv [11] and RFAConv [28] , they stack the 3 × 3 convolutional features in spatial dimensions. Then, a convolution operation with a step size of 3 is used to extract the features. However, this method targets square sampling shapes. Therefore, the features can be stacked on rows or columns to use the column convolution or row convolution to extract features corresponding to irregular sampling shapes. The features are extracted to use a convolutional kernel of the appropriate size and step size. Moreover, we can transform the features into four dimensions (C, N, H, W), and then use Conv3d with step size and convolution size (N,1,1) to extract the features. Of course, we can also stack the features on the channel dimension to (CN, H, W), and then use 1×1 convolution to reduce the dimension to (C, H, W). So all these methods mentioned above can extract features corresponding to irregularly sampled shapes. It is only necessary to reshape features and use the corresponding convolution operation. So in Fig. 3, the final "Reshape" and "Conv" represent any of the above methods.

Following RFAConv and Deformable Conv, we stack the resampled features in the column direction and then use row convolution with size (N, 1) and step size (N, 1). Therefore, AKConv can perfectly accomplish the irregular convolutional feature extraction process. AKConv completes the process of feature extraction by irregular convolution, and it can flexibly adjust the sample shape according to the offset and bring more exploration options for convolutional sampling shapes. Unlike Standard Convolution and Deformable Conv, they are limited by the idea of a regular convolution kernel.

3.3 Extended AKConv

We consider the design of AKConv to be a novel design that accomplishes the feat of extracting features from irregular and arbitrarily sampled shape convolutional kernels. Even without using the offset idea in Deformable Conv, AKConv can still make a variety of convolution kernel shapes. Because, AKConv can resample with the initial coordinates to present a variety of changes. As shown in Fig. 4, we design various initial sampling shapes for convolution of size 5. In Figure 4, we only show some examples of size 5. However, the size of AKConv can be arbitrary, therefore as the size increases, the initial convolutional sampling shapes of AKConv become richer and even infinite. Given that the target shape varies across datasets, it is crucial to design the convolution operation corresponding to the sampled shape. AKConv fully realizes it by designing the convolution operation with the corresponding shape according to the phase-specific domain. It can also be similar to Deformable Conv by adding a learnable offset to dynamically adapt to changes of the object. For a specific task, the design of the initial sampling location of the convolution kernel is an important, because it is an a prior

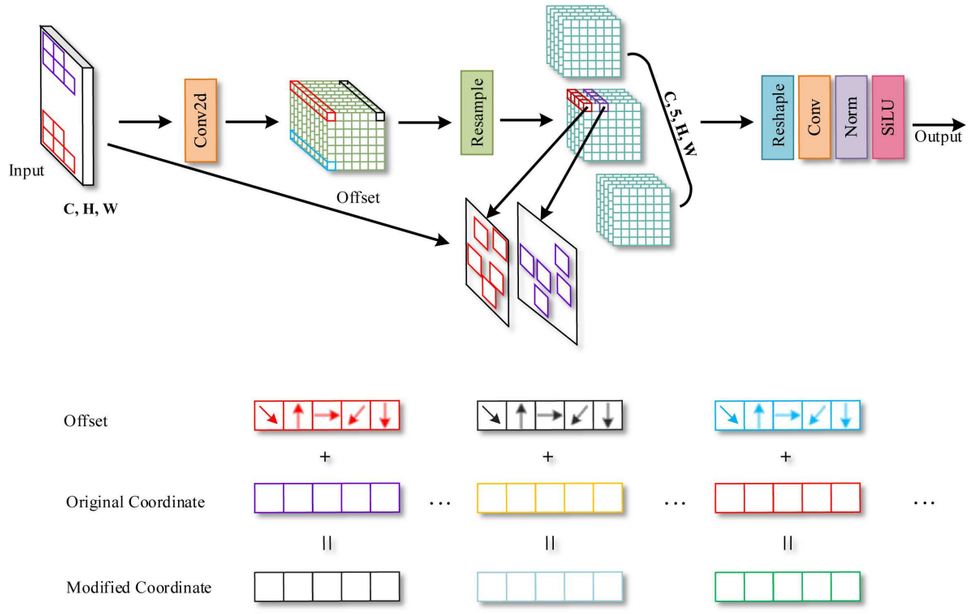

图3。它展示了AKConv结构的详细示意图。它为任意大小的卷积分配初始采样坐标，并使用可学习偏移量调整样本形状。

知识。如齐等人。[27], 他们为细长管状结构分割任务提出了具有相应形状的采样坐标，但他们的形状选择仅针对细长管状结构。

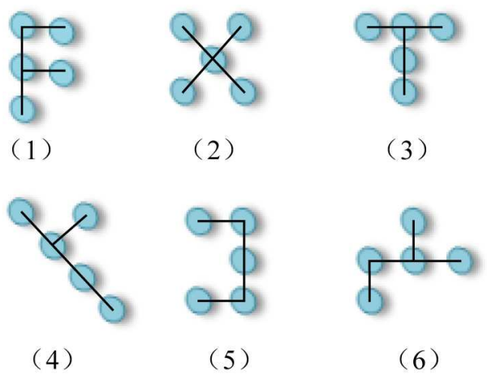

图4。它展示了大小5的初始样本形状。AKConv可以通过设计不同的初始采样形状来实现任意采样形状。

AKConv 真正实现了以任意形状的任意数量进行卷积核操作的流程，并且它可以使卷积核呈现出多种形状。可变形卷积 [11] 被设计用来弥补常规卷积的不足。而 DSConv [27] 被设计用于特定的对象形状。它们没有探索任意大小的卷积和任意样本形状的卷积。AKConv 的设计通过允许卷积操作通过偏移量高效地提取不规则样本形状的特征。AKConv 允许卷积具有任意数量的卷积参数，并且允许卷积呈现出多种形状。

4 实验部分

为了验证AKConv的优势，我们基于先进的YOLOv5 [35]，YOLOv7 [36] 和YOLOv8 [19] 分别进行了丰富的目标检测实验。实验中的所有模型均基于RTX3090进行训练。为了验证AKConv的优势，我们在代表性的COCO2017、VOC 7 + 12和VisDrone- DET2021数据集上分别进行了相关实验。

4.1 COCO2017上的目标检测实验

COCO2017 包括训练集（118287张图像）、验证集（5000张图像），并涵盖80个目标类别。它已成为计算机视觉研究领域的一个标准数据集，特别是在目标检测领域。我们选择了最先进的 YOLOv5n 和 YOLOv5s 检测器作为基线模型。然后，使用不同大小的 AK-卷积替换 YOLOv5n 和 YOLOv5s 的卷积操作。替换的细节与目标检测实验中的 [28] 相同。在实验中，除了 epoch 和 batch-size 参数外，网络使用默认参数。基于批量大小为32，我们训练每个模型300个 epoch。遵循先前的工作，我们报告 AP_{50}, AP_{75}, AP, AP_S,AP_M 和 AP_L。此外，我们还分别报告了大小为5、4、6、7、9 和 13 的 AKConv 在 YOLOv5n 和 YOLOv5s 上的目标检测结果。如表 1 所示，随着卷积核大小的增加，YOLOv5 的检测精度逐渐提高，而模型所需的参数数量和计算开销也逐渐增加。与标准卷积操作相比，AKConv 显著提高了 YOLOv5 在 COCO2017 上的目标检测性能。可以看出，当 AKConv 的大小为5时，它不仅减少了模型所需的参数数量和计算开销，还显著提高了 YOLOv5n 的检测精度。其 {v1},{v2} 和 AP 都提高了三个百分点，这非常出色。AKConv 提高了基线模型的 [28],{v4} 和 {v5}，但很明显，与小型和中型物体相比，AKConv 显著提高了大型物体的检测精度。我们断言，AKConv 使用偏移量来更好地适应大型物体的形状。

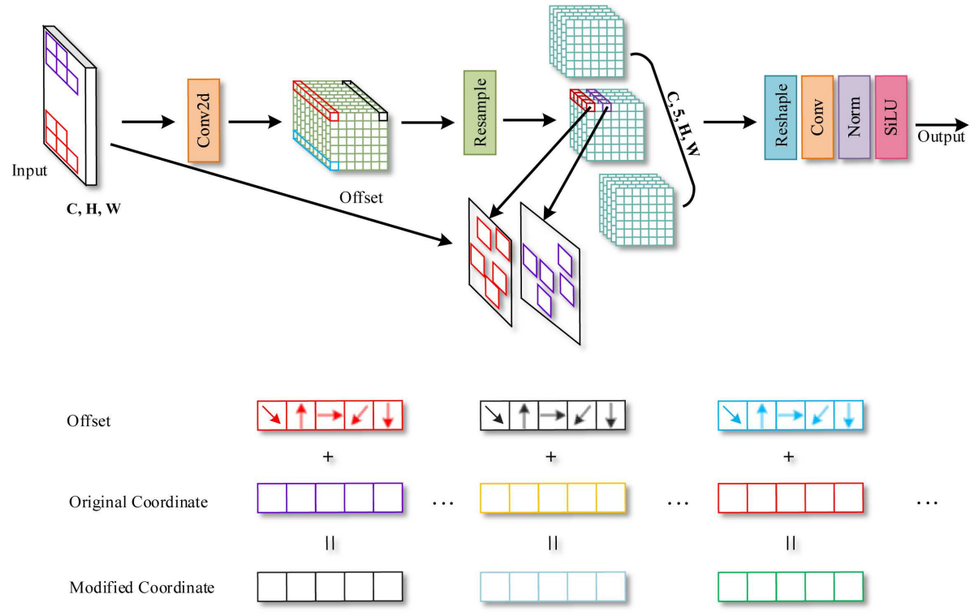

Fig. 3. It shows a detailed schematic of the structure of AKConv. It assigns initial sampling coordinates to a convolution of arbitrary size and adjusts the sample shape with the learnable offsets.

knowledge. As in Qi et al. [27], they proposed sampling coordinates with corresponding shapes for the elongated tubular structure segmentation task, but their shape selection was only for elongated tubular structures.

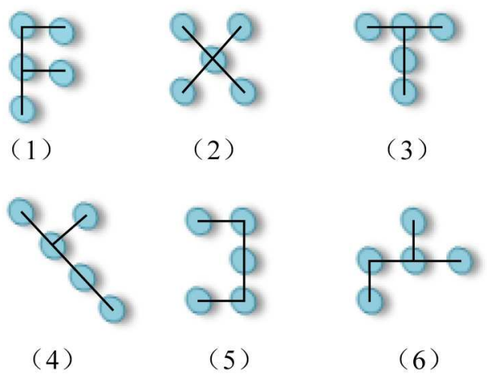

Fig. 4. It shows the initial sample shape of size 5. AKConv can achieve arbitrary sampling shapes by designing different initial sampling shapes.

AKConv really achieves the process of convolution kernel operation with any number of arbitrary shapes, and it can make the convolution kernel present a variety of shapes. Deformable Conv [11] was designed to compensate for the shortcomings of regular convolution. Whereas DSConv [27] was designed for specific object shapes. They have not explored convolution of arbitrary size and convolution of arbitrary sample shapes. The design of AKConv remedies these problems by allowing the convolution operation to efficiently extract the features of irregular sample shapes through Offset. AKConv allows the convolution to have any number of convolution parameters, and allows the convolution to take on a wide variety of shapes.

4 Experiments

To verify the advantages of AKConv, we conduct rich target detection experiments based on advanced YOLOv5 [35], YOLOv7 [36] and YOLOv8 [19] respectively. All models in the experiments are trained based on RTX3090. To validate the advantages of AKConv we perform related experiments on representative COCO2017, VOC 7 + 12 and VisDrone-DET2021 datasets respectively.

4.1 Object detection experiments on COCO2017

COCO2017 includes train (118287 images), val (5000 images), and covers 80 object classes. It has become a standard dataset in the field of computer vision research, especially in the field of target detection. We chose the state-of-the-art YOLOv5n and YOLOv5s detectors as the baseline model. Then, AKConv with different sizes is used to replace the convolution operations of YOLOv5n and YOLOv5s. The replacement details are the same as the target detection experiments in [28]. In the experiments, the default parameters of the network are used except for the epoch and batch-size parameters. Based on a batch size of 32, we trained each model for 300 epochs. Following previous work, we report AP_{50}, AP_{75}, AP, AP_S, AP_M and AP_L. Moreover, we also report target detection on YOLOv5n and YOLOv5s for AKCOnv with sizes 5, 4, 6, 7, 9, and 13, respectively. As shown in Table 1, the detection accuracy of YOLOv5 gradually increases with the increase of

模型	AKConv	AP_{50}(%)	AP_{75}(%)	AP(%)	AP_S(%)	AP_M(%)	AP_L(%)	GFLOPS	Params(M)
YOLOv5n (Baseline)	-	45.6	28.9	27.5	13.5	31.5	35.9	4.5	1.87
YOLOv5n	3	47.8	31.1	29.8	14.5	33.2	41	3.8	1.51
	5	48.8	32.6	31	14.6	34.1	43.2	4.1	1.65
	9	50.5	33.9	32.3	14.9	36.1	44.1	4.8	1.94
	13	51.2	34.5	33	15.7	36.3	45.6	5.5	2.23
YOLOv5s (基线)	-	57	39.9	37.1	20.9	42.4	47.8	16.4	7.23
YOLOv5s	4	58.2	41.9	39.2	21.4	43.2	53.4	14.1	6.01
	6	59.2	42.6	39.9	21.5	44.2	54.7	15.3	6.55
	7	59.4	43.2	40.4	21.5	44.6	55.1	15.9	6.82

表1. 目标检测 AP_{50}, AP_{75}, AP, AP_S, AP_M，以及 AP_L 在 COCO2017 验证集上。我们采用 YOLOv5n 和 YOLOv5s 检测框架，并将原始卷积替换为不同尺寸的 AKConv。

模型	AKConv	精度(%)	召回率(%)	mAP50(%)	mAP(%)	GFLOPS	参数(M)
YOLOv7-tiny (基线)	-	77.3	69.8	76.4	50.2	13.2	6.06
YOLOv7-tiny	3	80.1	68.4	76.1	50.3	12.1	5.56
	4	78.2	70.3	76.2	50.7	12.4	5.66
	5	77	71.1	76.5	50.8	12.6	5.75
	6	79.6	69.9	76.9	51	12.9	5.85
	8	78.6	70.1	76.7	51.2	13.4	6.04
	9	81	69.3	76.7	51.3	13.7	6.14

表 2. 基于 b　　基线数据　在 VOC 7 + 12 数据集上，它表明 AKConv 可以提高 YOLOv7-tiny 的 mAP50 和 mAP。

与标准卷积操作相比，AKConv 显著提高了 YOLOv5 在 COCO2017 上的目标检测性能。可以看出，当 AKConv 的大小为5时，它不仅减少了模型所需的参数数量和计算开销，还显著提高了 YOLOv5n 的检测精度。其 AP_{50}, AP_{75} 和 AP 都提高了三个百分点，这非常出色。AKConv 提高了基线模型的 AP_S, AP_M 和 AP_L，但很明显，与小型和中型物体相比，AKConv 显著提高了大型物体的检测精度。我们断言，AKConv 使用偏移量来更好地适应大型物体的形状。

4.2 在 VOC 7+12 上的目标检测实验

为了进一步验证我们的方法，我们在VOC 7+12数据集上进行了实验，该数据集是VOC2007和VOC2012的组合，包含16,551个训练集和4,952个验证集，涵盖20个目标类别。为了测试AKConv在不同架构下的泛化能力，我们选择了YOLOv7-tiny作为基线模型。由于YOLOv7和YOLOv5是具有不同架构的系统，因此可以比较AKConv在不同架构设置下的性能。在YOLOv7-tiny中，我们使用不同尺寸的AKConv来替换标准卷积操作。替换的细节遵循[28]中的工作。所有模型的超参数设置与前一部分一致。遵循先前的工作，我们同时报告了mAP50和mAP。如表2所示，随着AKConv中尺寸的增加，网络的检测精度逐渐提高，而模型的参数数量和计算需求也逐步上升。这些实验进一步证实了AKConv的优势。

网络的检测精度逐渐提高，而模型的参数数量和计算需求也逐步上升。这些实验进一步证实了AKConv的优势。

4.3 在VisDrone-DET2021上的目标检测实验

为了再次验证AKConv具有强大的泛化能力，基于VisDrone-DET2021数据集，我们进行了相关的目标检测实验。VisDrone-DET2021是一个由无人机在不同环境、天气和光照条件下拍摄的大型数据集，是中国无人机航拍覆盖范围最广的数据集之一。训练集数量为6471，验证集数量为548。如第4.1节所述，我们选择YOLOv5n作为基线，使用AKConv替换网络中的卷积操作。在实验中，batch-size设置为16，以促进对更大卷积尺寸的探索，并且所有其他超参数设置与之前相同。如上一节所述，我们分别报告了mAP50和mAP。如表3所示，很明显，基于不同尺寸的AKConv可以作为轻量级选项来减少参数数量和计算开销，并提高网络性能。在实验中，当AKConv的尺寸设置为3时，与基线模型相比，模型的检测性能有所下降，但相应的参数数量和计算开销要小得多。此外，我们可以逐渐调整AKConv的尺寸，以探索网络性能的变化。AKConv为网络带来了更丰富的选择。

Models	AKConv	AP_{50}(%)	AP_{75}(%)	AP(%)	AP_S(%)	AP_M(%)	AP_L(%)	GFLOPS	Params(M)
YOLOv5n (Baseline)	-	45.6	28.9	27.5	13.5	31.5	35.9	4.5	1.87
YOLOv5n	3	47.8	31.1	29.8	14.5	33.2	41	3.8	1.51
	5	48.8	32.6	31	14.6	34.1	43.2	4.1	1.65
	9	50.5	33.9	32.3	14.9	36.1	44.1	4.8	1.94
	13	51.2	34.5	33	15.7	36.3	45.6	5.5	2.23
YOLOv5s (Baseline)	-	57	39.9	37.1	20.9	42.4	47.8	16.4	7.23
YOLOv5s	4	58.2	41.9	39.2	21.4	43.2	53.4	14.1	6.01
	6	59.2	42.6	39.9	21.5	44.2	54.7	15.3	6.55
	7	59.4	43.2	40.4	21.5	44.6	55.1	15.9	6.82

Table 1. Object detection AP_{50}, AP_{75}, AP, AP_S, AP_M, and AP_L on the COCO2017 validation sets. We adopt the YOLOv5n and YOLOv5s detection framework and replace the original convolution with the different size AKConv.

Models	AKConv	Precison(%)	Recall(%)	mAP50(%)	mAP(%)	GFLOPS	Params(M)
YOLOv7-tiny (Baseline)	-	77.3	69.8	76.4	50.2	13.2	6.06
YOLOv7-tiny	3	80.1	68.4	76.1	50.3	12.1	5.56
	4	78.2	70.3	76.2	50.7	12.4	5.66
	5	77	71.1	76.5	50.8	12.6	5.75
	6	79.6	69.9	76.9	51	12.9	5.85
	8	78.6	70.1	76.7	51.2	13.4	6.04
	9	81	69.3	76.7	51.3	13.7	6.14

Table 2. Based on the baseline dataset VOC 7 + 12, it is shown that AKConv can improves the mAP50 and mAP for YOLOv7-tiny.

the convolutional kernel size, while the number of parameters required by the model and the computational overhead also gradually increase. Compared to standard convolutional operations, AKConv substantially improves the target detection performance of YOLOv5 on COCO2017. It can be seen that when the size of AKConv is 5, it not only makes the number of parameters and computational overhead required by the model decrease, but also significantly improves the detection accuracy of YOLOv5n. Its AP_{50}, AP_{75}, and also AP are all improved by three percentage points, which is outstanding. AKConv improves the AP_S, AP_M, and AP_L of the baseline model, but it is obvious that AKConv improves the detection accuracy of large objects significantly compared to small and middle objects. We assert that AKConv uses offsets to better adapt to the shape of large objects.

4.2 Object detection experiments on VOC 7+12

In order to further validate our method, we conduct experiments on the VOC 7+12 dataset, which is a combination of VOC2007 and VOC2012, comprising 16,551 training sets and 4,952 validation sets, and covers 20 object categories. To test the generalizability of AKConv across different architectures, we selected YOLOv7-tiny as the baseline model. Since YOLOv7 and YOLOv5 are systems with different architectures, it is possible to compare the performance of AKConv with different architectural settings. In YOLOv7-tiny, we use AKConv with different sizes to replace standard convolutional operation. The details of the replacement follows the work in [28]. The hyperparameter settings for all models are consistent with those in the previous section. Following previous work, we present both mAP50 and mAP. As demonstrated in Table 2, with the increasement of size in AKConv, the network's detection accuracy gradually improves, while the model's parameter count and computational demand also incrementally rise. These experiments further substantiate the advantages of AKConv.

4.3 Object detection experiments on VisDrone-DET2021

In order to verify again that AKConv has strong generalization ability, based on VisDrone-DET2021 data, we conducted relevant target detection experiments. VisDrone-DET2021 is a challenging dataset taken by UAVs in different environments, weather and lighting conditions. It is one of the largest datasets with the widest coverage of UAV aerial photography in China. The number of training sets is 6471 and the number of validation sets is 548. As in Section 4.1, we chose YOLOv5n as the baseline to use AKConv to replace convolutional operations in the network. In experiments, the batch-size is set 16 to facilitate the exploration of larger convolution sizes, and all other hyperparameter settings are the same as before. As in the previous section, we report mAP50 and mAP, respectively. As shown in Table 3, it is clear to see that AKConv based on different sizes can be used as a lightweight option to reduce the number of parameters and computational overhead and improve network performance. In experiments, when the size of AKConv is set to 3, the detection performance of the model decreases compared to the baseline model, but the corresponding number of parameters and computational overhead are much smaller. Moreover, we can gradually adjust the size of AKConv to explore the changes in network performance. AKConv brings richer options to the network.

模型	AKConv	精度(%)	召回率(%)	mAP50(%)	mAP(%)	GFLOPS	Params(M)
YOLOv5n (Baseline)	-	38.5	28	26.4	13.4	4.2	1.77
YOLOv5n	3	37.9	27.4	25.9	13.2	3.5	1.41
	5	40	28	26.9	13.7	3.8	1.56
	6	38.1	28.1	26.8	13.6	4	1.63
	7	39.8	28.2	27.5	14.2	4.2	1.7
	9	39.7	28.9	27.7	14.3	4.5	1.84
	11	40.4	28.8	27.7	14.2	4.8	1.99
	14	40	28.8	27.9	14.3	5.3	2.2

表3. 使用不同尺寸的AKConv替换卷积操作，在VisDrone-DET2021验证集上进行的对象检测mAP50和mAP。

模型	AP_{50}(%)	AP_{75}(%)	AP(%)	AP_S(%)	AP_M(%)	AP_L(%)	GFLOPS	Params(M)
YOLOv5s	54.8	37.5	35	19.2	40	45.2	16.4	7.23
YOLOv5s (DSConv=5)	43.2	23.5	23.9	13	27.6	30.5	14.8	6.45
YOLOv5s (AKConv=5)	56.6	40.7	38	20.8	41.8	52	14.8	6.45
YOLOv5s (AKConv=9)	57.8	41.4	38.7	20.8	42.8	52.3	17.1	7.37
YOLOv5s (AKConv=9, Padding)	58.3	41.9	39.2	21.6	43.2	53.5	17.1	7.37
YOLOv5s (可变形卷积 = 3)	58.5	41.8	39.1	20.8	43.4	53.6	17.1	7.37
YOLOv5s (AKConv=11)	58.5	42.1	39.3	21.9	43.3	53.8	18.3	7.91
YOLOv5s (AKConv=11, Padding)	58.6	42.1	39.5	21.3	43.7	53.2	18.3	7.91

表4. 目标检测 AP_{50}, AP_{75}, AP, AP_S, AP_M, 和 AP_L 在 COCO2017 验证集上。我们比较了 AKConv、可变形卷积和 DSConv 的性能，它们具有相同的大小。

4.4 对比实验

与可变形卷积 [11], AKConv为网络提供了更丰富的选择。AKConv弥补了可变形卷积的不足，可变形卷积仅使用常规卷积操作，而AKConv可以使用常规和不规则卷积操作。当AKConv的尺寸设置为K的平方时，AKConv成为可变形卷积。此外，DSConv[27] 也使用偏移量来调整采样形状，但其采样形状是为管状目标设计的，采样形状的变化有限。为了对比AKConv、可变形卷积和DSConv在相同尺寸下的优势，我们在COCO2017和VOC7 + 12 上基于YOLOv5s和YOLOv5n进行实验。如表4 和表5所示。当卷积核参数数量为9（即标准 3 × 3 卷积）时，可以看出AKConv和可变形卷积的性能相同。因为当卷积核尺寸为常规尺寸时，AKConv就是可变形卷积。但我们已经提到可变形卷积没有探索不规则卷积核尺寸。因此，参数数量为5和11的卷积操作无法实现。在设计AKConv时，我们没有对输入特征进行零填充。然而，在可变形卷积中使用了填充。因此，为了进行公平比较，在AKConv中，我们也对输入特征使用了零填充。实验表明，AKConv中的零填充有助于网络提高性能。由于DSConv是为特定的管状形状设计的，可以看出其在COCO2017和VOC 7 + 12 上的检测性能并不明显。在实现DSConv时，齐等人。 [27] 扩展了行或列的特征，最后使用了列卷积或列卷积来提取类似于我们的特征。所以他们的方法也可以使用参数 2、3、4、5、6、7 等实现卷积操作。在相同尺寸下，我们也进行了对比实验。因为 DSConv 没有完成下采样方法，在实验中，我们使用 AKConv 和 DSConv 来替换 3 × 3 YOLOv5n 中 C3 的卷积4 和 5表中的实验结果。AK-卷积比 DSConv 更有优势，因为 DSConv 旨在探索特定形状的目标，而不是设计来提高任意尺寸卷积核的性能。相比之下，AKConv 提供了丰富的卷积核选择和探索方式，可以有效地提高网络性能。

4.5 探索初始采样形状

如前所述，AKConv 可以通过使用任意大小和任意采样形状来提取特征。为了探索具有不同初始采样形状的 AKConv 对网络的影响，我们在 COCO2017 和 VisDrone-DET2021 上分别进行了实验。在 COCO2017 上，我们基于 batch-size 为 32 和 epoch 为 100 进行了实验。在 VisDrone-DET2021 上，我们基于 batch-size 为 16 和 epoch 为 300 进行了实验。所有其他超参数都是网络默认值。在 COCO2017 上，我们选择了 YOLOv8n 进行实验。如表6所示，AKConv 仍然可以提高网络的检测精度。YOLOv8 和 YOLOv5 的网络结构相似。区别在于 C3 和 C2f 的设计。可以看出，在 YOLOv8 中添加 AKConv 所获得的性能提升不如 YOLOv5。我们认为，在相同大小下，YOLOv8 需要的参数比 YOLOv5 多，因此更多的参数可以提供更好的

Models	AKConv	Precison(%)	Recall(%)	mAP50(%)	mAP(%)	GFLOPS	Params(M)
YOLOv5n (Baseline)	-	38.5	28	26.4	13.4	4.2	1.77
YOLOv5n	3	37.9	27.4	25.9	13.2	3.5	1.41
	5	40	28	26.9	13.7	3.8	1.56
	6	38.1	28.1	26.8	13.6	4	1.63
	7	39.8	28.2	27.5	14.2	4.2	1.7
	9	39.7	28.9	27.7	14.3	4.5	1.84
	11	40.4	28.8	27.7	14.2	4.8	1.99
	14	40	28.8	27.9	14.3	5.3	2.2

Table 3. Object detection mAP50 and mAP on the VisDrone-DET2021 validation set by using different size of AKConv to replace convolutional operation.

Models	AP_{50}(%)	AP_{75}(%)	AP(%)	AP_S(%)	AP_M(%)	AP_L(%)	GFLOPS	Params(M)
YOLOv5s	54.8	37.5	35	19.2	40	45.2	16.4	7.23
YOLOv5s (DSConv=5)	43.2	23.5	23.9	13	27.6	30.5	14.8	6.45
YOLOv5s (AKConv=5)	56.6	40.7	38	20.8	41.8	52	14.8	6.45
YOLOv5s (AKConv=9)	57.8	41.4	38.7	20.8	42.8	52.3	17.1	7.37
YOLOv5s (AKConv=9, Padding)	58.3	41.9	39.2	21.6	43.2	53.5	17.1	7.37
YOLOv5s (Deformable Conv = 3)	58.5	41.8	39.1	20.8	43.4	53.6	17.1	7.37
YOLOv5s (AKConv=11)	58.5	42.1	39.3	21.9	43.3	53.8	18.3	7.91
YOLOv5s (AKConv=11, Padding)	58.6	42.1	39.5	21.3	43.7	53.2	18.3	7.91

Table 4. Object detection AP_{50}, AP_{75}, AP, AP_S, AP_M, and AP_L on the COCO2017 validation sets. We compare the performance of the AKConv, Deformable Conv and DSConv with same size.

4.4 Comparison experiments

Unlike Deformable Conv [11], AKConv offers a richer choice for networks. AKConv compensates for the shortcomings of Deformable Conv, which only uses regular convolution operations, while AKConv can use both regular and irregular convolution operations. When the size of AKConv is set to the square of K, AKConv becomes a deformable Conv. Moreover, DSConv [27] also uses offsets to adjust the sampling shapes, but its sampling shape is designed for tubular targets, and the change of the sampling shape is limited. To contrast the advantages of AKConv, Deformable Conv, and DSConv at the same size. We perform experiments in COCO2017 and VOC 7 + 12 based on YOLOv5s and YOLOv5n. As shown in the Table 4 and Table 5. When the number of convolution kernel parameters is 9 (i. e., the standard 3 × 3 convolution), it can be seen that the performance of AKConv and Deformable Conv is the same. Because when the convolution kernel size is regular, the AKConv is the Deformable Conv. But we have mentioned that Deformable Conv has not explored the irregular convolution kernel size. Therefore, a convolution operation with a number of parameters of 5 and 11 cannot be implemented. When designing AKConv, we not implement zero-padding for input features. However, in Deformable Conv padding is used. Therefore, for a fair comparison, in AKConv, we also utilize zero-padding for input features. Experiments show that zero-padding in AKConv helps the network to improve performance. Since DSConv is designed for a specific tubular shape, it can be seen that its detection performance on COCO2017 and VOC 7 + 12 is not obvious. When implementing DSConv, Qi et al. [27] expands the features of rows or columns, and finally used column convolution or columns convolution to extract features similar to us. So their method can also implement convolution operations with parameters 2, 3, 4, 5, 6, 7, etc. Under the same size, we also conduct a comparison experiment. Because, the DSConv not completes the down-sample method, in experiments, we use the AKConv and DSConv to replace 3 × 3 convolution in C3 for YOLOv5n. Experimental results are shown in Table 4 and Table 5. AKConv is advantageous over DSConv, because DSConv is not designed to improve the performance of convolutional kernels of arbitrary size, but rather to explore for targets of specific shapes. In contrast, AKConv provides a rich choice of convolutional kernel selection and exploration that can effectively improve network performance.

4.5 Exploring the initial sampled shape

As mentioned earlier, AKConv can extract features by using arbitrary sizes and arbitrary sample shapes. To explore the effect of AKConv with different initial sample shapes on the network, we conducted experiments at COCO2017 and VisDrone-DET2021, respectively. On COCO2017, we conducted experiments based on a batch-size of 32 and an epoch of 100. In VisDrone-DET2021, we conducted experiments based on a batch-size of 16 and an epoch of 300. All other hyperparameters are network defaults. In COCO2017, we chose YOLOv8n for our experiments. As shown in Table 6, AKConv can still improve the detection accuracy of the network. The network structures of YOLOv8 and YOLOv5 are similar. The difference is the design of C3 and C2f. It can be seen that the performance increase obtained by adding AKConv in YOLOv8 is not as good as in YOLOv5. We think that YOLOv8 needs more parameters than YOLOv5 under the same size, so more number of parameters can provide better

模型	精度(%)	召回率(%)	mAP50(%)	mAP(%)	GFLOPS	Params(M)
YOLOv5n	73.8	62.2	68.1	41.5	4.2	1.77
YOLOv5n (DSConv=4)	63	50.4	54.2	26.1	3.7	1.55
YOLOv5n (AKConv=4)	76.5	63.6	70.8	46.5	3.7	1.55
YOLOv5n (DSConv=9)	60.6	50.8	53.4	25.3	4.8	1.9
YOLOv5n (AKConv=9)	76.7	65.2	71.8	48.4	4.8	1.9

表5。基于VOC 7 + 12，我们比较了其他尺寸的AKConv和DSConv，并分别报告了检测精度和其他评估指标。

模型	AP_{50}(%)	AP_{75}(%)	AP	AP_S(%)	AP_M(%)	AP_L(%)	GFLOPS	Params(M)
YOLOv8n	49	37.1	34.2	16.9	37.1	49.1	8.7	3.15
YOLOv8n-5 (采样形状 1)	49.5	37.6	34.9	16.8	38.2	50.2	8.4	2.94
YOLOv8n-5 (采样形状 2)	49.6	37.8	34.9	15.9	38.4	50.1	8.4	2.94
YOLOv8n-5（采样形状3）	49.6	38.1	35	16.6	38.2	50.9	8.4	2.94
YOLOv8n-6 (采样形状 1)	50.1	38.3	35.3	16.6	38.6	51.1	8.6	3.01
YOLOv8n-6（采样形状2）	50.2	38.2	35.4	16.6	38.3	51.3	8.6	3.01

表6。基于COCO2017 an d YOLOv8n，我们探索了具有不同初始采样形状的AKConv的不同尺寸。The “采样形状 i”表示不同的 AKConv 初始采样形状。

模型	初始形状	精度	召回率	mAP50(%)	mAP(%)
YOLOv5n	a	39.5	27.9	26.9	13.7
	b	39.4	28.2	26.8	13.6
	c	37.4	27.8	26.1	13.4
	d	37.5	27	25.5	12.9
	e	38.4	27.6	26.4	13.4

表7。它表明不同的 AK-卷积 初始采样形状在 VisDrone-DET2021 上获得了 YOLOv5n 的性能。

特征信息，就像 AKConv 做的那样。因此，在添加 AKConv 后，YOLOv8 的提升并不像 YOLOv5 那么显著。此外，在相同大小下，我们在 COCO2017 上测试了不同初始采样形状对网络性能的影响。很明显，在不同的初始样本下，网络获得的检测精度波动并不大。这得益于 COCO2017 的大量数据可以灵活调整偏移量。但是，这并不意味着网络在所有初始采样坐标上获得的检测精度都没有显著差异。为了再次探索具有不同初始形状的 AKConv 对网络的影响，我们在 VisDrone-DET2021 上基于 YOLOv5n 探索了大小为 5 且具有不同初始样本的 AKConv。如表 7 所示，网络在不同初始样本下获得了不同的检测精度。因此，具有不同初始采样形状的 AKConv 对网络性能有影响。此外，对于特定的网络和数据集，探索具有适当初始采样形状的 AKConv 以提高网络性能非常重要。

5 分析与讨论

我们在之前的实验中，在不同的采样位置初始AKConv大小5，以观察检测YOLOv5n的性能。可以明显地注意到，网络在不同初始采样形状下表现不同。这表明偏移量的调整能力也是有限的。为了测量每个给定位置的偏移量变化，我们给出了平均偏移的定义，其定义如下：

$$AO = (\sum_{i}^{2N} |Offset_i|)/(2N) \quad (3)$$

AO（平均偏移）通过求和偏移量并取平均值来衡量每个位置采样点的平均变化程度。为了观察偏移量的变化，我们选择了训练好的网络，并选择了AKConv的最后一层来分析偏移量的整体变化趋势。在分析中，我们从VisDrone-DET2021中随机选择了四张图像，然后可视化了大小为5的AKConv，这是针对不同采样位置的初始形状。如图5所示，我们可视化了每个采样位置的偏移量变化程度AO。图5中的不同颜色表示训练后不同初始样本在每个采样位置的偏移量变化。线的颜色对应中间的初始采样形状。图5 中的不同初始采样形状对应表7中的初始采样形状。可以得出结论，图5中的蓝色和红色初始采样形状变化较小。这意味着红色和蓝色初始样本比其他初始样本更适合这个数据集。如表7中的实验所示，可以看出蓝色和红色对应的初始采样形状获得了更好的检测精度。所有实验证明，AKConv能够为网络带来显著的性能提升。与可变形卷积不同，AKConv具有根据大小调整网络性能的灵活性。在所有实验中，我们广泛探索了大小为5的AKConv。因为当使用大量数据训练COCO2017时，我们发现当将AKConv的大小设置为5时，训练速度与原始模型没有太大差异。此外，随着AKConv大小的增加，训练时间

Models	Precison(%)	Recall(%)	mAP50(%)	mAP(%)	GFLOPS	Params(M)
YOLOv5n	73.8	62.2	68.1	41.5	4.2	1.77
YOLOv5n (DSConv=4)	63	50.4	54.2	26.1	3.7	1.55
YOLOv5n (AKConv=4)	76.5	63.6	70.8	46.5	3.7	1.55
YOLOv5n (DSConv=9)	60.6	50.8	53.4	25.3	4.8	1.9
YOLOv5n (AKConv=9)	76.7	65.2	71.8	48.4	4.8	1.9

Table 5. Based on VOC 7 + 12, we compared other sizes of AKConv and DSConv and reported detection accuracy and other evaluation metrics, respectively.

Models	AP_{50}(%)	AP_{75}(%)	AP	AP_S(%)	AP_M(%)	AP_L(%)	GFLOPS	Params(M)
YOLOv8n	49	37.1	34.2	16.9	37.1	49.1	8.7	3.15
YOLOv8n-5 (Sampled Shape 1)	49.5	37.6	34.9	16.8	38.2	50.2	8.4	2.94
YOLOv8n-5 (Sampled Shape 2)	49.6	37.8	34.9	15.9	38.4	50.1	8.4	2.94
YOLOv8n-5 (Sampled Shape 3)	49.6	38.1	35	16.6	38.2	50.9	8.4	2.94
YOLOv8n-6 (Sampled Shape 1)	50.1	38.3	35.3	16.6	38.6	51.1	8.6	3.01
YOLOv8n-6 (Sampled Shape 2)	50.2	38.2	35.4	16.6	38.3	51.3	8.6	3.01

Table 6. Based on COCO2017 and YOLOv8n, we explore the different size of AKConv with different initial sampled shapes. The ”Sampled Shape i” denotes different initial sampled shapes of AKConv.

Models	Initial Shape	Precison	Recall	mAP50(%)	mAP(%)
YOLOv5n	a	39.5	27.9	26.9	13.7
	b	39.4	28.2	26.8	13.6
	c	37.4	27.8	26.1	13.4
	d	37.5	27	25.5	12.9
	e	38.4	27.6	26.4	13.4

Table 7. It is shown that different initial sampled shapes of AK-Conv obtain the performance of YOLOv5n on VisDrone-DET2021.

feature information as AKConv does. Therefore with the addition of AKConv, the YOLOv8 boost is not as significant as the YOLOv5. Furthermore, at the same size, we test the effect of different initial sample shapes on network performance in COCO2017. It is obvious that under different initial samples, the fluctuation of the detection accuracy obtained by the network is not large. It benefits from the fact that the massive data of COCO2017 can flexibly adjust the offset. But, it does not mean that the network obtains detection accuracy that are not significantly different at all initial sampling coordinates. To explore again the effect of AKConv with different initial shapes on the network, we explore AKConv with size 5 and with different initial samples for experiments based on YOLOv5n on VisDrone-DET2021. It can be seen in Table 7 that the network obtains different detection accuracy with different initial samples. Therefore, AKConv with different initial sampling shapes has an impact on the performance of the network. Moreover, for specific networks and datasets, it is important to explore AKConv with appropriate initial sampling shapes to improve network performance.

5 Analysis and discussions

We initially AKConv of size 5 at different sampling positions in the previous experiment to observe the detection performance of YOLOv5n. It can be clearly noticed that the network behaves differently under different initial sampling shapes. It suggests that the adjustment ability of offsets is also limited. To measure the change in offset at each given position, we give the definition of the Average Offset, which is defined as follows:

$$AO = (\sum_{i}^{2N} |Offset_i|)/(2N) \quad (3)$$

AO (Average Offset) measures an average degree of change in the sampled points at each position by summing the offsets, and then taking the average. To observe the change of offsets, we selected the trained network and chose the last layer of AKConv to analyze the overall change trend of offsets. For the analysis, we randomly selected four images in VisDrone-DET2021 and then visualized the AKConv of size 5, which is initial for different sampling positions. As shown in Figure 5, we visualized the degree of change AO of offset at each sampling location. The different colors in Figure 5 represent the change in offsets at each sample position for different initial samples after training. The color of the line corresponds to the initial sampling shape in the middle. The different initial sample shapes in Figure 5 correspond to the initial sample shapes in Table 7. It can be concluded that OA changes less for the blue and red initial sample shapes in Figure 5. It means the red and blue initial samples are more suitable for this dataset than the other initial samples. As in the experiment in Table 7, it can be seen that the initial sampling shapes corresponding to blue and red obtained better detection accuracy. All the experiments proved that AKConv is able to bring significant performance improvement to the network. Unlike Deformable Conv, AKConv has the flexibility to scale network performance based on size. In all the experiments, we explore AKConv with size 5 extensively. Because when training COCO2017 with a large amount of data, we found that when setting the size of AKConv to 5, the training speed is not much different from the original model. Moreover, as the size of AKConv increases, the training time

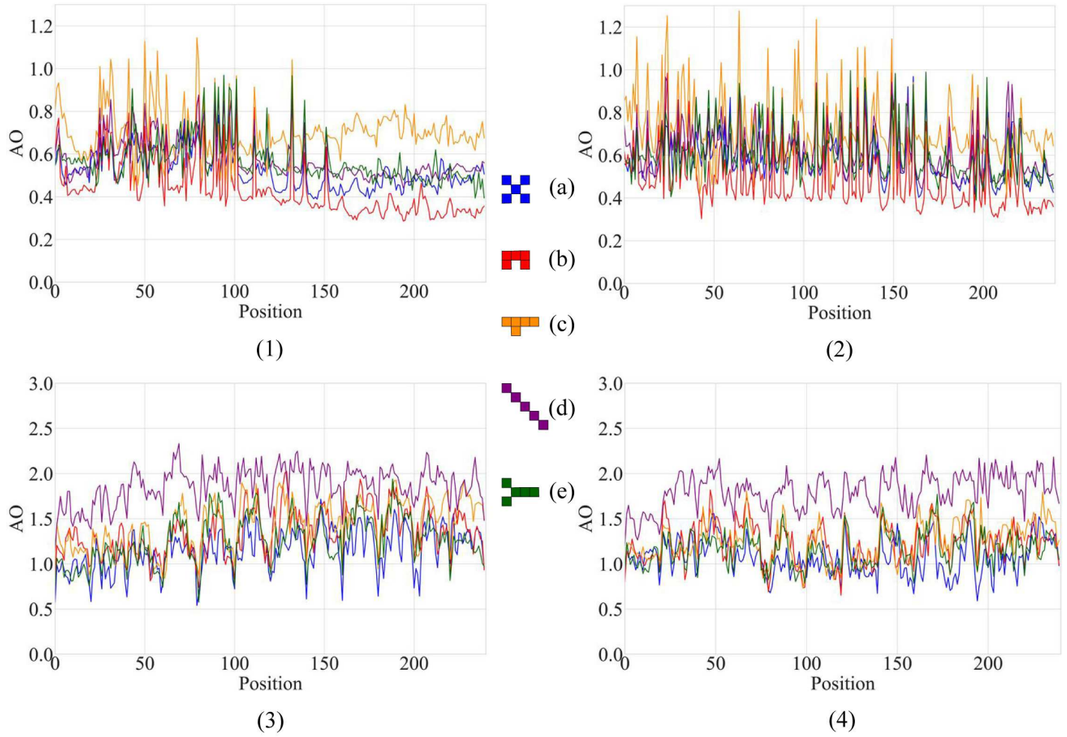

图5。它显示了AKConv的O，对于不同大小5的初始样本形状。它可以实现任意的采样形状通过为AKConv设计不同的初始采样 形状。

Fig. 5. It shows the variation of AO of AKConv for different initial sample shapes of size 5. It can achieve arbitrary sampling shapes by designing different initial sampling shapes for AKConv.

逐渐增加。在COCO2017、VOC7+12和VisDrone-DET2021的实验中，大小设置为5的AKConv为网络取得了良好的结果。当然，探索其他大小的AKConv也是可能的，因为参数数量呈线性增长和任意采样形状为AKConv的探索提供了丰富的选择。AKConv可以实现任意大小和任意样本的卷积操作，并可以通过偏移量自动调整样本形状以适应目标变化。所有实验表明，AKConv提高了网络性能，并为网络开销和性能之间的权衡提供了更丰富的选择。

6 结论

很明显，在现实生活中以及计算机视觉领域，物体的形状表现出各种变化。卷积操作的固定样本形状无法适应这些变化。尽管可变形卷积可以通过偏移量的调整灵活地改变卷积的样本形状，但它仍然存在局限性。因此，我们提出了AKConv，它真正实现了允许卷积具有任意样本形状和大小，这为卷积核的选择提供了多样性。此外，对于不同的领域，我们可以设计特定的采样坐标初始形状以满足实际需求。虽然在本论文中，我们仅设计了大小5的AKConv的多种采样坐标形状。然而，AKConv的灵活性在于它可以针对任何大小的采样核提取信息。因此，在未来，我们希望探索AKConv与...为特定任务领域提供任意形状的样本和任意数量的参数，这将推动后续任务的发展。

gradually increases. In the experiments of COCO2017, VOC 7+12, and VisDrone-DET2021, AKConv with size set to 5 gave good results for the network. Of course, the exploration of AKConv for other sizes is possible because the number of parameters that show linear growth and arbitrary sampling shapes bring a wealth of choices for the exploration of AKConv. AKConv can realize convolution operation with arbitrary size and arbitrary samples, and can automatically adjust the sample shape to adapt to the target change by offsets. All experiments demonstrate that AKConv improves network performance and provides richer options for the trade-off between network overhead and performance.

6 Conclusion

It is obvious that in real life as well as in the field of computer vision, the shapes of objects show various variations. The fixed sample shape of convolutional operation cannot adapt to such changes. Although Deformable Conv can flexibly change the sample shape of convolution with the adjustment of offset, it still has limitations. Therefore, we propose AKConv, which truly realizes to allow convolution to have arbitrary sample shapes and sizes, which provides diversity in the choice of convolution kernels. Moreover, for different domains, we can design specific initial shapes of sampling coordinates to meet the real needs. Although in this paper, we have designed multiple shapes of sampling coordinates only for AKConv of size 5. However, the flexibility of AKConv is that it can target any size of sampling kernel to extract information. Therefore, in the future, we would like to explore AKConv with appropriate sizes and sample shapes for specific tasks in the field, which will add momentum to the subsequent tasks.

参考文献

[1] K. He, X. Zhang, S. Ren, J. Sun, 深度残差学习用于图像识别，发表于：IEEE 计算机视觉与模式识别会议论文集，2016，第 770–778 页。 1[2] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, 密集连接卷积网络，发表于：IEEE 计算机视觉与模式识别会议论文集，2017，第 4700–4708 页。 1[3] J. Redmon, S. Divvala, R. Girshick,A. Farhadi, 一次仅看：统一、实时目标检测，发表于：IEEE 计算机视觉与模式识别会议论文集，2016，第 779–788 页。 1[4]C.-M. Chang, Y.-D. Liou, Y.-C. Huang, S.-E. Shen, P. Yu, T. Chuang, S.-J. Chiou, 基于YOLO的深度学习在针型仪表盘识别中的应用，发表于：测量与控制，2022，第 567–582 页。 1[5]Y. Xie, J. Zhang, C. Shen, Y. Xia, Cotr: 高效连接CNN和Transformer用于3D医学图像分割，发表于：Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, 2021年9月27–10月1日，第III部分，Springer，2021，第 171–180 页。 1[6] M. Abbasi, A. Shahraki, A. Taherkordi, 用于网络流量监控和分析（ntma）的深度学习：一项调查，发表于：计算机通信，2021，第 19–41 页。 1[7] H.-w. An, N. Moon, 基于 CNN-LSTM 的情感分析推荐系统设计，发表于：环境智能与人本计算期刊，2022，第 1–11 页。 1[8] J. Qin, W. Pan, X. Xiang, Y. Tan, G. Hou, 基于改进CNN的生物学图像分类方法，发表于：生态信息学，2020，第 101093 页。 1[9]X. Wang, N. He, C. Hong, Q. Wang, M. Chen, 基于改进YOLOX-X的无人机航拍目标检测算法，发表于：图像与视觉计算，2023，第 104697 页。 1[10] E. Yang, W. Zhou, X. Qian, J. Lei, L. Yu, Drnet: 带边界推理的双阶段精炼网络用于室内场景RGB-D语义分割，发表于：人工智能工程应用，2023，第 106729 页。 1[11] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, 可变形卷积网络，发表于：IEEE 国际计算机视觉会议论文集，2017，第 764–773 页。 1, 2, 3, 4, 6[12] X. Zhu, H. Hu, S. Lin, J. Dai, 可变形卷积网络v2：更可变形，更好结果，发表于：IEEE/CVF计算机视觉与模式识别会议论文集，2019，第 9308–9316 页。 1[13]Y. Zhao, L. Zhao, Z. Liu, D. Hu, G. Kuang, L. Liu, 用于合成孔径雷达图像中飞机检测的关注特征精炼和对齐网络，arXiv 预印本 arXiv:2201.07124，2022年。 1[14] T. Song, X. Zhang, D. Yang, Y. Ye, C. Liu, J. Zhou, Y. Song, 基于感受野特征增强卷积和三维注意力用于无人机拍摄图像的轻量级检测网络，发表于：图像与视觉计算，2023，第 104855 页。 1

[15] S. Huang, Z. Lu, R. Cheng, C. He, Fapn: 特征对齐金字塔网络用于密集图像预测，发表于：IEEE/CVF国际计算机视觉会议论文集，2021，第864–873页。 1[16] S. Zhao, S. Zhang, J. Lu, H. Wang, Y. Feng, C. Shi, D. Li, R. Zhao, 基于可变形卷积和YOLOv4的轻量级死鱼检测方法，Computers and Electronics in Agriculture 198 (2022) 107098。 1[17]A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, YOLOv4: 对象检测的最佳速度和精度，arXiv预印本 arXiv:2004.10934 (2020)。 1[18]W. Yang, J. Wu, J. Zhang, K. Gao, R. Du, Z. Wu, E. Firkat, D. Li, 可变形卷积和坐标注意力用于快速牛检测，Computers and Electronics in Agriculture 211 (2023) 108006。 1[19]J. Glenn, Ultralytics YOLOv8, https://github.com/ultralytics/ultralytics (2023)。 1, 4 [20] D. Li, Y. Li, H. Sun, L. Yu, 基于多尺度可变形卷积的深度图像压缩，Journal of Visual Communication and Image Representation 87 (2022) 103573。 1[21] T. Dumas, A. Roumy, C. Guillemot, 基于上下文自适应神经网络的图像压缩预测，IEEE Transactions on Image Processing 29 (2019) 679– 693。 1[22] J. Ball´e, D. Minnen, S. Singh, S. J. Hwang, N. Johnston, 带尺度超先验的变分图像压缩，arXiv预印本 arXiv:1802.01436 (2018)。 1[23]M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, pascal视觉对象类别挑战：回顾，International journal of computer vision 111 (2015) 98–136。 2[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, C. L. Zitnick, Microsoft COCO：上下文中的常见对象，发表于：计算机视觉–ECCV 2014：第13届欧洲会议，瑞士苏黎世，2014年9月6-12日，论文集，第V部分 13，Springer，2014，第740–755页。 2[25] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, H. Ling, 检测与跟踪遇见无人机挑战，IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (11) (2021) 7380–7399。 2[26] D. Li, J. Hu, C. Wang, X. Li, Q. She, L. Zhu, T. Zhang, Q. Chen, 卷积：逆转卷积在视觉识别中的固有性，发表于：IEEE/CVF计算机视觉与模式识别会议论文集，2021，第12321–12330页。 2[27] Y. Qi, Y. He, X. Qi, Y. Zhang, G. Yang, 基于拓扑几何约束的动态蛇形卷积用于管状结构分割，发表于：IEEE/CVF国际计算机视觉会议论文集，2023，第6070–6079页。 2, 4, 6[28] X. Zhang, C. Liu, D. Yang, T. Song, Y. Ye, K. Li, Y. Song, RFAConv：创新空间注意力和标准卷积操作，arXiv预印本 arXiv:2304.03198 (2023)。 2, 3, 4, 5[29] S. Woo, J. Park, J.-Y. Lee, I. S. Kweon, CBAM：卷积块注意力模块，发表于：

Bibliography

[1] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778. 1

[2] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708. 1

[3] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788. 1

[4] C.-M. Chang, Y.-D. Liou, Y.-C. Huang, S.-E. Shen, P. Yu, T. Chuang, S.-J. Chiou, Yolo based deep learning on needle-type dashboard recognition for autopilot maneuvering system, Measurement and Control 55 (7-8) (2022) 567–582. 1

[5] Y. Xie, J. Zhang, C. Shen, Y. Xia, Cotr: Efficiently bridging cnn and transformer for 3d medical image segmentation, in: Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III 24, Springer, 2021, pp. 171–180. 1

[6] M. Abbasi, A. Shahraki, A. Taherkordi, Deep learning for network traffic monitoring and analysis (ntma): A survey, Computer Communications 170 (2021) 19–41. 1

[7] H.-w. An, N. Moon, Design of recommendation system for tourist spot using sentiment analysis based on cnn-lstm, Journal of Ambient Intelligence and Humanized Computing (2022) 1–11. 1

[8] J. Qin, W. Pan, X. Xiang, Y. Tan, G. Hou, A biological image classification method based on improved cnn, Ecological Informatics 58 (2020) 101093. 1

[9] X. Wang, N. He, C. Hong, Q. Wang, M. Chen, Improved yolox-x based uav aerial photography object detection algorithm, Image and Vision Computing 135 (2023) 104697. 1

[10] E. Yang, W. Zhou, X. Qian, J. Lei, L. Yu, Drnet: Dual-stage refinement network with boundary inference for rgb-d semantic segmentation of indoor scenes, Engineering Applications of Artificial Intelligence 125 (2023) 106729. 1

[11] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 764–773. 1, 2, 3, 4, 6

[12] X. Zhu, H. Hu, S. Lin, J. Dai, Deformable convnets v2: More deformable, better results, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 9308–9316. 1

[13] Y. Zhao, L. Zhao, Z. Liu, D. Hu, G. Kuang, L. Liu, Attentional feature refinement and alignment network for aircraft detection in sar imagery, arXiv preprint arXiv:2201.07124 (2022). 1

[14] T. Song, X. Zhang, D. Yang, Y. Ye, C. Liu, J. Zhou, Y. Song, Lightweight detection network based on receptive-field feature enhancement convolution and three dimensions attention for images captured by uavs, Image and Vision Computing (2023) 104855. 1

[15] S. Huang, Z. Lu, R. Cheng, C. He, Fapn: Feature-aligned pyramid network for dense image prediction, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 864–873. 1

[16] S. Zhao, S. Zhang, J. Lu, H. Wang, Y. Feng, C. Shi, D. Li, R. Zhao, A lightweight dead fish detection method based on deformable convolution and yolov4, Computers and Electronics in Agriculture 198 (2022) 107098. 1

[17] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, Yolov4: Optimal speed and accuracy of object detection, arXiv preprint arXiv:2004.10934 (2020). 1

[18] W. Yang, J. Wu, J. Zhang, K. Gao, R. Du, Z. Wu, E. Firkat, D. Li, Deformable convolution and coordinate attention for fast cattle detection, Computers and Electronics in Agriculture 211 (2023) 108006. 1

[19] J. Glenn, Ultralytics yolov8, https://github.com/ultralytics/ultralytics (2023). 1, 4

[20] D. Li, Y. Li, H. Sun, L. Yu, Deep image compression based on multi-scale deformable convolution, Journal of Visual Communication and Image Representation 87 (2022) 103573. 1

[21] T. Dumas, A. Roumy, C. Guillemot, Context-adaptive neural network-based prediction for image compression, IEEE Transactions on Image Processing 29 (2019) 679–693. 1

[22] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, N. Johnston, Variational image compression with a scale hyperprior, arXiv preprint arXiv:1802.01436 (2018). 1

[23] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, International journal of computer vision 111 (2015) 98–136. 2

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755. 2

[25] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, H. Ling, Detection and tracking meet drones challenge, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (11) (2021) 7380–7399. 2

[26] D. Li, J. Hu, C. Wang, X. Li, Q. She, L. Zhu, T. Zhang, Q. Chen, Involution: Inverting the inherence of convolution for visual recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12321–12330. 2

[27] Y. Qi, Y. He, X. Qi, Y. Zhang, G. Yang, Dynamic snake convolution based on topological geometric constraints for tubular structure segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 6070–6079. 2, 4, 6

[28] X. Zhang, C. Liu, D. Yang, T. Song, Y. Ye, K. Li, Y. Song, Rfaconv: Innovating spatital attention and standard convolutional operation, arXiv preprint arXiv:2304.03198 (2023). 2, 3, 4, 5

[29] S. Woo, J. Park, J.-Y. Lee, I. S. Kweon, Cbam: Convolutional block attention module, in: Proceedings of the

欧洲计算机视觉会议 (ECCV), 2018, 第3–19页。 2[30] Q. Hou, D. Zhou, J. Feng, 坐标注意力用于高效移动网络设计，发表于：IEEE/CVF计算机视觉与模式识别会议论文集，2021，第13713–13722页。 2[31] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, Z. Liu, 动态卷积：卷积核上的注意力，发表于：IEEE/CVF计算机视觉与模式识别会议论文集，2020，第11030– 11039页。 2[32] M. Tan, Q. V. Le, MixConv：混合深度卷积核，arXiv预印本 arXiv:1907.09595 (2019).2[33]Q. Zhao, C. Zhu, F. Dai, Y. Ma, G. Jin, Y. Zhang, 用于球形图像的失真感知卷积神经网络，发表于：IJCAI，2018，第1198–1204页。 2[34]B. Coors, A. P. Condurache, A. Geiger, Spherenet：学习球形表示用于全向图像中的检测和分类，发表于：欧洲计算机视觉会议 (ECCV), 2018，第518–533页。 2[35]J. Glenn, YOLOv5版本v6.1,

https://github.com/ultralytics/yolov5/releases/tag/v6.1(2022)。 4[36]C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, YOLOv7：可训练的免费集合设定实时对象检测的新状态，发表于：IEEE/CVF计算机视觉与模式识别会议论文集，2023，第7464–7475页。 4

European conference on computer vision (ECCV), 2018, pp. 3–19. 2

[30] Q. Hou, D. Zhou, J. Feng, Coordinate attention for efficient mobile network design, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 13713–13722. 2

[31] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, Z. Liu, Dynamic convolution: Attention over convolution kernels, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11030–11039. 2

[32] M. Tan, Q. V. Le, Mixconv: Mixed depthwise convolutional kernels, arXiv preprint arXiv:1907.09595 (2019). 2

[33] Q. Zhao, C. Zhu, F. Dai, Y. Ma, G. Jin, Y. Zhang, Distortion-aware cnns for spherical images., in: IJCAI, 2018, pp. 1198–1204. 2

[34] B. Coors, A. P. Condurache, A. Geiger, Spherenet: Learning spherical representations for detection and classification in omnidirectional images, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 518–533. 2

[35] J. Glenn, Yolov5 release v6.1, https://github.com/ultralytics/yolov5/releases/tag/v6.1 (2022). 4

[36] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 7464–7475. 4