doi: 10.7690/bgzdh.2024.01.016

无人车目标识别主干网络技术特点对比分析

樊胜利，张玉芝，毕晓慧
（河北工业职业技术大学汽车工程系，石家庄 050019）

摘要：目标识别是无人车自动驾驶视觉感知模块的核心技术之一。当前，目标识别主要依靠主干网络提取特征，进而对目标进行分类与回归。通常情况下，无人车嵌入式计算平台的计算与存储能力有限，为了降低主干网络的算力与存储量，提升无人车的计算速度与效率，对目标分类任务的主干网络进行综合比较分析。围绕卷积核、感受野、池化层、全连接层、激活函数等，以 cifar10 和 cifar100 为实验数据，从理论分析与数据实践层面，对主干网络算子的选择与网络搭建进行分析对比，总结、归纳特征提取主干网络搭建的主要思路与做法。结果表明，该分析结论对目标分类主干网络在嵌入式无人车系统中的应用具有一定的理论指导与参考价值。

关键词：无人车；目标检测与分类；人工智能；卷积神经网络
中图分类号：TP183　　**文献标志码：**A

Comparison and Analysis of Technical Characteristics of Backbone Network for Target Recognition of Unmanned Vehicle

Fan Shengli, Zhang Yuzhi, Bi Xiaohui
(*Department of Automotive Engineering*, *Hebei Vocational University of Industry and Technology*, *Shijiazhuang* 050019, *China*)

Abstract: Target recognition is one of the core technologies of the visual perception module of unmanned vehicle automatic driving. At present, target recognition mainly relies on the backbone network to extract features, and then classify and regress the target. In general, the computing and storage capacity of embedded computing platform for unmanned vehicles is limited. In order to reduce the computing power and storage capacity of the backbone network and improve the computing speed and efficiency of unmanned vehicles, this paper makes a comprehensive comparison and analysis of the backbone network for target classification tasks. Focusing on convolution kernel, receptive field, pooling layer, fully connected layer and activation function, taking cifar10 and cifar100 as experimental data, the selection of backbone network operators and network construction are analyzed and compared from the theoretical analysis and data practice level, and the main ideas and practices of feature extraction backbone network construction are summarized and summarized. The results show that the analysis conclusion has a certain theoretical guidance and reference value for the application of the target classification backbone network in the embedded unmanned vehicle system.

Keywords: unmanned vehicle; target detection and classification; artificial intelligence; convolutional neural network

0　引言

基于视觉的 deep-learning 是无人车自动驾驶技术的重要组成部分。它不仅是陆战场无人车辆侦察，同时也是无人车发现目标、瞄准与实施火力打击的核心与关键技术之一。当前，针对无人战车所处的复杂环境，对于各种车辆、作战人员和飞行器等目标的检测与识别，在技术路线上主要包括基于 backbone 的集中式框架与分布式框架 2 种类型。

基于 backbone 的集中式分类任务框架[1-3]，每个 Neck 和 Head 与各自的分类任务紧耦合，所有分类任务共享 backbone，参数量相对较小，适用于内存与计算性能受限的嵌入式平台。不足之处在于：1) 在共享 backbone 的技术框架下各种参数与超参数在训练时，很难对所有训练模型都达到最优化；2) 只要有一个 head 或 neck 被替换，主干 backbone 就需要重新训练。这说明这类技术方案的耦合性较强，难以即插即用，随机替换。而基于 backbone 的分布式分类任务框架[4-7]，优点是模型可以随时替换，缺点则是参数量比较大，对嵌入式平台的内存与计算性能要求比较高。不难看出，两者都有各自的优缺点。然而，无论哪一种技术框架，backbone 的选择都是训练模型或推理模型进行特征提取的关键。鉴于此，笔者对当前流行的 backbone 框架的优缺点进行了总结与归纳，为自动驾驶中 deep-learning 技术框架的选择与确定提供理论参考与依据。

收稿日期：2023-09-08；**修回日期：**2023-10-15
第一作者：樊胜利(1976—)，男，河北人，博士。
通信作者：张玉芝(1978—)，女，河北人，硕士。

1　目标分类网络 backbone 分析

1.1　Lenet-5[8-11]的 backbone 分析

1.1.1　模型分析

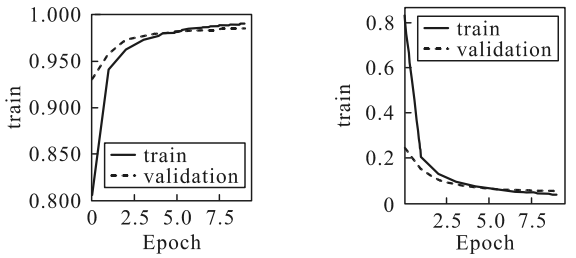
传统的基于文本分类网络存在 3 个问题：1) 特征提取器和分类器强依赖于先验知识和特定任务，只能局限于低维空间分类，分类网络的适应性不高；2) 全连接网络包含上千万权重参数，系统的消耗与内存占用很大，影响了网络应用的实时性；3) 全连接网络忽略了图像的 2 维局部结构。针对上述问题，以 LeCun 为代表的研究学者结合梯度学习、反向传播、损失函数，并考虑到图像的拓扑结构，利用权值共享、局部连接的思想，提出了基于卷积神经网络的 Lenet-5 模型其结构分析如表 1 所示。

表 1　Lenet-5 网络结构分析

Index	C1	S2	C3	S4	C5	FC6	FC7
Input Size	32*32	28*28	14*14	10*10	5*5	120*1	84*1
Feature Map Size	28*28	14*14	10*10	5*5	1*1	84*1	10*1
Neutrons	28*28*6	14*14*6	10*10*16	5*5*16	1*1*120	84	10
Parameters	156	12	1 516	32	48 120	10 164	850
Connections	122 304	5 880	15 100	2 000	48 120	10 164	850
Kernel Size	5*5	2*2	5*5	2*2	5*5		
Stride	1	2	1	2	1		
Receptive Field	5	8	12	15	19		
Flops	235 200	37 632	80 000	12 800	96 000	20 160	1 680

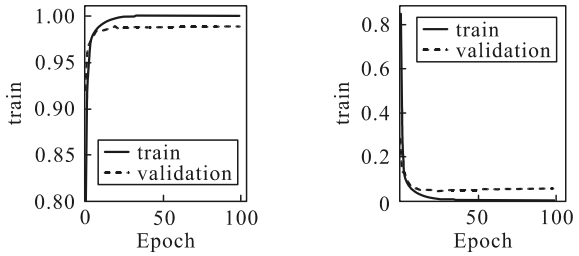
从上表可知，Lenet-5 是一个包含卷积层、池化层和全连接层，运用卷积神经网络的感受野、权重共享、局部连接特点的 7 层网络。它的池化层主要采用的是 Max Pooling，激活函数为 sigmoid。Lenet-5 网络架构的提出，为后续分类网络的迭代与演进奠定了基础。例如，在 Lenet-5 的 C3 卷积层，为了减小训练参数和提取 C2 特征图的多种组合形式，它把 C2 所有的特征图分成 4 种组合模式，在 C3 层构成了 16 个特征图。这不仅是 Lenet-5 分类网络的特点之一，这一思路在后续的 AlexNet 网络结构中得到了进一步的延续与推广，继而推出了 Group Convolution 和 Depth-wise Convolution 等概念。

再如 Lenet-5 的池化层，池化层的操作主要包括 Average Pooling 和 Max Pooling[12]，这 2 个操作的共同特点不仅可以通过下采样，去除冗余信息，减少参数，控制过拟合与提高模型性能，而且可以保证平移不变性。二者区别在于 Average Pooling 可以降低因邻域大小受限而造成估计方差增大的问题，Max Pooling 则可降低减小因卷积层参数误差造成的估计值偏移误差的问题。然而，在实际的神经网络结构设计中，Max Pooling 更为着重局部语义，即细节信息，如纹理或边缘特征等信息，主要应用在深度网络的前端(具体训练与验证结果如图 1 和 2 所示)；Average Pooling 则更为着重全局语义，更多保留的是背景信息，主要应用在深度网络的后端(具体训练与验证结果如图 3 和 4 所示)。而 Letnet-5 网络的结构设计则主要采用的是 Max Pooling，在网络结构较浅的情况下，比较容易丢失细节信息。这也是后续分类网络继续优化演进的方向之一。



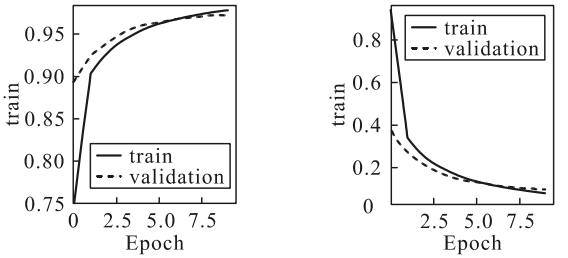
(a) 基于最大池化层的训练精度　(b) 基于最大池化层的损失函数

图 1　epoch=10 条件下基于 Max Pooling 的测试与验证指标



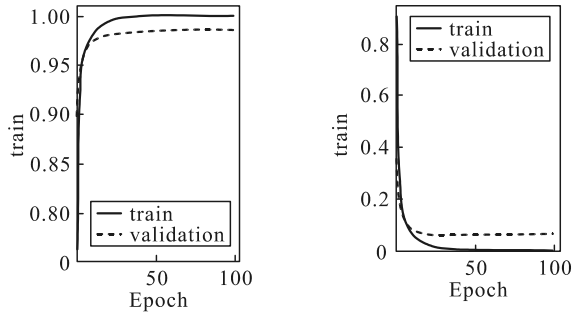
(a) 基于最大池化层的训练精度　(b) 基于最大池化层的损失函数

图 2　epoch=100 条件下基于 Max Pooling 的测试与验证指标



(a) 基于平均池化层的训练精度　(b) 基于平均池化层的损失函数

图 3　epoch=10 条件下基于 Average Pooling 测试与验证指标



(a) 基于平均池化层的训练精度　(b) 基于平均池化层的损失函数

图 4　epoch=100 条件下基于 Average Pooling 测试与验证指标

1.1.2　实验数据分析

不同 epoch 条件下 Max Pooling 与 Average Pooling 测试和验证指标对比如表 2 所示。

根据图 1—4 以及表 2 的对比可以看出，在 epoch=10 的条件下，Max Pooling 相较于 Average Pooling 而言，Accuracy 提高了 1.133%，Val_accuracy 提高了 1.131%，Loss 提高了 56.54%，Val_loss 提高了 51.47%；在 epoch=100 的条件下，Max Pooling 相较于 Average Pooling 而言，Val_accuracy 提高了 0.375%，Loss 提高了 52%，Val_loss 提高了 22.8%。不难看出：在网络深度比较浅的条件下，Max Pooling 比 Average Pooling 对于分类网络性能指标的提升更为显著。

从表 3 的对比分析可以看出：sigmoid 激活函数由于会出现“梯度消失”现象，所以无论从 training 还是 Validation，它的 Accuracy、Loss、Val_accuracy、Val_loss 与 Relu 激活函数相比，都有不同程度的下降。可以看出，正是由于“梯度消失”问题，才会造成 training 与 validation 的曲线拟合度貌似较好，并没有出现“欠拟合”或者“过拟合”，然而，这只是一种“假象”。与此形成对比的是，基于 Relu 激活函数由于它的梯度不存在“消失”或者爆炸的现象，所以 Accuracy、Loss、Val_accuracy、Val_loss 的评价指标会比较好；但是，当 epoch=100 时，由于训练样本不足，会出现“过拟合”现象，需要通过补充训练数据解决。

表 2　不同 epoch 条件下 Max Pooling 与 Average Pooling 测试和验证指标对比

Index	epoch	Accuracy(max)	Val_accuracy(max)	Loss(max)	Val_loss(max)
Max pooling	10	0.990 4	0.986 2	0.033 2	0.044 5
	100	1.000 0	0.988 7	7.117 9e−05	0.052 8
Average Pooling	10	0.977 4	0.973 4	0.076 4	0.091 7
	100	1.000 0	0.985 0	1.484 9e−04	0.068 4

表 3　不同 epoch 条件下激活函数 sigmoid 与 Relu 测试和验证指标对比

Index	epoch	Accuracy(max)	Val_accuracy(max)	Loss(max)	Val_loss(max)
Relu	50	1.000 0	0.987 9	9.199 8e−04	0.057 0
	100	1.000 0	0.986 7	1.023 0e−04	0.079 3
Sigmoid	50	0.984 6	0.983 7	0.045 8	0.053 1
	100	0.987 6	0.987 3	0.036 6	0.043 0

1.2　AlexNet 的 backbone 分析

1.2.1　理论分析

1.2.1.1　Relu 激活函数与 signoid 激活函数的分析

传统的 sigmoid 径向基激活函数在 0 附近有比较好的激活特性，然而它在正负饱和区或网络较深时容易导致梯度弥散或消失的现象。针对此问题，AlexNet[13-15]网络采用了 ReLU 激活函数。ReLU 激活函数的优点主要有 3 点：1) 在负半区的导数为零，即代表任何一个神经元在负半区的反馈都是零，表明该神经元不参与训练，具备稀疏性；2) 在正半区导数大于零，不会导致梯度消失；3) 与 signoid 激活函数相比，它的计算速度更快。

1.2.1.2　LRN 机制分析

ReLU 激活函数[16]在解决基于 sigmoid 径向基函数在网络较深时的梯度弥散或消失问题的同时，也带来了新的问题。例如，由于 ReLU 激活函数的值域为 $[0,+\infty)$，这会让训练数据或测试数据的数据分布各不相同，不仅增加了网络的学习难度，降低了网络的学习速度和效率，而且降低了网络的泛化能力。针对这一问题，AlexNet 网络提出了(Local Response Normalization，LRN)[17-19]机制。它对局部神经元的活动创建竞争机制，使得其中响应较大的神经元的权重变得相对更大，并抑制反馈较小的神经元的权重；因此，LRN 通过归一化训练数据与测试数据，不仅可以保持深度网络的学习速度，而且增强了网络的泛化能力。

AlexNet 深度网络有 3 个全连接层，它的参数较多，大约有 60 M 参数的增加加剧了模型的复杂性，在训练数据不充足的情况下，极容易造成复杂

模型与小数据之间的过拟合现象。针对这一问题，Alexnet 网络在全连接层采用了 Inverted Dropout 方法。通过 Bernoulli 概率模型在网络的全连接层设置神经元的删除概率，通过反复学习迭代，相当于生成了多个不同的深度学习网络，并对结果进行了加权平均。这样做有 2 方面优点：1) 可以使一些“相反的”过拟合效果互相抵消，在整体上降低了模型过拟合发生的概率；2) 减少神经元相互之间复杂的共适应性关联，增强网络的自适应性、泛化能力和鲁棒性。

1.2.2　实验数据分析

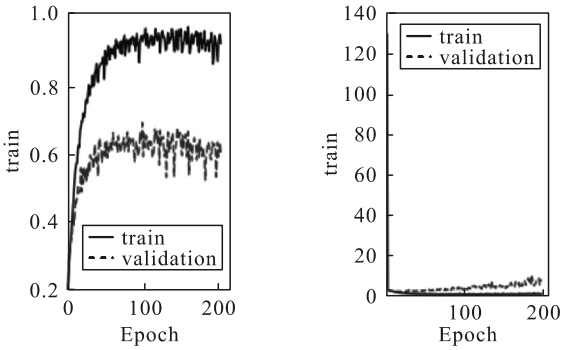
由上述分析可知，对于 AlexNet 网络，其特点可归纳为 4 点：1) 采用 GUP 加速；2) 使用 Relu 激活函数；3) 使用 LRN 响应归一化；4) 全连接层使用 Dropout 随机失活神经元。为此，文中的实验主要针对 LRN 与 Dropout 对 AlexNet 网络模型的作用进行分析与验证。

1.2.2.1　LRN 影响分析

$$b^i_{x,y}=a^i_{x,y}\Bigg/\left(k+\alpha\sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)}(a^j_{x,y})^2\right)^{\beta}。\qquad(1)$$

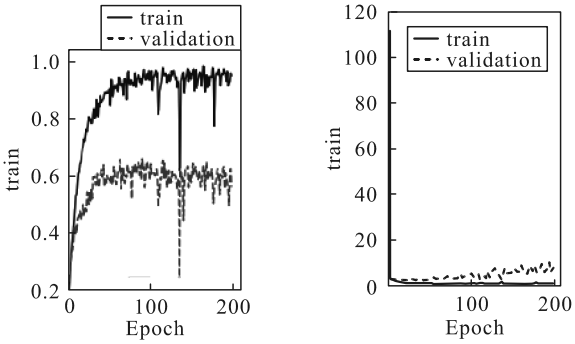
式中：i 为第 i 个卷积核；(x, y)为卷积核中神经元的位置；k, n, α, β 为超参数，一般情况下 k=2, n=5, α=e−4, β=0.75。

从图 5 和 6 的对比可以看出，使用 LRN 的 AlexNet 与未使用 LRN 的 AlexNet，无论是训练集还是验证集的 accuracy 与 loss，并没有显著提升。相反，对于使用 LRN 的 AlexNet，它的训练集与验证集的 accuracy 和 loss 反而会有 1～2 处 accuracy 的下降。因此，AlexNet 分类网络之后，LRN 逐步被 BN 或 Dropout 方法代替，并没有得到进一步的应用与推广。



(a) 未考虑 LRN 时的训练精度　　(b) 未考虑 LRN 时的损失函数

图 5　Epoch=10 时未使用 LRN 下的 accuracy 与 loss 对比

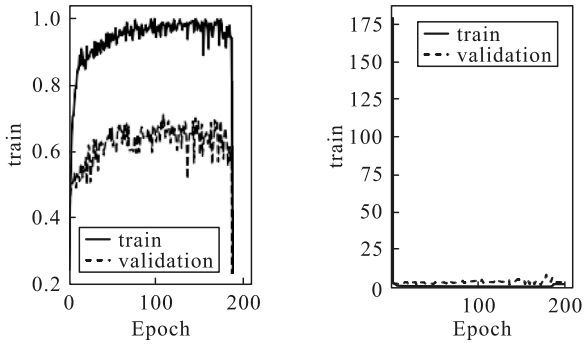


(a) 考虑 LRN 时的训练精度　　(b) 考虑 LRN 时的损失函数

图 6　Epoch=100 时使用 LRN 下的 accuracy 与 loss 对比

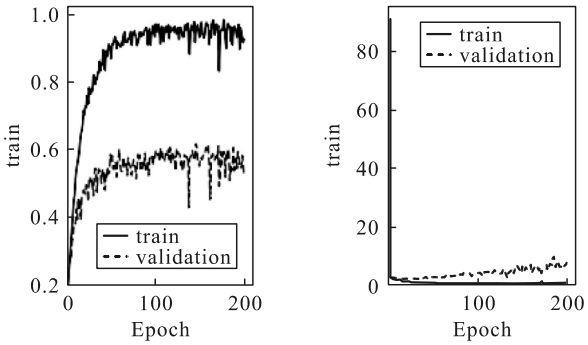
1.2.2.2　Dropout 影响分析

Dropout[20-21]是一种正则化方法。它通过对网络中神经元设置被丢弃概率的办法，以达到降低方差，实现正则化的目的。从图 7 和 8 的比较中可以看到：采用 Dropout 方法后对于训练集和测试集的 accuracy 和 loss 都有提升。



(a) 未考虑 Dropout 时的训练精度　　(b) 未考虑 Dropout 时的损失函数

图 7　未采用 Dropout 的 accuracy 与 loss 的比较



(a) 考虑 Dropout 时的训练精度　　(b) 考虑 Dropout 时的损失函数

图 8　采用 Dropout 的 accuracy 与 loss 的比较

1.3　VGGNet 的 backbone 分析

1.3.1　理论分析

VGG−16 与 VGG−19 对比分析如表 4 所示。

表 4 VGG-16 与 VGG-19 对比分析

指标	VGG-16	VGG-19
Input Size	224*224*3	224*224*3
Kernel Size	3*3 1*1 (stride = 1)	3*3
Receptive Field	44*44	50*50
Parameters/M	138	143
Connections/G	11	13.5
Flops/G	20.7	26.3
Pooling	max pooling (size = 2*2 stride = 2)	max pooling
Padding	same	same

VGGNet 一般分为 VGG-11、VGG-13、VGG-16 和 VGG-19 4 种网络结构[22-27]。当前，VGG-16 和 VGG-19 是在提取图像特征过程中，具备良好识别精度的代表性网络。其主要特点包括以下 2 方面：

1) 小卷积核，深层网络。

在 AlexNet 和 Lenet 网络模型中，卷积核尺寸大多为 5*5，网络的深度不超过 7 层。而在 VGG-16 和 VGG-19 深度学习网络模型中，在不改变感受野、不增加参数量的情况下，通过减小卷积核尺寸，既增加了网络深度，也增强了深度学习网络模型的抗过拟合能力，进而显著提升了分类准确度。例如，在 VGG-16 网络模型中用 2 个 3*3 卷积核和 3 个 3*3 卷积核分别替代了 5*5 和 7*7 的卷积核，感受野仍然保持 5 和 7，然而参数量则分别减小了 64% 和 42.9%。

2) 调参难度高，存储容量大，嵌入式系统应用受限。

VGG-16 和 VGG-19 模型分类精度的提升，是以扩展网络宽度，增加网络深度为代价的。这同时也表明网络参数调整的难度相比 Alexnet 和 Lenet 会显著增加。同时，从表 4 中也可以看出：VGG-16 和 VGG-19 的参数量分别约为 138 和 143 M，假设每个参数类型为 float32，则模型的参数文件大小则约为 552 和 572 MB，这对于嵌入式系统的应用会受到很大限制。

1.3.2 实验数据分析

VGG-16 和 VGG-19 是 VGGNet 分类网络的主要代表，其主要优点是具有良好的网络深度和感受野，特征图可以有效提取目标的全局和局部特征，进而达到比较高的目标识别精度。从表 5 可以看出：对于训练数据集的精度，VGG-16 和 VGG-19 区分不明显，但在验证数据集的精度上，VGG-16 的平均精度大约是 78.77%，而 VGG-19 则是 87.09%，与 VGG-16 相比，VGG-19 提高了将近 10%。含有全连接层的 VGG-19 相比 VGG-16，参数量却增加了 34.49%；因此，VGG-19 网络模型识别精度的提升是以增加网络的复杂度和存储空间为代价的。

表 5 VGG-16 与 VGG-19 典型指标对比

指标	VGG-16		VGG-19	
	With FC	Without FC	With FC	Without FC
mean loss	0.115 6	0.106 2	0.127 3	0.111 8
mean accuracy	0.960 4	0.962 8	0.955 7	0.960 8
mean val_accuracy	0.787 7	0.787 2	0.870 9	0.875 9
mean mae	4.417 9	4.426 4	0.012 0	0.010 8
mean mse	27.710 7	27.765 0	0.006 0	0.005 4
mean val_mae	4.428 7	4.394 2	0.028 3	0.027 2
mean val_mse	27.626 6	27.410 1	0.020 7	0.020 2
total parameters	33 638 218	14 719 818	45 239 370	20 044 874

从 VGG-16 和 VGG-19 的网络结构分析可以看出：全连接层的存在是上述 2 个网络参数量激增的主要因素之一。为此，文中尝试将这 2 个网络的全连接层去掉，并对精简后的网络与原网络进行对比。从表 5 可以看出：无论是 VGG-16，还是 VGG-19，当以同样的训练集数据和验证集数据进行对比时，无论是精度、损失还是平均绝对误差或平均均方误差，全连接层的存在与否对于上述指标并没有显著影响，但是参数量却降低了 56.24%和 55.69%。

1.4 GoogleNet 的 backbone 分析

1.4.1 理论分析

1) 卷积层的降维与卷积核分解。

GoogleNet[28]的 Inception 模块，从 V1 演变到 V4，卷积核的变化是它的一条技术主线。主要特点主要分为 2 方面：

1) 采用1×1卷积核进行降维，从而降低了计算复杂度，减少了计算量。如表 6 所示，在 Inception

V1 模块中，对于一组输入为192×32×32 、输出为256×32×32 的特征图，采用一个1×1卷积核与3×3卷积核串联的方式，与3×3卷积核相比，参数量减少了 49.55%，计算量降低了 45.83%。

表 6　不同卷积核与计算性能关联分析

Performance Index	Precondition	
	Input size:192×32×32	Output size:256×32×32
	3×3 kernel	1×1 and 3×3 kernel contamination
Parameters	442 624	223 298
Operations including multiplying and adding	905 969 664	490 773 568

2) 在 Inception V2 模块中，采用了与 V1 模块不同的另一种降维方法，把一个5×5卷积核串联为2 个3×3卷积核。可以看到，这 2 种方式的感受野都是一样的，均为 5。然而，后者却显著降低了参数量与计算量。例如参数量降低了 28%。因此，无论 V1 中1×1卷积核的应用，还是 V2 中对于5×5卷积核的分解，它的最终目的都是增加网络深度，提高网络的非线性拟合能力；同时，减小参数量与计算量，加速计算。

2) 多尺度并行计算卷积与重聚合。

GoogleNet 网络中的 Inception 模块[29−33]运用了多个尺度的卷积核和池化操作，例如 1*1, 3*3, 5*5, 1*3, 3*1 等，每个不同尺寸的卷积核会对不同尺寸上的特征图进行卷积或池化操作，最后再进行融合或重聚合。这种做法的优点不仅可以使特征的提取更为丰富，有利于提高分类的准确度；而且，当网络深度和宽度不断延展和拓宽时，特征图会逐步演变为稀疏矩阵。如果只用单一卷积核进行卷积计算，会把算力过多浪费在稀疏矩阵中的“0”值较多的区域，从而造成算力利用率下降。针对这种情况，利用多个不同尺度的卷积核提取特征，则相关性较强的特征比较容易融合在一起，单一尺度的稀疏矩阵就会分解为多个不同尺度的密集矩阵，从而进一步提高了算力的利用效率，加快了收敛速度。

深度学习网路为了加快收敛速度，提高泛化能力，一般要求输入数据服从独立同分布。然而，经过每层的卷积，非线性映射和池化操作，输入数据波动较大，比较容易产生“Internal Covariance Shift”的问题，会引发“梯度消失”或“梯度爆炸”，从而影响网络的收敛速度与泛化能力。针对该问题，Inception 模块引入了 Batch Normalization 的方法，在进入激活函数(例如 Relu 或 Sigmoid)之前，利用数据的均值与方差，将其“归一化”为标准正态分布。

1.4.2　实验数据分析

GoogleNet Inception_v1/v2/v3/v4 网络模型演进的主线之一是卷积的变化。例如，Inceptioin_v1 中的卷积是以 5×5 和 3×3 为主，在 Inception_v2 中则是利用 2 个 3×3 替代 5×5，Inception_v3 中则是利用 1×3 和 3×1 代替 3×3 卷积，Inception_v4 则是加入残差网络。因此，不难看出：卷积核的变化准则是在保持感受野不变的条件下，降低网络参数数量；同时，加深网络深度并拓展网络宽度。这种思想在增加网络的结构复杂度的同时，兼顾降低网络参数量与防止网络过拟合二者的统一。然而，模型精度的提升是以大的参数文件为代价的，这势必会影响模型在嵌入式平台的推广与应用。

1.5　Resnet 的 Backbone 分析

1.5.1　理论分析

通常情况下，人们认为：网络深度越深，训练集的 loss 会越来越小。例如，从 VGG−19 到 GoogleNet，网络深度从 19 层延伸到 22 层，loss 的确有了明显改善与提升。然而，按照这一思路逐渐加深网络时，训练集的 loss 反而不升反降；因此，为解决深度网络的退化问题，研究人员提出了 Resnet 深度网络，其主要特点包括 2 点：

1) 残差网络结构与捷径连接。

对于传统的深度网络，随着反向传播的多次迭代，梯度会越来越小，最终会趋于零，从而导致权重无法更新。而 Resnet 网络经过残差网路与捷径连接之后，输出的表达式为：

$$H(\boldsymbol{x})=F(\boldsymbol{x})+\boldsymbol{x}。\tag{2}$$

式中 $H(\boldsymbol{x})$表示残差网络的输出。

从上式不难看出，此时进行梯度反向传播时，梯度的表达式不再是 $F'(\boldsymbol{x})$，而是 $H'(\boldsymbol{x})=F'(\boldsymbol{x})+1$。可以看出，误差的梯度大于 1，进而会解决深度网络梯度“消失”的现象。

2) Building−block and Bottleneck。

Building−block 是 ResNet[34−36]网络结构中进行跳跃连接的基本模块。它的适用场景一般在深度有限的条件下，例如 ResNet34。而 Bottleneck 则针对深度更深的网络，例如 Resnet50 或 Resnet101。究

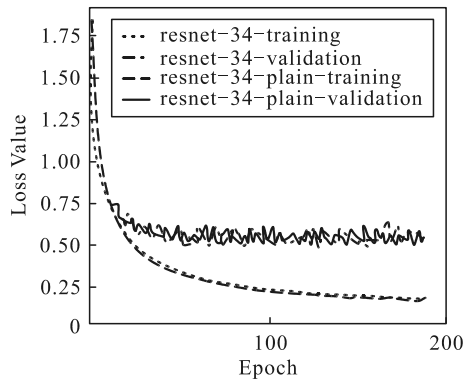
其原因在于它可以在增加网络深度的情况下尽量的减少参数量。例如，若不采用 Bottleneck 模块，当输维度是 256 时，参数量为 1 179 648。而如果采用 Bottleneck 模块的话，参数量为 69 632。后者比前者缩小了 94.1%，即占前者参数量的 5.9%；因此，Bottleneck 模块相较于 Building-block 更适用于深度网络的延展。

1.5.2 实验数据分析

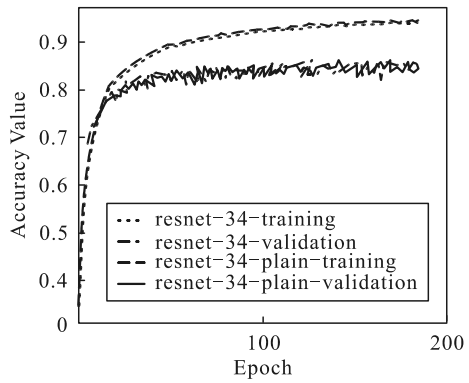
1.5.2.1 基于残差的 resnet 网络和普通 resnet 网络分析

从 1.5.1 节分析可知，为避免梯度“消失”现象，Resnet 网络引入了残差的概念。可以看出：有残差模块的 Resnet18 相较于没有残差模块的 Resnet18-plain[37]，无论是训练数据集的 loss 和 accuracy，还是验证数据集的 loss 和 accuracy，其性能指标都要优于后者。

从图 9 则看出：有残差模块的 Resnet34 与没有残差模块的 Resnet34-plain 相比，前者在训练数据集的性能明显弱于后者。然而，在验证数据集，二者的差别不大。这说明，带有残差模块的 Resnet 网络性能的提升也需要借助超参数的调整。否则，网络的深度优势不能充分发挥。



(a) 不同 Resnet-34 网络的损失函数

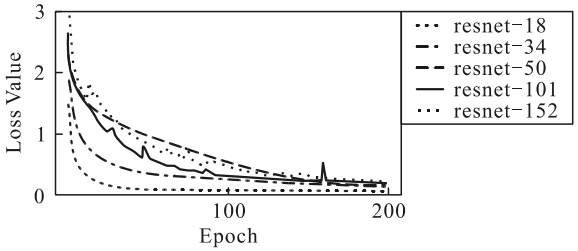


(b) 不同 Resnet-34 网络的训练精度

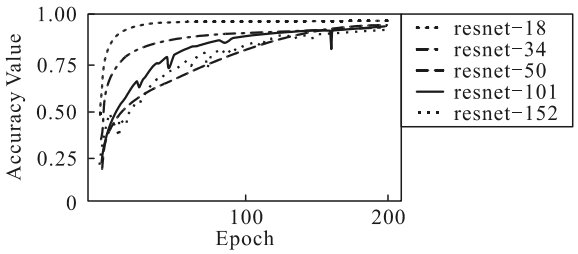
图 9 多种不同类型 resnet-34 网络的损失与训练精度对比

1.5.2.2 Resnet 网络对比分析

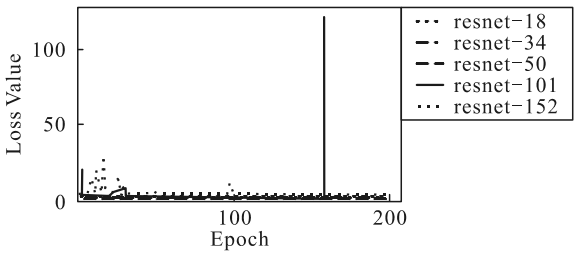
一般来讲，Resnet 网络的目标分类精度是随着网络深度不断加深而逐步提升的。例如在图 10 中 Resnet50 和 Resnet101，Resnet152 的比较中，可以看到：在训练阶段，Resnet50 的 loss 是低于 Resnet101 和 Resnet152 的，而 accuracy 则相差不多。而到了验证阶段，Resnet101 和 Resnet152 的 accuracy 则明显高于 Resnet50。在同等条件下，一般而言(一般指学习率和优化器等超参数的取值一样)，网络深度越深，则目标识别精度越高；但是，从 Resnet18 和 Resnet34 在图 9 中的训练与验证结果显示，可以看到：二者的 loss 与 accuracy 明显强于 Resnet50、Resnet101 和 Resnet152，这说明找到适应于网络的超参数对于性能的提升具有重要作用。



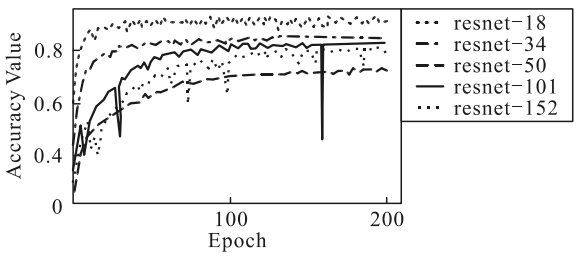
(a) 不同 Resnet 网络的训练损失函数



(b) 不同 Resnet 网络的训练精度



(c) 不同 Resnet 网络的验证损失函数



(d) 不同 Resnet 网络的验证精度

图 10 多种不同类型 resnet 网络的 loss 与 accuracy 对比

2　典型分类网络的结构比较与分析

2.1　典型分类网络的结构比较

在目标分类网络发展初期，受限于卷积核与全连接层的影响，模型的参数量和模型权重文件的大小，呈现大幅上升的现象。例如参数量由 Lenet-5 的 60 K 剧增至 VGG-16/19 的 138 M，增长幅度达到了 230 倍之多(如表 7 所示)。而模型文件从 Lenet-5 的 174 K 上升到 vgg-16/19 的 550 M，权重文件增长了约 5 000 倍。而在后续的网络演进中，无论是参数量还是模型权重文件，都有所降低。其主要原因在于以 Googlenet 和 Resnet 为代表的网络模型采取了多种方法：如减少全连接层的数目，在不影响感受野的情况下采用了参数量更少的 1×1、3×3 卷积核和 1×n 及 n×1 为代表的非对称卷积核，采用全局平均池化代替全连接层等。

表 7　典型分类网络对比分析

Network	Network Depth	Parameters/M	Con filter	Batch Normalization	Model File Size/M
Lenet-5	5	0.060	5×5	No	0.174
Alexnet	7	62.37	5×5/3×3	Yes	238
VGG-16/19	16/19	138	3×3	Yes	550
GoogleNetV1/V4	22	5	1×1/3×3/5×5	Yes	28～50
Resnet10-152	10～152	14.3～117.364	1×1/3×3	Yes	90

2.2　典型网络的仿真对比与分析

通过上述分析可知，当前面向分类任务的主干网络的技术演进是以网络深度、卷积核、池化操作、优化器、学习率和激活函数等为切入点。例如，从 Lenet 到 Resnet 网络可以看到：网络深度在不断延伸；在不改变感受野的前提下，采用更为丰富的卷积核；池化层操作从平均池化和全局池化逐步发展为全局平均池化和全局最大池化，以及混合池化和随机池化等。学习率的调整也从固定学习率演进为自适应学习率，以避免陷入局部最优。在采用 Cifar10 数据进行训练和验证时，无论是训练集的 loss 和 accuracy，还是验证集的 loss 和 accuracy，它们随着网络深度的不断加深，性能指标都得到了较大的改善。同时，由于 Googlenet 采用了 inception v4 的模块，引入了残差，所以在 Cifar10 的性能对比上并不逊色于 resnet50。然而，在采用 Cifar100 数据进行训练和验证时，同样的网络模型，性能的对比却出现了较大差异。例如，Lenet 和 Alexnet 模型在 Cifar10 数据上的 accuracy 可以达到 60%和 70%，但在 Cifar100 数据上，accuracy 只能达到 30%和 50%。即使对学习率进行了自适应调整，其他网络模型也呈现同样的效果。这说明，分类任务的不断丰富也对网络结构提出了更高的要求。

上述各种 backbone 特征提取网络，已广泛应用在无人车视觉感知模块中。其中，Lenet-5 与 Alexnet 网络深度较浅，特征提取能力有限，它尚不能适应复杂的城市环境，在无人车实际应用场景中多见于目标单一、简单、空旷的场景。例如无人车在飞机场的异物检测。VGG 模型对于频台的算力与存储能力要求较高，只能在一些高算力平台部署，例如英伟达的 Orin 以及地平线的 J5 等。目前，在无人车特征提取网络应用中较为常见的是 Resnet 模型，其对算力的要求较为适中，多见于高速、城市环境等复杂场景。

3　结论

从具有代表性的分类任务主干网络的发展中，不难看出：从 Lenet-5 到 Googlenet，主干网络的演进主要着眼于 2 个问题，一是防止梯度消失或梯度爆炸，例如从 sigmoid 激活函数到 Relu 激活函数；二是防止过拟合，例如 LRN、BN、重叠池化和 Dropout 等方法。它对于嵌入式环境下，由于硬件资源与运算速度受限，所面临的轻量级计算与运算需求并没有过多考虑；因此，分类网络在后续的演进中，主要着眼于拓展 Googlenet 的模块化设计理念与嵌入式环境的应用与推广的目的。同时，加深网络深度，减小网络规模与参数量。无论是 Resnet、Squeezenet 还是 Mobilenet、Shufflenet，都在卷积的选择上进行了重新考量与设计。例如，Resnet 中的残差模块，Squeezenet 中的 Fire Module，Mobilenet 中的 Depth-wise Convolution 与 Point-wise Convolution，Shufflenet 中的 Group Convolution 等。它们都是通过对卷积层的设计以达到拓展深度、降参数量的目的，进而更好地适应嵌入式平台的应用场景。对于 backbone 技术框架的选择，相比于分布式 backbone，集中式 backbone 对于算力与存储的要求不高，如无人车嵌入式计算平台算力有限，可选择此种部署策略。但是其缺点是多任务的各个模型不能“即插即用”，需要重新训练；相反，如果嵌入式计算平台的算力相对充足，可选择分布式

backbone 技术框架，各个任务模块可以灵活替换。未来，随着语义分割、实例分割、对抗网络、RNN 网络研究的不断深入与拓展，分类网络也必将呈现新的特点，具备更高的研究价值与更广的战场落地空间。

参考文献：

[1] OMAR E, YOUNES A, NOOR A, et al. Backbones-Review: Feature Extraction Networks for Deep Learning and Deep Reinforcement Learning Approaches[J]. ArXiv, 2022(6): 08016.

[2] PIETIKÄINEN M, SILVEN O. Challenges of Artifificial Intelligence–From Machine Learning and Computer Vision to Emotional Intelligence[J]. ArXiv: 2022(1): 01466.

[3] ZHOU J, LIU L, WEI W, et al. Network representation learning from preprocessing, feature extraction to node embedding[J]. ACM Computing Surveys (CSUR), 55(2): 1−35.

[4] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[J]. ArXiv, 2013(11): 2524.

[5] ELHARROUSS O, ABBAD A, MOUJAHID D, et al. A block-based background model for moving object detection[J]. ELCVIA: electronic letters on computer vision and image analysis, 2016, 15(3): 17−31.

[6] MORALES E F, MURRIETAC R, BECERRA I, et al. A survey on deep learning and deep reinforcement learning in robotics with a tutorial on deep reinforcement learning[J]. Intelligent Service Robotics, 2021, 14(5): 773−805.

[7] SIMONYAN K, ZISSERMAN A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. ArXiv, 2014(9): 1556.

[8] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classifification with deep convolutional neural networks[J]. Advances in Neural Information Processing Systems, 2012(25): 1097−1105.

[9] KONG T, YAO A, CHEN Y, et al. HyperNet: towards accurate region proposal generation and joint object detection[C]// In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas: IEEE, 2016: 845−853.

[10] AMJOUD A B, AMROUCH M. Convolutional neural networks backbones for object detection[C]// In International Conference on Image and Signal Processing. Springer Cham, 2020: 282−289.

[11] AYOUB B A, MUSTAPHA A. Convolutional Neural Networks Backbones for Object Detection[C]// ICISP: Image and Signal Processing. ICISP, 2020: 282−289.

[12] WEI G F, LI G, ZHAO J, et al. Development of a LeNet-5 Gas Identification CNN Structure for Electronic Noses[J]. Sensors, 2019, 19(1): 217.

[13] HE K M, ZHANG X Y, REN S Q, et al. Deep Residual Learning for Image Recognition[C]// IEEE Conference on Computer Vision and Pattern Recognition(CVPR). IEEE, 2015: 770−778.

[14] CHRISTIAN S, SERGEY I, VINCENT V. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning[J]. ArXiv, 2016(2): 07261.

[15] MA Z, WEI X, HONG X, et al. Bayesian loss for crowd count estimation with point supervision[C]// in Proc IEEE Int. Conf. Comput. IEEE, 2019: 6142−6151.

[16] 卢宏涛, 张秦川. 深度卷积神经网络在计算机视觉中的应用研究综述[J]. 数据采集与处理, 2016, 31(1): 1−17.

[17] 胡正平, 陈俊岭, 王蒙. 卷积神经网络分类模型在模式识别中的新进展[J]. 燕山大学学报, 2015, 39(4): 282−289.

[18] 贺昱曜, 李宝奇. 一种组合型的深度学习模型学习策略[J]. 自动化学报, 2016, 42(6): 953−958.

[19] DUCHI J C, HAZAN E, SINGER Y. Adaptive subgradient methods for online learning and stochastic optimization[J]. The Journal of Machine Learning Research, 2011, 12: 2121−2159.

[20] SENIOR A, HEIGOLD G, RANZATO M A, et al. An empirical study of learning rates indeep neural networks for speech recognition[C]//In: Proceedings of the 2013 IEEE International Conferenceon Acoustics, Speech, and Signal Processing. Vancouver, BC: IEEE, 2013: 6724−6728.

[21] 毛宁, 杨德东, 杨福才, 等. 基于分层卷积特征的自适应目标跟踪[J]. 激光与光电子学进展, 2016, 53(12): 195−206.

[22] 芮挺, 费建超, 周游, 等. 基于深度卷积神经网络的行人检测[J]. 计算机工程与应用, 2016, 52(13): 163−168.

[23] 黄曦, 张艾群, 陈俊. 基于着陆器平台的一种运动目标检测算法[J]. 电子测量技术, 2016, 39(9): 77−81.

[24] KALAL Z, MIKOLAJCZYK K, MATAS J. Tracking-learning-detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1409−1422.

[25] KRIZHEVSKY A, SUTSKEVER I, HINTON G. Imagenet classifi-cation with deep convolutional neural networks[C]. Advancesin Neural Information Processing Systems (NIPS2012), 2012: 1106−1114.

[26] KLEIN S, PLUIM J P W, STARING M, et al. Adaptive stochastic gradient descent optimisation for image registration[J]. International Journal of Computer Vision, 2009, 81(3): 227−239.

[27] SCHMIDHUBER J. Deep learning in neural networks: an overview[J]. Neural Networks, 2015, 61(7553): 85−117.

[28] 张俊, 李鑫. TensorFlow 平台下的手写字符识别[J]. 电脑知识, 2016, 12(16): 199−201.

[29] 张子夫. 基于卷积神经网络的目标识别跟踪算法研究与实现[D]. 长春: 吉林大学, 2015.

[30] 吴翔, 钟雨轩, 岳琪琪, 等. 基于深度学习的尺度自适

应海面目标跟踪算法[J]. 水下无人系统学报, 2020, 28(6): 618−625.
[31] 江波, 屈若锟, 李彦冬, 等. 基于深度学习的无人机航拍目标检测研究综述[J]. 航空学报, 2021, 42(4): 137−151.
[32] 李祥霞, 吉晓慧, 李彬. 细粒度图像分类的深度学习方法[J/OL]. 计算机科学与探索: 1-14[2021-06-21]. http: // kns. cnki. net/kcms/detail/11. 5602. TP. 20210603. 1655. 009. html.
[33] SIMONYAN K, ZISSERMAN A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. ArXiv preprint arXiv: 1409. 1556, 2014.
[34] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, 2016: 770−778.
[35] CHRISTIAN S, LIU W, JIA YQ, et al. Going deeper with convolutions[C]// 2015 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, 2015: 1−9.
[36] 江绪庆. Zynq UltraScale+MPSoC 的嵌入式最小系统开发[J]. 单片机与嵌入式系统应用, 2019, 19(1): 26−29.
[37] 张超. P-R 曲线与模型评估问题研究[J]. 现代信息科技, 2020, 31(4): 23−24, 27.

****

（上接第 34 页）

综上，使用多个小型捏合机来等效代替大型捏合机的方式不仅可以提升捏合工艺生产中的安全性与效率，而且为自动化生产线的建立提供了安全性与效率的保障，有利于解放人力，提升产品质量，推进国防工业的发展。

参考文献：

[1] 吴凯, 刘玉存, 刘仕瑞. PBX 炸药概述及其发展与前景[J]. 山西化工, 2012, 32(2): 36−39.
[2] 马秀清, 林群章, 李晓卫. 捏合块元件混合含能材料的安全性分析[J]. 塑料, 2019, 48(1): 88−91.
[3] 关立峰, 吴奎先, 张明, 等. 大药量浇注 PBX 炸药工程化设计[J]. 兵工自动化, 2010, 29(4): 23−24, 29.
[4] 王正方, 翟瑞清. 立式捏合机搅拌桨的设计[J]. 固体火箭技术, 1993(1): 65−69.
[5] 梁建, 李锡文, 史铁林, 等. 立式捏合机桨叶结构参数对混合釜流场影响的仿真分析[J]. 固体火箭技术, 2017, 40(3): 347−352.
[6] 张峰峰. 浇注 PBX 炸药物料混合及固化工艺数值模拟与安全评估[D]. 太原: 中北大学, 2020.
[7] 左亚帅, 张会锁, 姚静晓, 等. 水滴帷幕对战斗部自动化生产线防殉爆性能仿真[J]. 兵工自动化, 2022, 41(5): 85−91.
[8] 曾昭雄. 塑料粘结炸药配方设计中爆热的估算[J]. 火炸药, 1981(5): 29−32.
[9] 张先峰, 李向东, 沈培辉, 等. 终点效应学[M]. 北京: 北京理工大学出版社, 2017: 85−86.

****

（上接第 71 页）

[7] HOLLNAGEL E. The phenotype of erroneous actions[J]. International Journal of Man-machine Studies, 1993, 39(1): 1−32.
[8] SWAIN A D, GUTTMANN H E. Handbook of human reliability analysis with emphasis on nuclear power plant applications(NUREG/CR-1278)[R]. Washington DC: U.S. Nuclear Regulatory Commission, 1983.
[9] RASMUSSEN J. Skills, rules and knowledge; Signals, signs and symbols, and other distinctions in human performance models[J]. IEEE Transactions on System, Man, and Cybernetics, 1983, 13(3): 257−266.
[10] REASON J T. Human error[M]. Cambridge: Cambridge University Press, 1990.
[11] HOLLNAGEL E. Cognitive reliability and error analysis method(CREAM)[M]. London: Elsevier Science Ltd., 1998.
[12] USNRC. Technical basis and implementation guidelines for A Technique for Human Event Analysis (NUREG-1624)[R]. Washington DC: U.S. Nuclear Regulatory Commission, 2000.
[13] BLACKMAN H S, GERMAN D I, BORING R L. Human error quantification using performance shaping factors in the SPAR-H method[C]//52th annual meeting of the human factors and ergonomics society. Los Angeles: SAGE Publications, 2008.
[14] 孙志强, 史秀建, 李欣欣, 等. 基于认知模型的人为差错分类方法[J]. 国防科技大学学报, 2008, 30(1): 73−77.