# RelativeNAS: Relative Neural Architecture Search via Slow-Fast Learning

Hao Tan , Ran Cheng , Senior Member , IEEE, Shihua Huang, Cheng He , Member , IEEE, Changxiao Qiu, Fan Y ang, and Ping Luo

**摘要**

尽管卷积神经网络(CNN)在计算机视觉方面取得了显著的成功，但手动设计卷积神经网络非常耗时且容易出错。在旨在实现高性能神经网络设计自动化的各种神经结构搜索方法中，可区分神经结构搜索方法和基于种群的神经结构搜索方法因其独特的特性而受到越来越多的关注。为了取长补短，本文提出了一种新的NAS方法RelativeNAS。作为高效搜索的关键，RelativeNAS以成对方式在快速学习器(即损失值相对较低的解码网络)和慢学习器之间进行联合学习。此外，由于RelativeNAS只需要低保真的性能估计来区分每对快学习器和慢学习器，因此为训练候选体系结构节省了一定的计算成本。提出的RelativeNAS具有几个独特的优势：1）它在ImageNet上实现了最先进的性能，TOP-1错误率为24.88%，即分别比DARTS和AmoebaNet-B高1.82%和1.12%；2）在单个1080TiGPU上仅需9h即可获得发现的单元，即分别比DARTS和AmoebaNet快3.75倍和7875倍；3）将在CIFAR-10上发现的单元直接用于目标检测、语义分割和关键点检测，在PASCAL VOC上得到了73.1%的MAP，在CIFAR-10上得到了78.7%的MIOU，在MSCOCO上得到了68.5%的AP。

**关键词：** AutoML；卷积神经网络(CNN)；神经体系结构搜索(NAS)；基于种群的搜索；慢速-快速学习。

## 1 引言

深度卷积神经网络(CNN)在各种计算机视觉任务（如图像分类[1]-[3]、目标检测[4]、语义分割[5]、[6]）中取得了显著的效果，自2012年以来专家们设计了一系列最先进的网络[7]-[9]。由于人工设计CNN严重依赖于专家知识和经验，因此通常费时且容易出错。为此，研究人员转向为任何给定任务自动生成高性能网络架构，也称为神经架构搜索(NAS)[10]。

尽管最近的NAS方法在图像分类[12]、[13]、目标检测[14]、[15]、语义分割[16]、[17]以及设计生成式对抗网络[18]等方面都有很好的性能，但仍然存在两大挑战：1）由于缺乏关于体系结构与其性能之间的确切函数关系的先验知识，NAS被视为一个黑箱优化问题；2）NAS由于需要对候选体系结构进行大量的性能评估而存在计算代价过高的问题。

在各种NAS方法中，可微分NAS（即DARTS）[19]和基于种群的NAS[20]因其在应对各种挑战方面的独特优点而最受欢迎：DARTS主要受益于搜索效率高的优点，其优点是将搜索空间放宽为连续的；基于种群的NAS主要受益于种群中各种候选结构的优点，并使用遗传算子(如交叉/突变)来驱动搜索过程。然而，它们也存在一些不足：由于DART联合训练超网并仅通过梯度搜索最优解，在灵活性和通用性方面，它的健壮性比较低，基于种群的NAS主要依靠随机交叉/变异进行搜索，通常需要大量的计算代价来进行性能评估。

一个研究问题是：我们能否从可微分NAS和基于种群的NAS的优点中获益，同时克服它们的不足？为了回答这个问题，本工作提出了一种新的RelativeNAS方法。

特别地，受[19]的启发，通过考虑成对结点和相应操作之间的联系，提出了一种新颖的基于单元的搜索空间的连续编码方案。然而，与[19]中给出的编码方法相比，该编码方法没有可微性要求，也没有考虑选择运算的概率/权重；相反，它以一种幼稚的方式将成对节点之间的运算直接编码为实值（如图1所示）。提出的连续编码方法的主要优点是：1）它提供了更多的灵活性和通用性；2）当应用于基于种群的NAS时，扩大的搜索空间鼓励对不同体系结构的搜索[21]。

在所提出的连续编码方案的基础上，受[22]的启发，该工作进一步提出了一种在编码空间中进行高效搜索的慢-快学习范式。在这个范例中，体系结构向量的群体被迭代地配对和更新。具体地说，在所提出的慢-快学习的每一次迭代中，体系结构向量被随机配对；对于每对体系结构向量，性能较差的体系结构向量（表示为慢学习器）通过向性能较好的体系结构向量（表示为快学习器）学习来更新。与大规模进化[23]和AmoebaNet[20]等基于群体的NAS方法相比，提出的慢-快学习范式不涉及任何遗传算子（例如交叉/变异），但本质上是使用伪梯度机制来更新体系结构向量，该机制旨在隐式地学习每对慢学习者和快学习者之间的联合分布。具体来说，伪梯度是由学习速度快和学习速度慢的学生之间的两两学习决定的。提出的慢-快学习范式的主要优点是：1）它提供了一种在通用的连续搜索空间中执行NAS的方案，而无需考虑其特定的性质（例如，可微性）；2）它提供了一种学习多个体系结构的联合分布的方法。

为了提高RelativeNAS的计算效率，本文进一步采用权集作为知识库，用于比较每对体系结构的性能，权集是所有候选体系结构的操作集合，也是种群中有前途的知识的集合。由于慢-快学习在区分每对中的慢学习者和快学习者时只需要低保真的性能估计，因此新发现的网络只需要训练一个历元来获得估计的性能，从而节省了大量的性能评估计算代价。值得注意的是，权重集不是直接训练的，而是通过在线方式训练每个配对网络的权值。

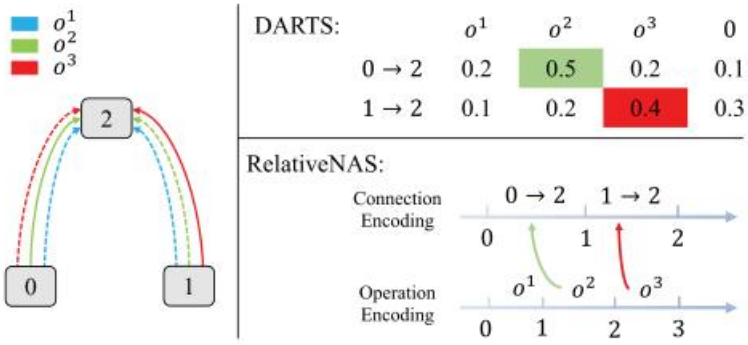


图 1: DARTS和RelativeNAS的编码方案示例

## 2 相关工作

本篇论文对比了可微分NAS，这里主要以DARTs为代表，和基于种群的NAS，阐述了他们各自的优缺点，DARTs的主要优势在于它把离散的搜索空间放宽为连续的，具有较高的搜索效率，但是在灵活性和鲁棒性方面性能较差。基于种群的NAS优势在于其候选体系结构较多，会大量使用遗传算子如交叉变异这些操作来驱动搜索过程，因此计算成本也相对较大，搜索效率较低。于是本文提出了RelativeNAS的方法，将两者的优势结合（图2）。

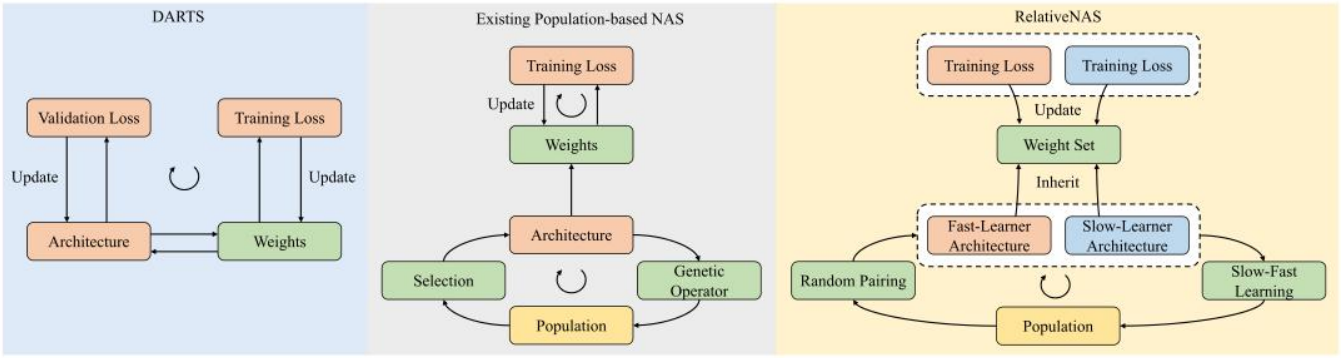


图 2: DARTS，基于种群的NAS和RelativeNAS的一般框架

## 3 本文方法

### 3.1 搜索空间

为了控制搜索空间的大小，这里采用了正常单元和缩减单元这两种单元形式，这两种单元的区别在于输出特征的大小，正常单元不改变特征的大小，缩减单元通过设置步长来减小特征的大小。看中间这个图，前两个单元各有两个输入节点。每个中间节点包含两个前置节点，边对应于它们的应用操作，仅允许边从索引较低的节点指向索引较高的节点。另一方面，一个单元只包含一个输出，并且所有中间节点都连接到输出节点。此外，还原单元是在正常单元之后连接的，如右图所示。同时由于输入节点和输出节点是固定的，该工作只需要对每个中间节点及其两个前驱节点和相应的操作进行编码。图3给出了上述编码过程的一个说明性示例。这项工作使用块列表来表示图3（左）所示的体系结构向量。每个块代表单元中的一个中间节点，需要由四个变量指定，包括Pre1 Node（第一个前置节点）和Pre2 Node（第二个前置节点），以及它们对应的操作。

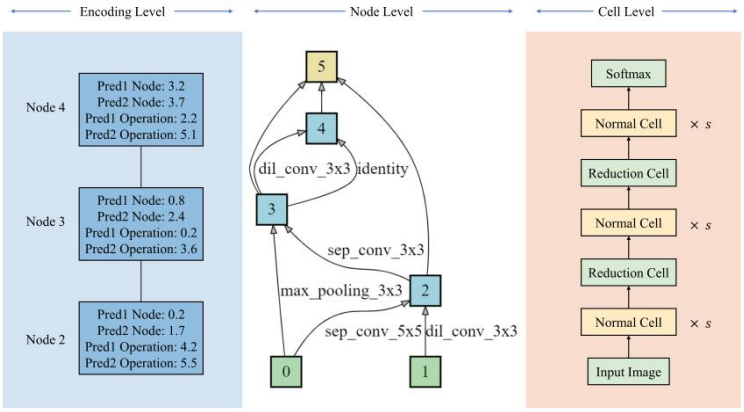


图 3: 映射到基于单元的结构的中间节点的编码向量的示例

### 3.2 慢-快学习

NAS的总目标是搜索体系结构向量$\alpha^* \in A$，使得解码的体系结构$C_{\alpha^*}$最小化验证损失$Lvld(C_{\alpha^*}, \omega^*_{\alpha^*})$，其中与该体系结构相关联的权重$\omega^*_{\alpha^*}$通过最小化训练损失$Ltra(C_{\alpha^*}, \omega_{\alpha^*})$来获得。第g代的体系结构向量是这样更新的，为了有效地生成伪梯度项，本工作提出使用N个体系结构向量的种群，在每一代，种群被随机分成N/2对。然后，对于每对p，通过验证损失值的偏序来指定快学习器和慢学习器，其中具有较小损失的一个是快学习器，另一个是慢学习器（图4）。然后通过慢速学习器向快速学习器学习进行更新，最终，所有的快学习器和更新的慢学习器被重新合并成为下一代g+1的新种群。通过这样一个慢-快学习的迭代过程，种群中的每个结构向量通过向比它们收敛得更快的结构向量学习而朝着最优方向移动。

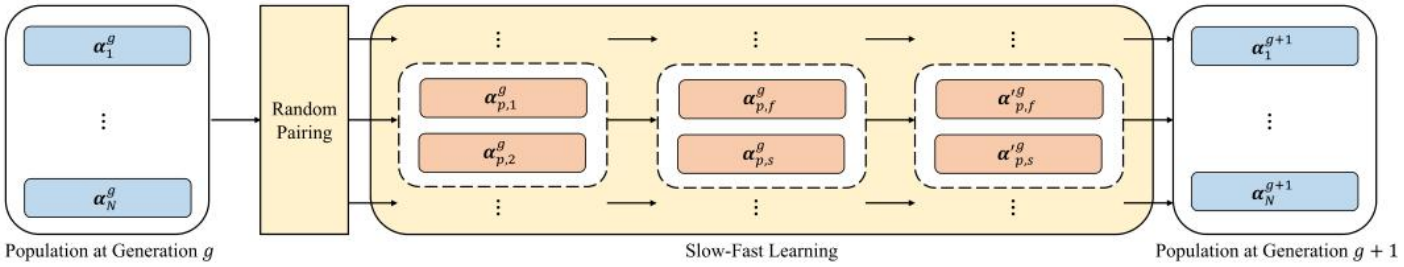


图 4: 第g代的慢-快学习过程

### 3.3 性能评估

提出的RelativeNAS需要评估从每一代群体中的体系结构向量解码的候选体系结构的性能，以便对于每一对体系结构向量，可以通过验证损失来区分快速学习器和慢速学习器

为了减少RelativeNAS中性能评估的计算开销，因为训练候选体系结构可能在计算上相当昂贵，特别是当在迭代的慢速-快速学习过程中获得大量候选体系结构时

与现有的可微NAS方法（例如，DARTS[19]）不同，RelativeNAS中的验证损失并不直接涉及候选体系结构的更新；相反，它们仅用于确定每对候选体系结构中的偏序（即，区分快学习者和慢学习者）。因此，在RelativeNAS中，使用性能估计（而不是精确的性能评估）来获得候选体系结构的近似验证损失是直观可行的。

随机初始化一个权重集，在搜索过程中，给定第p个候选体系结构对$\{C_{\alpha^g_p}, j\}_{j=1,2}$，它们分别根据自己的操作继承相应的权重$\omega_{\alpha^g_{p,1}}$和$\omega_{\alpha^g_{p,2}}$，利用继承的权重作为预热，只需要通过一步优化在训练集$D_{trn}$上更新$\{C_{\alpha^g_{p,j}}\}_{j=1,2}$的权重，然后利用更新好的权重去评估候选架构在验证数据集上的损失，最后，对初始化的这个权重集进行更新。第一个表示从快速学习器上接收所有权重，因为假设$\omega_{\alpha^g_p}$，f作为快速学习器的权重通常比$\omega_{\alpha^g_{p,s}}$更有价值。第二个$\{\omega_{\alpha^g_{p,s}} - \omega_{\alpha^g_{p,f}} \cap \omega_{\alpha^g_{p,s}}\}$表示接收$\omega_{\alpha^g_{p,s}}$中的权重，但不接收$\omega_{\alpha^g_{p,f}}$中的权重。第三个术语$\{\Omega - \omega_{\alpha^g_{p,f}} \cup \omega_{\alpha^g_{p,s}}\}$表示保持这些未使用的权重不变。

之后，我们只对每个网络进行一个时期的训练，并将其区分为快学习器和慢学习器。最后，通过所有训练权重$\omega_g$来更新权重集。

## 4 复现细节

### 4.1 与已有开源代码对比

该复现过程引用参考了作者的源代码，包括了模块和数据库。本次复现在作者原来的基础上，将作者所设置的50个epoch，用3个stage来代替，每个stage包含了20个epoch（图5），通过调整该设置相当于在每个stage结束后对参数进行了更新，可以减轻在迭代后期验证精度和验证损失变化较慢的问题。

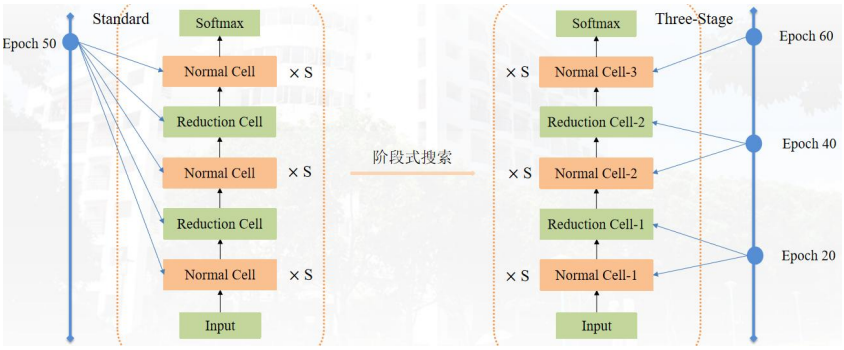


图 5: Three-Stage示意图

### 4.2 结构搜索

基本上，这项工作是在CIFAR-10[37]上执行的，CIFAR-10[37]被广泛用于基准图像分类。具体地说，CIFAR-10包含60K图像，空间分辨率为32×32，这些图像被平均分为10个类别，其中训练集和测试集分别为50K和10K。随机抽取CIFAR-10训练图像的一半作为搜索验证集。

在RelativeNAS中，种群大小N和世代数分别被设置为20和60。为了评估所发现的体系结构，每个体系结构向量α首先被解码成一个8个单元（即，s=2）的小网络，初始信道设置为16。然后，该工作使用权值衰减和批量大小分别设置为$3\times10^{-4}$和256时的随机梯度下降法在训练集上对这些网络进行一个历元的训练。

另外，初始学习率LR是0.1，其在将Tmax设置为世代数的余弦退火表之后衰减到零，并且长度为16的裁剪[38]和概率为0.3的路径丢弃[39]都被应用于正则化。在验证集上对训练好的网络进行评估，通过比较验证损失来区分快学习和慢学习。总而言之，使用独特的1080Ti大约需要12小时才能完成上述搜索过程。

### 4.3 慢-快学习分析

为了实证分析慢-快学习过程，本工作随机选取了分别在第1代、第30代和第60代获得的三对体系结构，以加以说明性示例。在第一代，快学习者和慢学习者的体系结构在开始时随机初始化。因此，快学习者和慢学习者之间存在着本质上的差异，慢学习者在从快学习者那里学习后，其连接和操作都会发生实质性的变化。在第30代，快学习器和慢学习器之间有一些常见的连接模式，例如，结点4的两个前驱节点在正常细胞中都是结点1，结点3的两个前辈节点在约简细胞中都是结点0。因此，学习速度慢的人不会在慢-快的学习过程中改变这些模式。相反，由于快慢学习者在节点2和节点0之间的连接中Dil Conv 3×3和Sep Conv 5×5的不同，慢学习者向快学习者学习并改变为Dil Conv 3×3。在第60代，正常细胞的连接是完全相同的，因此慢学习者仅通过向快速学习者学习来对其操作进行一些微小的调整。

## 5 实验结果分析

在本次实验中，种群大小N和世代数分别被设置为20和60。为了评估所发现的体系结构，每个体系结构向量α首先被解码成一个8个单元(即，s=2)的小网络，初始信道设置为16。然后，该工作使用权值衰减和批大小分别设置为$3\times10^{-4}$和256时的SGD在训练集上对这些网络进行训练。

另外，初始学习率LR是0.1，其在将Tmax设置为世代数的余弦退火表之后衰减到零，并且长度为16的裁剪和概率为0.3的路径丢弃都被应用于正则化。在验证集上对训练好的网络进行评估，通过比较验证损失来区分快学习和慢学习。在本次复现过程中，使用1080Ti大约搜索了12小时，在cifar10数据集上搜索到的最好验证精度为97.268%。

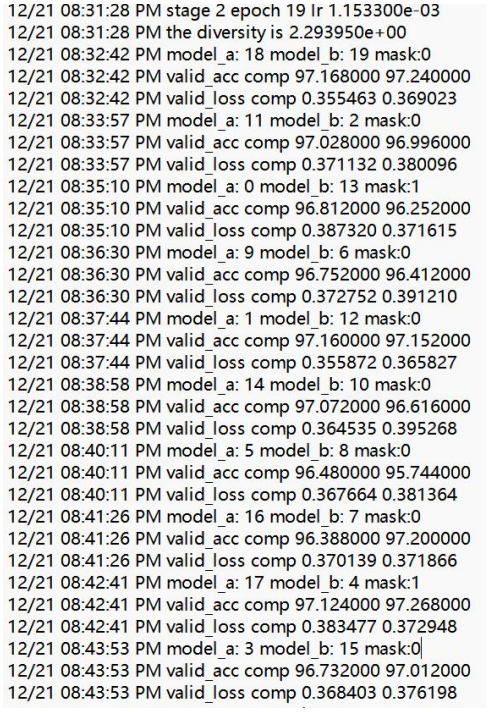


图 6：搜索结果示意图

# 6 总结与展望

本文提出了一个称为RelativeNAS的框架，用于有效和高效地自动设计高性能网络。在RelativeNAS中，首次提出了一种新的基于单元搜索空间的连续编码方案。为了进一步利用连续编码的搜索空间，应用了慢-快学习范例作为优化器来迭代地更新体系结构向量。

与NAS中已有的学习优化方法不同，该方法不直接使用基于损失的知识来更新体系结构。相反，候选体系结构通过成对生成的伪梯度相互学习，即在每对候选体系结构中，慢学习器向快学习器学习。此外，还提出了一种性能评估策略，以降低评估候选体系结构的成本。这种策略的有效性在很大程度上可以归因于这样一个事实，即验证损失仅用于通过偏序来区分慢学习器和快学习器，这只需要估计的（而不是准确的）损失值。

本次复现RelativeNAS在CIFAR-10上搜索大约需要12个1080Ti GPU小时（即0.5 GPU Day）。此外，使用该方法搜索出来的网络已经能够在CIFAR-10上超越或匹配其他最先进的方法和NAS网络。

# 参考文献


[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf.Process. Syst., 2012, pp. 1097－1105.

[2] Y . Kong et al., “Constructing an automatic diagnosis and severity classification model for acromegaly using facial photographs by deep learning,” J. Hematology Oncol., vol. 13, no. 1, pp. 1－4, Dec. 2020.

[3] Z. Lu, K. Deb, and V . N. Boddeti, “MUXConv: Information multiplexing in convolutional neural networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 12044－12053.

[4] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards realtime object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91－99.

[5] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networksfor semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 3431－3440.

[6] C. Y u, J. Wang, C. Peng, C. Gao, G. Y u, and N. Sang, “BiSeNet: Bilateral segmentation network for real-time semantic segmentation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 325－341.

[7] C. Szegedy et al., “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1－9.

[8] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.

[2] (CVPR), Jun. 2016, pp. 770－778.

[9] G. Huang, Z. Liu, L. V an Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 4700－4708.

[10] T. Elsken, J. H. Metzen, and F. Hutter, “Neural architecture search: A survey,” J . M a c h . L e a r n . R e s ., vol. 20, no. 55, pp. 1－21, 2019.

[11] Z. Lu et al., “Multiobjective evolutionary design of deep convolutional neural networks for image classification,” IEEE Trans. Evol. Comput., vol. 25, no. 2, pp. 277－291, Apr. 2021.

[12] Z. Zhong et al., “BlockQNN: Efficient block-wise neural network architecture generation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 7, pp. 2314－2328, Jul. 2021.

[13] Z. Lu, G. Sreekumar, E. Goodman, W. Banzhaf, K. Deb, and V . N. Boddeti, “N eural architecture transfer,” IEEE Trans. Pattern Anal. Mach. Intell., early access, Jan. 19, 2021, doi: 10.1109/TPAMI.2021. 3052758.

[14] G. Ghiasi, T.-Y . Lin, and Q. V . Le, “NAS-FPN: Learning scalable feature pyramid architecture for object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 7036－7045.

[15] H. Xu, L. Y ao, Z. Li, X. Liang, and W. Zhang, “Auto-FPN: Automatic network architecture adaptation for object detection beyond classification,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 6649－6658.

[16] C. Liu et al., “Auto-DeepLab: Hierarchical neural architecture search for semantic image segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 82－92.

[17] W. Chen, X. Gong, X. Liu, Q. Zhang, Y . Li, and Z. Wang, “FasterSeg: Searching for faster real-time semantic segmentation,” in Proc. Int. Conf. Learn. Represent., 2020, pp. 1－14.

[18] X. Gong, S. Chang, Y . Jiang, and Z. Wang, “AutoGAN: Neural architecture search for generative adversarial networks,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 3224–3234.

[19] H. Liu, K. Simonyan, and Y . Y ang, “DARTS: Differentiable architecture search,” in Proc. Int. Conf. Learn. Represent., 2019, pp. 1–12.

[20] E. Real, A. Aggarwal, Y . Huang, and Q. V . Le, “Regularized evolution for image classifier architecture search,” in Proc. AAAI Conf. Artif.Intell., vol. 33, 2019, pp. 4780–478.

[21] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, “Designing neural networks through neuroevolution,” Nature Mach. Intell., v o l . 1 , pp. 24–35, Jan. 2019.

[22] C. Feichtenhofer, H. Fan, J. Malik, and K. He, “SlowFast networks for video recognition,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 6202–6211.

[23] E. Real et al., “Large-scale evolution of image classifiers,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 2902–2911.

[24] S. Xie, H. Zheng, C. Liu, and L. Lin, “SNAS: Stochastic neural architecture search,” in Proc. Int. Conf. Learn. Represent., 2019, pp. 1–17.

[25] H. Cai, L. Zhu, and S. Han, “ProxylessNAS: Direct neural architecture search on target task and hardware,” in Proc. Int. Conf. Learn. Represent., 2019, pp. 1–13.

[26] H. Liang et al., “DARTS+: Improved differentiable architecture search with early stopping,” 2019, arXiv:1909.06035. [Online]. Available: http://arxiv.org/abs/1909.06035

[27] X. Y ao, “Evolving artificial neural networks,” Proc. IEEE, vol. 87, no. 9, pp. 1423–1447, Sep. 1999.

[28] K. O. Stanley and R. Miikkulainen, “Evolving neural networks through augmenting topologies,” Evol. Comput., vol. 10, no. 2, pp. 99–127, 2002.

[29] M. Suganuma, S. Shirakawa, and T. Nagao, “A genetic programming approach to designing convolutional neural network architectures,” in Proc. Genetic Evol. Comput. Conf., Jul. 2017, pp. 497–504.

[30] Z. Lu et al., “NSGA-Net: Neural architecture search using multi-objective genetic algorithm,” in Proc. Genetic Evol. Comput. Conf., Jul. 2019, pp. 419–427.

[31] Z. Lu, K. Deb, E. Goodman, W. Banzhaf, and V . N. Boddeti, “NSGANetV2: Evolutionary multi-objective surrogate-assisted neural architecture search,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 35–51.

[32] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, “SMASH: One-shot model architecture search through hypernetworks,” in Proc. Int. Conf.

Learn. Represent., 2018, pp. 1–5.
[33] G. Bender, P .-J. Kindermans, B. Zoph, V . V asudevan, and Q. Le, "Understanding and simplifying one-shot architecture search," in Proc. Int. Conf. Mach. Learn., 2018, pp. 549–558.
[34] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1251–1258.
[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proc. Int. Conf. Mach. Learn., vol. 37, Jul. 2015, pp. 448–456.
[36] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, pp. 533–536, Oct. 1986.
[37] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.
[38] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, arXiv:1708.04552. [Online]. Available: http://arxiv.org/abs/1708.04552
[39] G. Ghiasi, T.-Y . Lin, and Q. V . Le, "Dropblock: A regularization method for convolutional networks," in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 10727–10737.