中国科学技术大学人工智能实践课程
技术报告

基于 CAFormer 的猫狗图像二分类

朱云沁 PB20061372

中国科学技术大学

2024 年 1 月

目录

1 实验目标

图像二分类. 数据集中有猫和狗两类图片, 设计模型, 尽可能正确分类猫和狗.

1.1 数据描述

数据集中分别有 1000 张猫和 1000 张狗的图片. 我们采用 8:2 的比例划分训练集和测试集, 将测试集进一步划分为 5 折, 用于交叉验证. 各子集中猫和狗的数量如下表所示.

	Training Set					Test Set
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
# of Cat	160	160	160	160	160	200
# of Dog	160	160	160	160	160	200

表 1: 数据集划分

通过随机采样, 得到训练集中若干图像样本如下图所示.

图 1: 训练集中的图像样本

2 国内外研究现状

传统的图像分类方法难以处理庞大的图像数据, 且无法满足人们对图像分类精度和速度的要求, 而基于深度学习的图像分类方法突破了此瓶颈, 成为目前图像分类的主流方法.

2.1 基于 CNN 的图像分类模型

自上世纪 90 年代以来, 卷积神经网络 (CNN) 在图像分类领域取得了巨大的成功. 1998 年, LeCun 等人率先提出 LeNet-5 [1], 采用卷积层和池化进行特征提取,

成为深度学习的先驱之一. AlexNet [2] 由 Krizhevsky 等人于 2012 年提出, 通过深度架构实现了重大突破, 利用 Dropout 进行正则化, 并使用 ReLU 激活函数. VGG [3] 是由 Simonyan 和 Zisserman 于 2014 年提出, 强调采用统一的架构和小的感受野. GoogLeNet [4] 由 Szegedy 等人于 2014 年设计, 引入了 Inception 模块, 促进了在不同尺度上并行处理特征. 微软亚洲研究院的何恺明 (毕业于香港中文大学) 等人于 2015 年提出 ResNet [5], 通过残差连接解决了梯度消失问题, 使得训练极深网络成为可能. Meta 的刘壮 (毕业于清华大学) 等人于 2022 年设计 ConvNeXt [6], 作为一种现代化的纯卷积网络模型, 在多项任务上取得了先进的性能.

2.2 基于 Transformer 的图像分类模型

近几年, 随着 Transformer [7] 在自然语言处理领域的成功, 人们开始将其应用于计算机视觉领域. 2020 年, Dosovitskiy 等人率先提出 ViT [8], 摒弃了卷积架构, 采用自注意机制捕捉长程依赖关系. DeiT [**?**] 是由 Touvron 等人于 2020 年提出, 侧重于通过知识蒸馏实现高效训练, 使用更少的参数实现竞争性能. 微软亚洲研究院的刘泽 (即将毕业于中国科学技术大学) 等人于 2021 年提出 Swin Transformer [9], 引入了分层架构和移位窗口, 增强了可伸缩性并有效地捕捉全局上下文.

这些模型共同推动了图像分类领域, 每个模型都为特征提取、深度、效率和鲁棒性等方面的挑战提供了独特的创新.

3 课题总体方案设计

根据调研结果, 我们采用 CAFormer [10] 作为底层网络架构. 为了进一步提升模型的泛化能力, 我们采用 RandAugment [11] 及 MixUp [12] 两种数据增强方法. 作为另一个改进方向, 我们使用在 ImageNet-1K 数据集上预训练得到的权重初始化模型参数, 在猫狗图像数据集上进行微调, 并将其与随机初始化的模型进行对比.

3.1 CAFormer: 一种先进的视觉网络架构

近期的一些研究表明, Transformer 在视觉领域的能力归因于其一般架构, 而并不依赖于其注意力机制. 基于此观察, 来自新加坡 Sea AI Lab 的余伟浩等人从抽象出 MetaFormer 作为通用的视觉网络架构 [13]. 该架构不限制特定的 Token Mixer, 而是允许用任意模块混合不同 Token 的特征, 例如注意力、卷积、池化、恒等映射. 作为一种实例, CAFormer 在不同阶段先后采用可分离卷积 (Separable Convolution) 和多头自注意力 (Multi-Head Self-Attention) 混合不同 Token 的特征, 在 ImageNet 上取得了最先进的性能, 超越了 ConvNeXt 和 Swin Transformer

等单独采用卷积或注意力的模型. 因此, 我们借鉴 CAFormer 的思想, 设计猫狗图像分类的网络架构.

一个用于猫狗图像分类的四阶段 CAFormer 如下图所示.

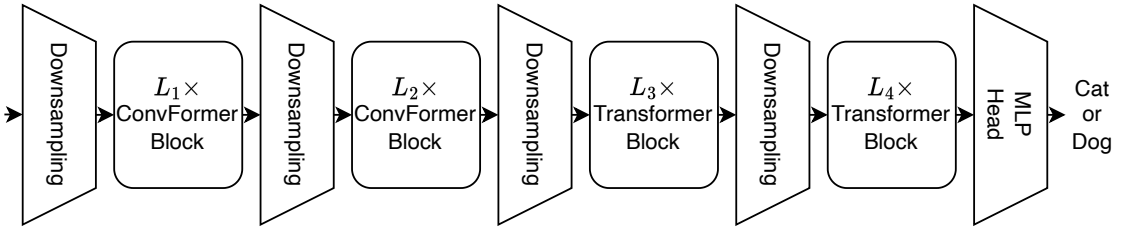


图 2: 四阶段 CAFormer 的架构

其中, 第一个下采样层采用 7×7 卷积, 后续下采样层采用 3×3 卷积, 分类器采用 2 层感知机. ConvFormer Block 和 Transformer Block 是 CAFormer 的基本模块, 分别采用可分离卷积和自注意力作为 Token Mixer, 其结构如下图所示.

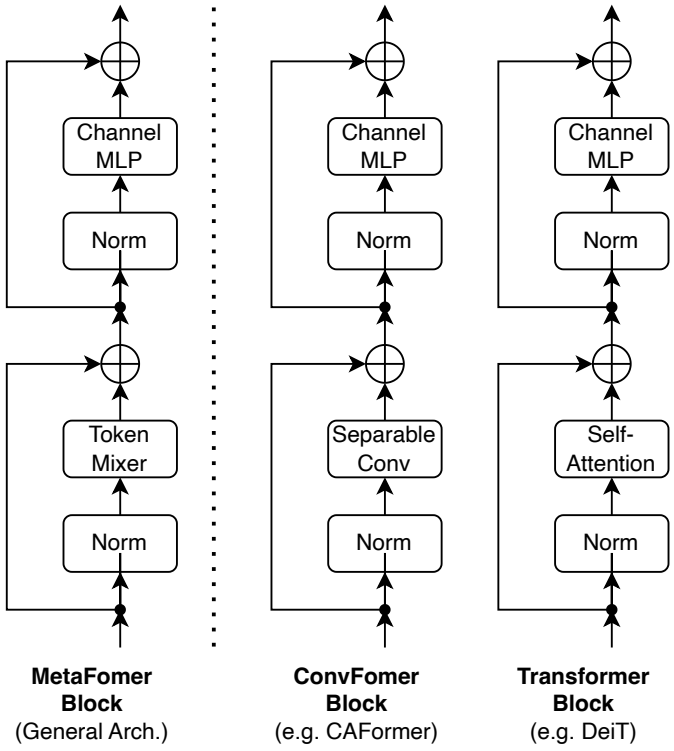


图 3: MetaFormer Block, ConvFormer Block 及 Transformer Block 的对比

其中, 可分离卷积是指对特征图应用 7×7 逐深度 (Depthwise) 卷积和一维逐点 (Pointwise) 卷积的组合, 在保持常规卷积有效性的同时大大减少了参数量和计算复杂性; 自注意力是指将特征图各个位置的特征向量视作序列中的 Token, 然后应用常规的多头自注意力机制. 在早期阶段, 特征图尺寸较大, CAFormer 采用卷

积以捕捉局部特征; 在后期阶段, 特征图尺寸较小, CAFormer 采用注意力以捕捉全局特征. 我们统一选用 Layer Normalization 作为归一化方法, 并在 Transformer Block 的残差连接中添加 LayerScale [14] 以进一步提升训练的稳定性.

3.2 (改进) 基于 RandAugment 和 MixUp 的数据增强

用于训练的猫狗图像数据集规模较小. 为了提升模型的鲁棒性, 避免过拟合, 应当采用数据增强方法.

RandAugment [11] 是 Google Brain 团队于 2019 年提出的一种数据增强方法, 通过随机组合多种简单的数据增强方法, 生成复杂的数据增强策略. 该方法避免了过大的超参数搜索空间, 有效减少了人工设计数据增强策略的工作量. 具体而言, RandAugment 涉及如下 14 种变换:

Identity	AutoContrast	Equalize
Rotate	Solarize	Color
Posterize	Contrast	Brightness
Sharpness	ShearX	ShearY
TranslateX	TranslateY	

对于每一张训练图像, 我们首先从上述所有变换集合中以均匀概率选取 $N = 2$ 种变换, 然后以幅度 $M = 9$ 依次施加每一种变换.

MixUp [12] 是由张宏毅等人于 2018 年提出的一种简易而有效的数据增强方法, 通过随机凸组合生成新的训练样本, 以减小模型对对抗样本的敏感性. 具体而言, 对于每一对训练样本 (x_i, y_i) 和 (x_j, y_j), 该方法首先从 Beta 分布 $\mathrm{Beta}(\alpha, \alpha)$ 中采样得到随机系数 $\lambda \in [0, 1]$, 然后生成新的训练样本 $(\lambda x_i + (1-\lambda)x_j, \lambda y_i + (1-\lambda)y_j)$. 在本实验中, 我们选取 $\alpha = 0.2$.

3.3 (改进) ImageNet-1K 预训练

深度神经网络的训练需要大量的数据, 但猫狗图像数据集规模较小. 相较给定的猫狗图像数据集, ImageNet-1K 数据集包含 1000 个类别, 共 1281167 张训练图像和 50000 张验证图像, 规模更大, 质量更高. 因此, 通过在 ImageNet-1K 数据集上预训练, 可以更好地初始化模型参数, 从而提升模型的泛化能力.

具体而言, 我们在猫狗图像数据集上对 [10] 给出的 CAFormer 权重[1]进行微调. 微调时, 下采样层、ConvFormer Block 和 Transformer Block 的权重一律冻结, 仅对 MLP 二分类器的权重进行更新. 我们计划将预训练-微调范式下的模型与从头开始训练的模型进行对比.

[1] https://huggingface.co/timm/caformer_s18.sail_in1k_384

4 模型测试实验设计

4.1 数据预处理

本实验数据集中各图片尺寸及宽高比不一致, 需要按照一定原则进行缩放和裁剪. 考虑到模型训练和评估的不同要求, 应当对训练样本和验证/测试样本采取不同预处理策略.

对于验证/测试样本, 我们首先将图像按最短边缩放为 384 像素, 然后从中心裁剪为 384 像素的正方形; 对于训练样本, 我们随机地从原图像中截取相对面积在 $[0.8, 1.0]$ 之间, 宽高比在 $[0.8, 1.2]$ 之间的矩形区域, 然后将其缩放为 384 像素的正方形, 并作随机水平翻转和进一步数据增强. 上述缩放过程中统一选用双线性插值.

下图展示了对一些样本进行预处理及数据增强的过程.

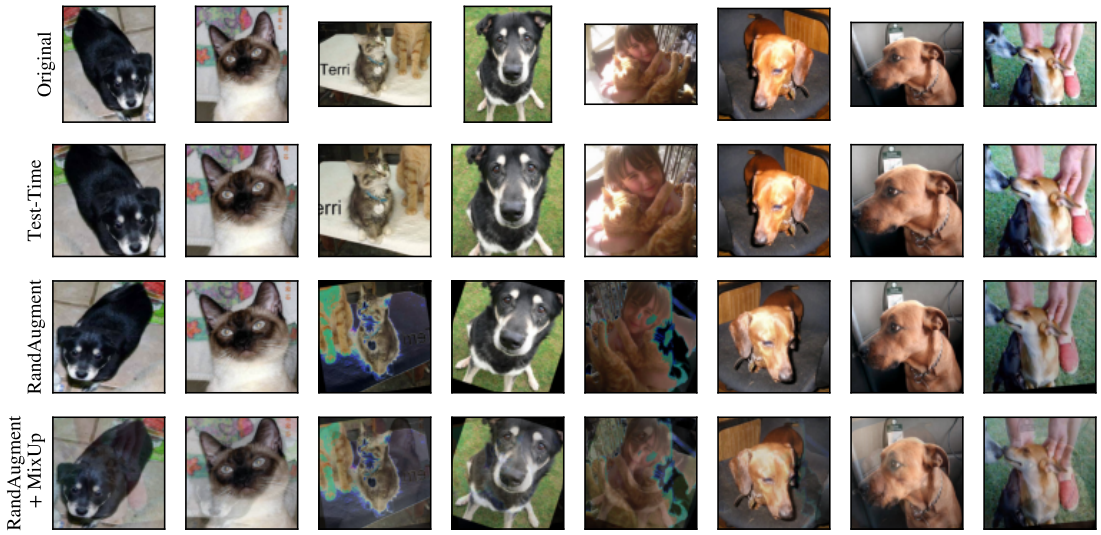


图 4: 数据预处理及增强的示例

4.2 消融实验设计

为了展示两个改进方向的有效性, 我们设计四组实验, 分别对应四种模型:

1. **Baseline**: 随机初始化 + 从头训练, 不进行数据增强.
2. **RandAug.**: 随机初始化 + 从头训练, 仅采用 RandAugment 数据增强.
3. **RandAug. & MixUp**: 随机初始化 + 从头训练, 采用两种数据增强.
4. **Fine-Tuned**: ImageNet-1K 预训练 + 微调, 不进行数据增强.

对于每一个模型, 我们使用 Fold 2~5 作为训练集, Fold 1 作为验证集. 将准确率和 F_1 分数作为评价指标. 在训练过程中记录验证集上的损失函数及准确率变化. 最终, 我们将在测试集上比较猫狗分类的准确率及 F_1 分数.

4.3 超参数设置

对于在 ImageNet-1K 上预训练的模型 (记为 Fine-Tuned), 我们遵循 [13] 中提供的超参数设置. 对于从头训练的模型 (记为 Baseline), 考虑到在小规模数据集上的过拟合风险, 我们独立设计一个三阶段的小型 CAFormer. 其中, 前两阶段采用 ConvFormer Block, 第三阶段采用 Transformer Block. 模型总参数量及部分核心超参数如下表所示.

Model	Total Param.	# of Channels	# of Blocks	Token Mixer
Baseline	113 K	(16,32,64)	(1,1,1)	(Conv,Conv,Attn)
Fine-Tuned	24 M	(64,128,320,512)	(3,3,9,3)	(Conv,Conv,Attn,Attn)

表 2: 模型尺寸及核心超参数设置

所有 MetaFormer Block 采用 StarReLU [10] 激活函数, MLP 分类器采用 SquaredReLU 激活函数. 对于 MetaFormer Block 的残差连接部分, 我们在从头训练和微调时分别采用最大概率为 0.15 和 0.3 的 DropPath [15]. 对于 MLP 分类器的输出层, 我们仅在微调时采用概率为 0.4 的 Dropout. 模型架构的更多细节参见实验代码.

对于所有模型, 使用交叉熵损失函数, 不使用标签平滑. 采用 AdamW 优化器, 从头训练和微调的学习率分别为 1×10^{-4} 和 1×10^{-5}, 权重衰减系数统一为 1×10^{-2}. Batch 大小统一为 32, 每个 Epoch 共计 40 个 Batch. 从头训练的总 Epoch 数设为 1000, 当验证集上的损失函数连续 100 个 Epoch 未下降时, 提前终止训练. 微调的总 Epoch 数设为 3, 共计 120 个 Step.

模型及训练代码采用 PyTorch 实现. 所有实验在 1 块 NVIDIA GeForce GTX 1080 Ti GPU 上进行.

5 模型测试结果及分析

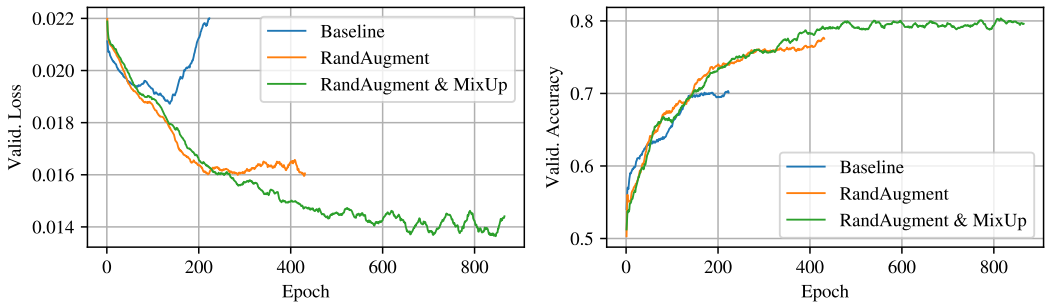
实验主要结果如表 3 和图 5 所示.

5.1 性能比较

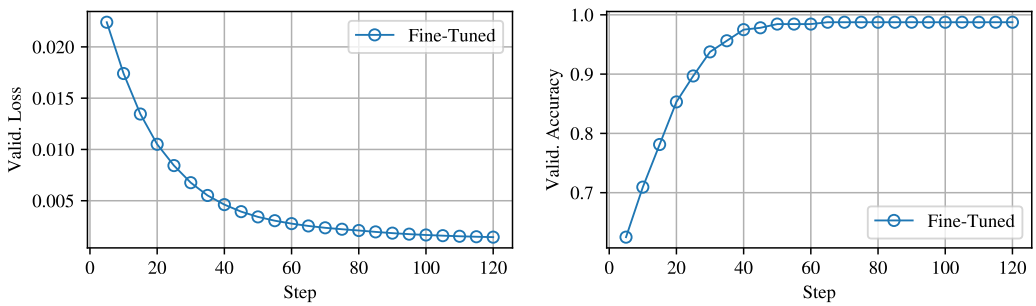
表 3 给出了四个版本模型在训练集、验证集和测试集上的分类准确率及 F_1 分数. 可以看出, 在训练时采取 RandAugment 和 MixUp 有效地提升了模型性能. 单独使用 RandAugment, 得到的模型准确率相较 Baseline 提高了 8.19%; 同时采用两种数据增强策略, 得到的模型准确率相较 Baseline 提高了 11.74%. 从头训练的三个版本模型均表现出过拟合现象, 验证集和测试集的准确率普遍低于训练集.

Dataset	Model	Accuracy	F_1 Score
Train.	Baseline	0.8023	0.8021
	RandAug.	0.8805	0.8804
	RandAug. & MixUp	0.9766	0.9766
	Fine-Tuned	0.9961	0.9961
Valid.	Baseline	0.7031	0.7029
	RandAug.	0.7750	0.7750
	RandAug. & MixUp	0.7969	0.7969
	Fine-Tuned	0.9875	0.9875
Test	Baseline	0.7025	0.7013
	RandAug.	0.7600	0.7598
	RandAug. & MixUp	0.7850	0.7847
	Fine-Tuned	0.9975	0.9975

表 3: 不同版本模型的分类准确率及 F_1 分数



(a) 随机初始化 + 从头训练



(b) ImageNet-1K 预训练 + 微调

图 5: 训练过程中的损失函数及准确率变化

此外, 我们发现, 预训练-微调范式极大地提升了模型性能, 训练集、验证集和测试集的准确率及 F_1 分数几乎达到 100%. 这可能归因于预训练模型的规模较大, 参数量约为 Baseline 的 200 倍; 也归因于在高质量 ImageNet-1K 数据集上进行的充分的预训练, 使得视觉模型能够提取一般的、鲁棒的、可泛化的图像特征; 当然, 由于 ImageNet-1K 涵盖了猫和狗两类图像, 预训练模型也有可能已经记忆了猫和狗的视觉概念 (本实验数据集来源未知, 因此我们不能排除本实验数据集为 ImageNet-1K 子集的可能性). 无论如何, 预训练-微调范式下模型的性能优势是显而易见的, 充分证明了大数据和视觉基础模型的重要性.

5.2 训练曲线分析

图 5 给出了四个版本模型在训练过程中的损失函数及准确率变化. 为了便于观察, 我们对从头训练的模型训练曲线采取了窗口大小为 20 的滑动平均. 可见, 在不采取数据增强策略时, 损失函数很快收敛, 在 150 个 Epoch 后甚至出现了上升的趋势 (归因于过大的权重衰减); 在采取 RandAugment 之后, 模型损失函数收敛于更低的水平, 准确率也有所提升; 在同时采取 RandAugment 和 MixUp 之后, 模型损失函数进一步收敛, 准确率也进一步提升, 乃至 800 个 Epoch 后才触发早停. 这些现象揭示了数据增强策略在避免过拟合、提升模型泛化能力上的重要意义.

6 实验总结

本实验中, 我们采用 CAFormer 作为底层网络架构设计了猫狗图像分类模型, 并在此基础上进行了两个改进: 一是采用 RandAugment 和 MixUp 两种数据增强方法, 二是使用在 ImageNet-1K 数据集上预训练得到的权重初始化模型参数, 在猫狗图像数据集上进行微调. 通过消融实验, 我们发现, 采用 RandAugment 和 MixUp 两种数据增强方法有效地提升了模型性能, 准确率达到约 80%; 预训练-微调范式极大地提升了模型性能, 准确率几乎达到 100%. 这些结果充分证明了先进的模型架构、数据增强策略, 以及大规模数据、视觉基础模型的重要性.

7 参考文献

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. 86(11):2278–2324. Conference Name: Proceedings of the IEEE.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification

with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

[3] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition.

[4] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE.

[6] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale.

[9] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002. IEEE.

[10] Weihao Yu, Chenyang Si, Pan Zhou, Mi Luo, Yichen Zhou, Jiashi Feng, Shuicheng Yan, and Xinchao Wang. MetaFormer baselines for vision. 46(2):896–912.

[11] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. RandAugment: Practical automated data augmentation with a reduced search space.

[12] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization.

[13] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. MetaFormer is actually what you need for vision.

[14] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. version: 2.

[15] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. FractalNet: Ultra-deep neural networks without residuals. version: 4.