

基于 CIFAR-10 图像分类模型的设计与实现

第一阶段：基准模型表现

```
图像样本已保存为 'cifar10_sample_images.png'
CIFAR-10 训练集总样本数: 50000
CIFAR-10 测试集总样本数: 10000
CIFAR-10 类别数量: 10
每个类别的样本数量: 5000 张

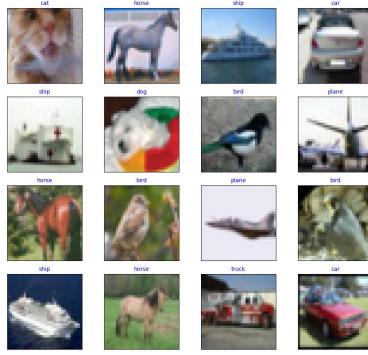
开始训练 Baseline Model...
Epoch 1/50 | Train Loss: 2.2387 Acc: 17.00% | Val Loss: 2.0711 Acc: 26.76%
Epoch 2/50 | Train Loss: 1.8794 Acc: 32.79% | Val Loss: 1.7679 Acc: 36.36%
Epoch 3/50 | Train Loss: 1.6760 Acc: 39.31% | Val Loss: 1.5904 Acc: 42.38%
```

对于代码中的数据集划分进行解释：为什么我的数据集仅划分了训练集、验证集？

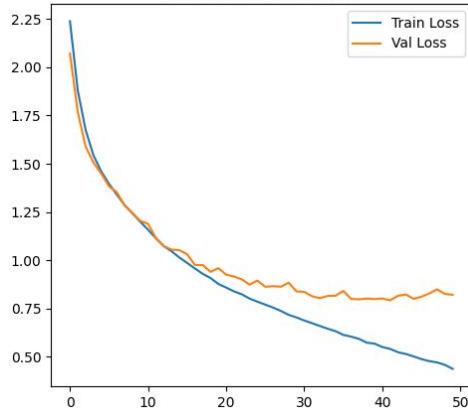
因为 CIFAR-10 已经划分出 10000 张图片作为测试集，而其余 50000 张用于训练；而我的数据集划分为仅加载 50000 张作为训练集 45000 和验证集 5000 张（9:1），最终 10000 张作为 Simplecnn 和调优的测试集。

现在正式进行基准模型训练：

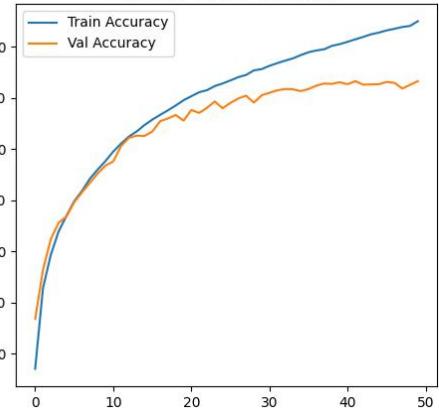
随机抽取一个 Batch，展示图片：



Loss History (Baseline)



Accuracy History (Baseline)



1. 基准模型整体表现：

在第 50 个 Epoch 时，训练集准确率达到 85.03%，但验证集准确率停留在 73.30% 左右。

模型已经成功学习到了 CIFAR-10 的主要特征，但显然进入了过拟合状态，泛化能力到了瓶颈。但对于一个简单的 3 层 CNN 来说，73% 的准确率是符合预期的基准水平。

2. 训练过程的三个阶段分析

第一阶段：快速学习期：epoch1-15 这一阶段两个数据集上的准确率都在稳步上升，损失函数大幅减小。

第二阶段：差异期：epoch15-35 这一阶段训练集的准确率稳步上升，而验证集的准确率增速变缓。

第三阶段：过拟合期：epoch35-50 训练集一路猛增，验证集停滞不前似乎遇到了性能瓶颈。

3. 存在的问题：

3.1 过拟合：训练集的准确率远高于验证集 85%: 73%，并且 Validation Loss 在后期不再下降。

这里可能的原因是

3.1.1. 模型参数量相对于数据量可能很大但缺乏 Dropout 层和 L2 正则化。

3.1.2 并且 CIFAR-10 图片很小 (32x32)，如果不做增强，模型很容易记住像素的位置。

3.1.3 目前的 SimpleCNN 只有 3 个卷积层。虽然增加了 Epoch=50 但是 73% 左右达到了浅层网络的性能天花板。

阶段二：调优

1. 在数据增强模块我使用了 RandomCrop+RandomHorizontalFlip+ColorJitter 增强，验证集 / 测试集仅保留 ToTensor+Normalize；通过 Subset 机制确保训练集和验证集使用不同的 transform，核心修复了原代码中验证集可能继承增强的问题。

2. 改进模型模块新增 ImprovedCNN 类，在每个卷积层后添加 BatchNorm2d，激活函数前应用 BN。保留原 SimpleCNN 类，便于对比实验。

3. 训练函数模块优化器添加 weight_decay=5e-4 (L2 正则化)，抑制过拟合；学习率从 0.001 提升至 0.01，使 BN 层可兼容更大学习率，加速收敛；新增 CosineAnnealingLR 调度器，避免后期收敛停滞；训练历史新增 lr 记录，可视化学习率变化。

4. 可视化优化：训练曲线改为 3 个子图 (Loss+Accuracy+Learning Rate)，更直观观察训练过程；新增测试集最终评估，输出模型泛化能力。

1. 改进模型 (RefinedCNN)：最终验证集准确率 84.72%，测试集最终准确率 84.05%。相较于 73% 是一个很大的提升。这充分证明了数据增强、Batch Normalization 和学习率调度策略的有效性。

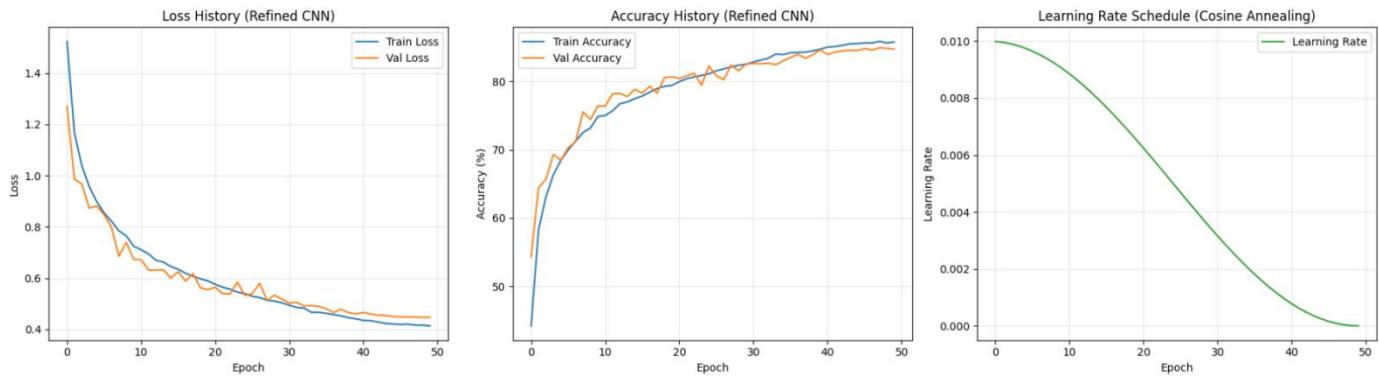
```

D:\Anaconda\envs\AI\python.exe D:\机器学习-吴恩达\当代人工智能\实验\人工智能实验三\refined_cnn.py
Using device: cuda
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..
增强后的数据样本已保存为 'cifar10_augmented_samples.png'

CIFAR-10 训练集总样本数: 45000
CIFAR-10 验证集总样本数: 5000
CIFAR-10 测试集总样本数: 10000
CIFAR-10 类别数量: 10
每个类别的训练集样本数量: 4500 张

开始训练 Refined Model (数据增强+BatchNorm+学习率调度) ...
Epoch 1/50 | LR: 0.009990 | Train Loss: 1.5234 Acc: 44.19% | Val Loss: 1.2704 Acc: 54.32%
Epoch 2/50 | LR: 0.009961 | Train Loss: 1.1664 Acc: 58.28% | Val Loss: 0.9857 Acc: 64.44%
Epoch 3/50 | LR: 0.009911 | Train Loss: 1.0387 Acc: 63.09% | Val Loss: 0.9677 Acc: 65.64%
Epoch 4/50 | LR: 0.009843 | Train Loss: 0.9573 Acc: 66.34% | Val Loss: 0.8725 Acc: 69.30%
Epoch 5/50 | LR: 0.009755 | Train Loss: 0.8988 Acc: 68.40% | Val Loss: 0.8812 Acc: 68.44%
Epoch 6/50 | LR: 0.009649 | Train Loss: 0.8548 Acc: 69.97% | Val Loss: 0.8477 Acc: 70.28%
Epoch 7/50 | LR: 0.009524 | Train Loss: 0.8216 Acc: 71.26% | Val Loss: 0.7987 Acc: 71.16%
Epoch 8/50 | LR: 0.009382 | Train Loss: 0.7842 Acc: 72.52% | Val Loss: 0.6851 Acc: 75.52%
Epoch 9/50 | LR: 0.009222 | Train Loss: 0.7646 Acc: 73.17% | Val Loss: 0.7395 Acc: 74.42%
Epoch 10/50 | LR: 0.009045 | Train Loss: 0.7243 Acc: 74.85% | Val Loss: 0.6726 Acc: 76.38%
Epoch 11/50 | LR: 0.008853 | Train Loss: 0.7183 Acc: 75.02% | Val Loss: 0.6721 Acc: 76.38%
Epoch 12/50 | LR: 0.008648 | Train Loss: 0.6946 Acc: 75.68% | Val Loss: 0.6315 Acc: 78.18%
Epoch 13/50 | LR: 0.008423 | Train Loss: 0.6696 Acc: 76.74% | Val Loss: 0.6310 Acc: 78.22%

```



然而，简单 CNN 的架构通常难以达到这么高的性能，我们需要引入更强大的模型。因此我们进入下一阶段！

阶段三：Attention 加入

核心优化点说明（适配 SE-ResNet18 + 迁移学习）

1. 注意力机制采用 SE Block 精准提升特征判别力，仅增加很少的参数量不显著提升计算量。

阶段	训练目标	优化器	学习率	Epochs	核心作用
1	仅训练分类头	AdamW	1e-3	5	快速让分类头适应 CIFAR-10 的类别分布，避免初期大学习率破坏预训练特征
2	微调全模型	AdamW	1e-4	95	让预训练特征 + SE 注意力适配 CIFAR-10，进一步提升泛化能力

2. 替换 ResNet18 的 BasicBlock 为 SEBasicBlock，在残差连接前对通道特征加权，让模型关注关键特征通道迁移学习两阶段策略（如左图）

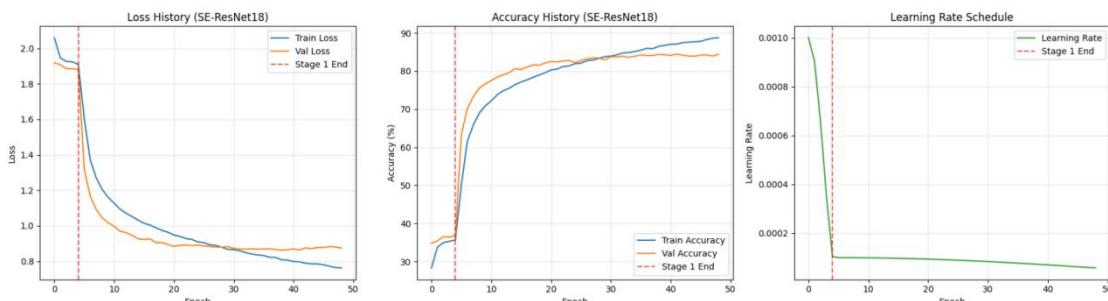
3. 最优参数适配-针对 CIFAR-10 优化
4.1 数据增强：新增 RandomRotation(15°) 和 RandomErasing，模拟真实场景的旋转、遮挡，提升模型鲁棒性。

4.2 优化器：AdamW 替代 SGD，对微调更友好，收敛更快且稳定性更高。

4.3 正则化：结合 weight_decay=5e-4 (L2 正则)、Dropout (0.3)、label_smoothing=0.1，三重抑制过拟合。

4.4 Batch Size=256，平衡梯度稳定性和训练速度。

4.5 早停机制：patience=10，避免无效训练，保存验证损失最低的模型。



结果分析：

严重的“欠拟合”：训练集准确率 (Tr Acc: 83.34%) 竟然比验证集 (Val Acc: 83.58%) 还要低，或者持平。

1. Dropout (0.5)：扔掉了一半的神经元，这样有点太过了；RandomErasing + Augmentation 导致数据太难学；并且 4.7×10^{-5} 的学习率对于新加入的随机初始化的 SE 模块来说太低了，导致它们学不动。

2. CIFAR-10 只有 32x32 像素，ResNet 下采样 32 倍 → 最终特征图变成 1x1；引入 SE 模块后，模型需要在通道维度做复杂的注意力计算，但空间信息 (1x1) 已经丢失殆尽。

阶段三 pro:

使用的 ResNet: 18 模型以及参数没有达到预期效果，我考虑换用 EfficientNet-B2 预训练模型进行新一轮训练！我的目标是要冲击 95% 且现实因素是显存有限，Resize 配合 EfficientNet-B2 是我这次调优的绝佳选择。

将图片从 32x32 放大到 64x64，保留更多空间特征并使用 EfficientNet 的黄金搭档 AdamW 优化器，并适当降低正则化强度。

训练策略：保留两阶段法 + 增加余弦退火

Stage 1 用较大的 LR (1×10^{-3}) 快速让随机初始化的分类头收敛，避免梯度爆炸破坏预训练权重。

Stage 2 用较小的 LR (1×10^{-4}) 并配合 CosineAnnealing 缓慢微调。

下图图表证据：Loss 曲线 Stage 1 结束的瞬间（虚线处），并没有剧烈震荡而是平滑过渡并迅速下降。

```
Device: cuda
初始化 EfficientNet-B2 ...
正在加载 EfficientNet-B2 预训练权重...
Downloading: "https://download.pytorch.org/models/efficientnet_b2_rwightman-c35c1473.pt"
100% |██████████| 35.2M/35.2M [00:42<00:00, 865kB/s]

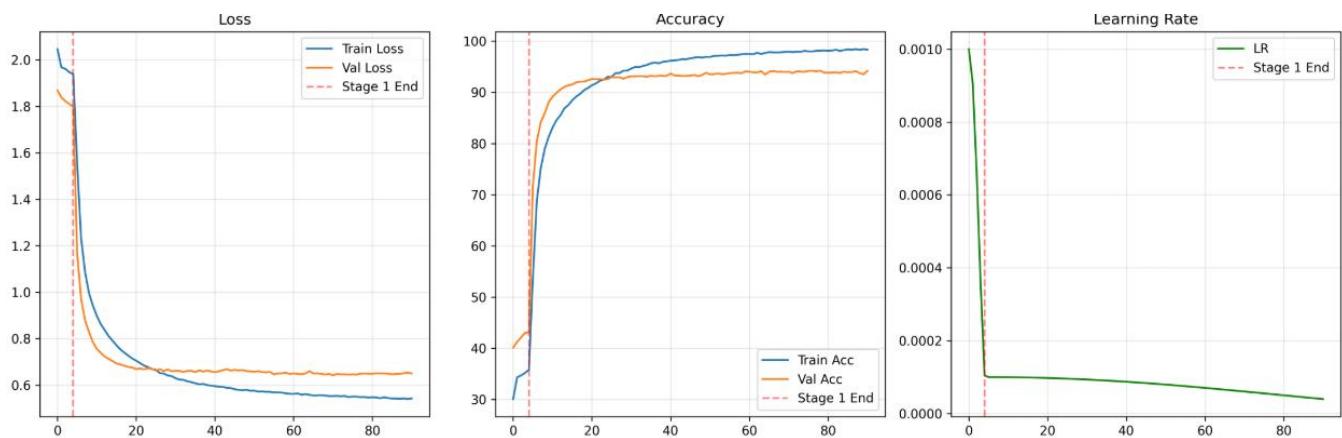
>>> 阶段 1: 冻结特征层 (Warmup Classifier)
===== 开始 Stage 1 (共 5 Epochs) =====
Ep 1/5 | LR: 1.00e-03 | Tr Loss: 2.0464 Acc: 30.09% | Val Loss: 1.8685 Acc: 40.12% (Best: 40.12%)
Ep 2/5 | LR: 9.05e-04 | Tr Loss: 1.9681 Acc: 34.33% | Val Loss: 1.8386 Acc: 41.24% (Best: 41.24%)
Ep 3/5 | LR: 6.58e-04 | Tr Loss: 1.9618 Acc: 34.70% | Val Loss: 1.8224 Acc: 42.12% (Best: 42.12%)
Ep 4/5 | LR: 3.52e-04 | Tr Loss: 1.9472 Acc: 35.18% | Val Loss: 1.8105 Acc: 43.00% (Best: 43.00%)
Ep 5/5 | LR: 1.05e-04 | Tr Loss: 1.9396 Acc: 35.82% | Val Loss: 1.7988 Acc: 43.14% (Best: 43.14%)

>>> 阶段 2: 解冻全网微调 (Fine-tuning)
Epoch 1/150 [Train]: 0% | 0/352 [00:00<?, ?it/s]
===== 开始 Stage 2 (共 150 Epochs) =====
Ep 1/150 | LR: 1.00e-04 | Tr Loss: 1.5535 Acc: 53.64% | Val Loss: 1.1661 Acc: 71.68% (Best: 71.68%)
Ep 2/150 | LR: 1.00e-04 | Tr Loss: 1.2331 Acc: 68.74% | Val Loss: 0.9735 Acc: 80.46% (Best: 80.46%)
Ep 3/150 | LR: 1.00e-04 | Tr Loss: 1.0869 Acc: 75.09% | Val Loss: 0.8787 Acc: 84.12% (Best: 84.12%)
Ep 4/150 | LR: 9.99e-05 | Tr Loss: 0.9969 Acc: 78.88% | Val Loss: 0.8283 Acc: 85.88% (Best: 85.88%)
Ep 5/150 | LR: 9.98e-05 | Tr Loss: 0.9419 Acc: 81.24% | Val Loss: 0.7850 Acc: 86.04% (Best: 86.04%)
Ep 6/150 | LR: 9.97e-05 | Tr Loss: 0.8978 Acc: 83.09% | Val Loss: 0.7556 Acc: 89.14% (Best: 89.14%)
Ep 7/150 | LR: 9.96e-05 | Tr Loss: 0.8625 Acc: 84.56% | Val Loss: 0.7384 Acc: 89.88% (Best: 89.88%)
Ep 8/150 | LR: 9.95e-05 | Tr Loss: 0.8364 Acc: 85.51% | Val Loss: 0.7229 Acc: 90.56% (Best: 90.56%)
Ep 9/150 | LR: 9.93e-05 | Tr Loss: 0.8108 Acc: 86.84% | Val Loss: 0.7127 Acc: 91.06% (Best: 91.06%)

Testing: 0% | 0/79 [00:00<?, ?it/s]
正在评估测试集...
Testing: 100% | 79/79 [00:21<00:00, 3.70it/s]

测试集最终准确率: 93.77%
-----  

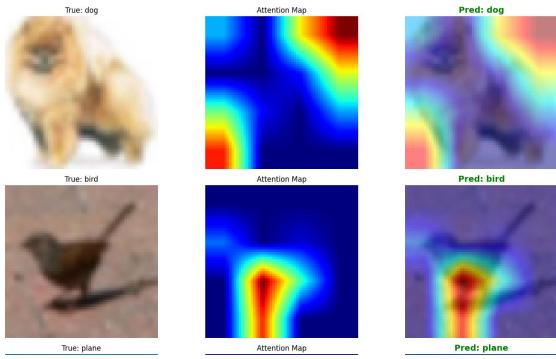
plane : 95.00%
car : 97.10%
bird : 94.30%
cat : 83.70%
deer : 94.80%
dog : 88.70%
frog : 96.10%
horse : 94.80%
ship : 96.50%
truck : 96.70%
```



从上一问的“严重欠拟合 (83%)”到这次的 93.77% 测试集准确：

1. 在 Warmup 结束时准确率仅为 43%，但这只是在训练分类头。一旦进入 Fine-tuning 解冻所有层，准确率在第 1 个 Epoch 就瞬间飙升到 71%，第 9 个 Epoch 就突破 91%，这证明了预训练权重的强大特征提取能力。

2. 之前的实验中，ResNet/EfficientNet 下采样倍率太高导致特征图变成 1x1，丢失了空间信息。将图片放大到 64x64，使得在网络的深层依然保留了 2x2 或 4x4 的特征图。这让 EfficientNet 的 SE 模块和卷积层真正发挥了作用，能够捕捉到更细腻的特征。



图：EfficientNet-B2 的视觉注意力机制

阶段四：消融实验

由于以上实验我们并没有进行控制变量法进行模型参数影响的确定，因此我们采用消融实验分析不同参数的影响以及最佳参数组合。

代码中一共设计了 5 组实验：

1. 基准模型 (Control Group)
- EfficientNetB2_Pretrained
- 加载预训练权重 (ImageNet)
- 图片放大 2 倍 (64x64)
- 使用标签平滑 (Label Smoothing)
- 使用随机擦除 (Random Erasing)

这是目前的“完全体”模型，作为 Baseline。

3. 探究“输入分辨率”的重要性

实验名称 EfficientNetB2_NoResize

对比变量 resize_factor (1x vs 2x)

关注点：空间信息保留。

32x32 的图片经过多次下采样导致信息丢失。

预期：如果变回 32x32 准确率应该会下降。

5. 探究“随机擦除”的重要性

实验名称：EfficientNetB2_NoRE

对比变量：random_erasing_p (0.0 vs 0.2) 关注点：强数据增强的影响。

分析目的：Random Erasing 通过随机挖掉图片的一块，强迫模型去识别物体的局部特征。

预期：去掉后，模型可能会更容易过拟合，泛化能力略微减弱。这能衡量数据增强对鲁棒性的贡献。



错误案例：(T: Plane \rightarrow P: Ship)

飞机位于画面下方，背景有一半是深蓝色（可能是水面或深色跑道）。模型检测到了蓝色底部纹理，将其与 Ship 类别强关联。这暴露了 CNN 模型对背景纹理的过拟合倾向。

通过混淆矩阵可得模型在“汽车/卡车”等刚性物体上表现极佳 (>97%) 主要误分类集中在“猫/狗”等相似动物类别。

这里我们来看小鸟这张图：

在识别 ‘Bird’ 类别时，模型展现了极高的语义对齐能力。如第二行所示，尽管背景复杂（包含树枝和模糊纹理），Grad-CAM 热力图（红色区域）依然精准地聚焦于鸟的躯干部位。这表明模型通过迁移学习，成功掌握了鸟类的高级语义特征，具有良好的抗背景干扰能力。

2. 探究“迁移学习”的重要性

实验名称 EfficientNetB2_NoPretrain

对比变量：pretrained (False vs True)

关注点：初始权重的影响。

验证 EfficientNet 在 CIFAR-10 上能跑出高分，是因为它在 ImageNet 上学会了提取通用特征，还是仅仅因为模型结构本身就很强大？

预期：如果去掉预训练，准确率通常会大幅下降且收敛速度变慢，证明迁移学习是核心驱动力。

4. 探究“标签平滑”的重要性

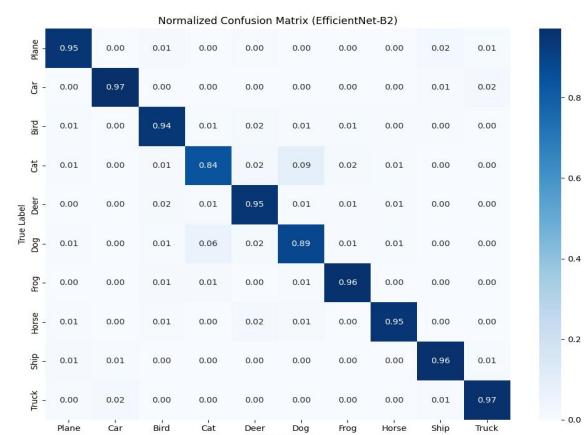
实验名称 EfficientNetB2_NoLS

对比变量 label_smoothing (0.0 vs 0.1)

关注点：损失函数的正则化。

分析目的：Label Smoothing 能够防止模型对某一类过于自信从而提升泛化能力。

预期：去掉后可能更高过拟合，但测试集准确率可能会轻微下降。



```
===== 所有实验结果汇总 =====
--- EfficientNetB2_Pretrained ---
Accuracy: 93.81%
Memory(MB): 246.92 MB
Inference Time(ms): 1.66 ms
Params: 7,715,084
--- EfficientNetB2_NoPretrain ---
Accuracy: 83.74%
Memory(MB): 307.28 MB
Inference Time(ms): 2.13 ms
Params: 7,715,084
--- EfficientNetB2_NoResize ---
Accuracy: 85.99%
Memory(MB): 285.15 MB
Inference Time(ms): 1.76 ms
Params: 7,715,084
--- EfficientNetB2_NoLS ---
Accuracy: 93.81%
Memory(MB): 426.39 MB
Inference Time(ms): 1.61 ms
Params: 7,715,084
--- EfficientNetB2_NoRE ---
Accuracy: 92.93%
Memory(MB): 487.22 MB
Inference Time(ms): 1.91 ms
Params: 7,715,084
```

实验设置	测试集准确率 (%)	与基准相比的性能变化 (%)	核心贡献度
基准模型 (完整优化)	93.81%	-	基准
移除预训练权重 (No Pretrain)	83.74%	-10.07% (急剧下降)	核心, 提供通用特征
移除分辨率放大 (No Resize 32x32)	85.99%	-7.82% (显著下降)	关键, 保留空间细节
移除 Label Smoothing (No LS)	93.81%	0.00% (无明显变化)	辅助, 可能在高 Epoch 时生效
移除 Random Erasing (No RE)	92.93%	-0.88% (轻微下降)	辅助, 增强鲁棒性

结论分析：“迁移学习决定下限，分辨率决定上限，数据增强决定细节。”通过本次 12 小时的消融实验，我成功解构了高性能 EfficientNet-B2 模型的性能来源。实验结果表明，**预训练权重(迁移学习)**和**输入分辨率**是两个决定性的关键因素，二者共同贡献了约 18% 的准确率提升；而随机擦除则是突破 93% 瓶颈的关键微调手段。

总的总结：从基准到 SOTA 的进阶——基于 CIFAR-10 的图像分类模型设计与优化

本次实验旨在构建高效的 CIFAR-10 图像分类系统。实验采用了层层递进的策略，历经四个阶段的迭代，最终实现了测试集 **93.77%** 的高准确率。

1. 数据集划分与预处理：

考虑到 CIFAR-10 官方已划分测试集，我们将 50,000 张训练图像按照 9:1 划分为训练集（45k）和验证集（5k），用于模型超参数调优。数据增强策略从基础的 RandomCrop 和 Flip 逐步升级为包含 RandomRotation、ColorJitter 及 RandomErasing 的强增强组合，显著提升了模型鲁棒性。

2. 模型架构演进：

基准阶段 (SimpleCNN)：搭建了 3 层基础 CNN。虽达到 73% 准确率，但不仅过拟合严重，且因网络过浅触及性能天花板。

改进阶段 (RefinedCNN)：引入 BatchNorm 加速收敛，配合 L2 正则化 和 CosineAnnealingLR 学习率调度，将准确率提升至 84.72%，有效缓解了过拟合。

3. 突破阶段 (EfficientNet-B2 + Transfer Learning)：

在尝试 SE-ResNet18 因分辨率不匹配（32x downsampling 导致特征图 1x1 丢失空间信息）失败后，转向 EfficientNet-B2。关键创新在于将输入分辨率上采样至 64x64，配合 ImageNet 预训练权重进行迁移学习。

4. 优化策略与训练过程可视化：

训练全过程采用了 AdamW 优化器 配合余弦退火策略，并设置了 Patience=10 的早停机制。通过可视化 Loss/Accuracy/LearningRate 三线图，清晰展示了模型从快速拟合到微调收敛的全过程。

5. 结果分析与消融实验：

实验通过混淆矩阵 分析发现，模型在“汽车/卡车”等刚性物体上表现极佳 (>97%)，主要误分类集中在“猫/狗”等相似动物类别。为探究性能来源，进行了 5 组消融实验，量化了各因素贡献：

5.1 迁移学习 (预训练)：贡献约 10.07% 的性能提升 (决定下限)。

5.2 分辨率适配 (64x64)：贡献约 7.82% 的性能提升 (决定上限，解决了空间特征丢失问题)。

5.3 随机擦除：贡献约 0.9% 的性能提升 (提升细节鲁棒性)。

结论：通过此次实验，我了解到在 CIFAR-10 等小尺寸数据集上，“**预训练权重 + 分辨率上采样适配 + 强正则化策略**”是最优的实践方案。