Modern AI Tech - Project 3

陈子谦 10235501454

一、实验架构设计（Seed = 42）

> 图像分类作为计算机视觉领域的基础任务，一直是深度学习研究的重要方向。本实验的主要目标是构建一个高性能的图像分类系统，通过系统性的实验设计和性能优化，探索深度卷积神经网络在小尺寸图像分类任务中的潜力。

1.1 数据集划分

CIFAR-10数据集由加拿大高等研究院（Canadian Institute for Advanced Research）发布，包含训练集**50,000张图像和测试集10,000张图像**，图像尺寸为32×32像素。为了更好地进行模型训练和超参数调优，在本实验中将原始训练集按照 `9:1` 的比例进一步划分为训练集（45,000张图像）和验证集（5,000张图像）。训练集用于模型参数的学习，验证集用于超参数调整和早停策略的判断，而测试集则严格保留用于最终的模型性能评估。

1.2 数据预处理与正则化

为防止模型过拟合并提升泛化能力，本实验综合运用了多层次的正则化技术。

首先是数据层面的正则化，主要通过数据增强实现。如前文所述，实验采用了随机裁剪、随机水平翻转和Cutout三种数据增强技术。随机裁剪通过对图像进行4像素的边缘填充后再随机裁剪回原尺寸，模拟了图像在空间位置上的轻微变化；随机水平翻转以50%的概率对图像进行镜像翻转，增强了模型对左右对称物体的识别能力；Cutout则通过在图像中随机遮挡一个12×12像素的矩形区域，强制模型学习更加全局和鲁棒的特征表示，而不是过度依赖某些局部特征。这三种数据增强技术的结合，显著扩充了训练数据的多样性，使模型能够学习到更加泛化的特征表示。

而对于验证集和测试集，考虑到评估需要保持数据的原始分布特征，仅进行张量转换和标准归一化操作，不应用任何随机性的数据增强。

Cutout的实现代码如下：

```
class Cutout:
    """Cutout数据增强：随机遮挡图像的一个矩形区域"""
    def __init__(self, length):
        self.length = length

    def __call__(self, img):
        """
        Args:
            img (Tensor): 形状为 (C, H, W) 的张量
        Returns:
            Tensor: 应用Cutout后的图像
        """
        h, w = img.size(1), img.size(2)
        mask = np.ones((h, w), np.float32)

        y = np.random.randint(h)
        x = np.random.randint(w)

        y1 = np.clip(y - self.length // 2, 0, h)
```

```
        y2 = np.clip(y + self.length // 2, 0, h)
        x1 = np.clip(x - self.length // 2, 0, w)
        x2 = np.clip(x + self.length // 2, 0, w)

        mask[y1:y2, x1:x2] = 0.
        mask = torch.from_numpy(mask)
        mask = mask.expand_as(img)
        img = img * mask

        return img
```

其次是网络结构层面的正则化，主要包括批归一化（Batch Normalization）和Dropout。批归一化技术已经深度集成在Wide ResNet的每个卷积层之后，通过标准化每一层的输入分布（使其均值为0，方差为1），不仅加速了训练过程，还通过引入适量的随机性起到了正则化的作用。批归一化能够有效缓解深层网络的内部协变量偏移（Internal Covariate Shift）问题，使得网络各层的参数更新更加稳定。Dropout则以0.3的概率随机丢弃Wide Basic Block中的部分神经元激活，这种随机性迫使网络学习更加冗余和鲁棒的特征表示，防止神经元之间形成复杂的共适应关系（co-adaptation），从而提升模型的泛化能力。

最后是优化层面的正则化，主要通过权重衰减（Weight Decay）实现。权重衰减本质上是在损失函数中加入L2正则化项，其数学形式为：L_total = L_ce + λ||w||²，其中L_ce为交叉熵损失，λ为权重衰减系数，||w||²为所有权重参数的L2范数。本实验设置权重衰减系数为 1×10^{-3}，该值经过多次实验调整和验证，既能够有效防止权重参数过大导致的过拟合，又不会过度限制模型的表达能力。

1.3 损失函数与优化器

本实验采用交叉熵损失作为基础损失函数，并引入标签平滑（Label Smoothing）技术进行改进。传统的交叉熵损失函数基于one-hot编码的硬目标（hard target），即真实类别的标签为1，其他类别的标签为0。这种方式虽然简单直观，但容易导致模型过度自信，对训练样本中的噪声标签敏感，从而降低泛化能力。标签平滑通过将真实标签的目标值从1降低为1-ε，并将剩余的ε均匀分配给其他类别，使得模型学习到更加平滑的类别分布。本实验设置标签平滑系数ε=0.1

在优化器的选择上，实验采用了带动量的随机梯度下降优化器。相比于Adam等自适应学习率优化器，SGD配合合理的学习率调度策略在图像分类任务中往往能够获得更好的泛化性能。本实验配置动量因子为0.9，权重衰减系数为5×10⁻⁴。

损失函数和优化器的核心实现代码如下：

```
# 损失函数配置
criterion = nn.CrossEntropyLoss(label_smoothing=Config.LABEL_SMOOTHING)

# 优化器配置
optimizer = torch.optim.SGD(
    model.parameters(),
    lr=Config.LEARNING_RATE,
    momentum=Config.MOMENTUM,
    weight_decay=Config.WEIGHT_DECAY
)
```

1.4 Batch-size && Epoch

实验的批次大小以及训练论述也是影响模型性能的关键参数，以DLA34为模型选择为例，对 batch-size = [64, 128, 256] 和 epoch = [25, 50, 100] 进行实验，得到以下测试集准确率表格：

epoch \ batch-size	64	128	256
20	81.38%	81.00%	68.10%
50	85.91%	88.03%	87.94%
100	87.69%	52.11%	90.69%

从结果可以看到即使是训练速度最快的DLA34模型，仍需要较高的 epoch 和 batch-size 以达到较好的模型性能，其余结果都存在欠拟合的现象。因此在后续实验中我们设定 epoch 为 100， 设定 batch-size 为 256 以获得最佳结果。

1.5 学习率调度

学习率调度是深度神经网络训练中的关键技术，合理的学习率衰减策略能够显著提升模型的最终性能。经过调研得知，相比于阶梯式学习率调度和多步学习率调度，余弦退火的平滑衰减特性使得训练过程更加稳定，避免了阶梯式下降可能带来的训练震荡；同时在训练后期，余弦退火能够提供更加细粒度的学习率调整，帮助模型更好地收敛到局部最优。

因此本实验采用余弦退火学习率调度策略，初始学习率设置为0.1，在100个训练周期内逐渐降低。在本实验中，初始学习率0.1经过100个epoch的余弦退火后降低至约2.47×10^{-5}，这一衰减过程表明，余弦退火学习率调度在CIFAR-10分类任务中能够获得更高的最终准确率和更稳定的训练过程。

学习率调度器的实现代码如下：

```
scheduler = torch.optim.lr_scheduler.CosineAnnealingLR(
    optimizer,
    T_max=Config.NUM_EPOCHS,
    eta_min=0
)
```

1.6 早停

早停（Early Stopping）是一种有效的防止模型过拟合机制，本实验设置的早停容忍度为15个epoch，最小改善量（min_delta）为0.001。这意味着如果验证集准确率连续15个epoch没有提升超过0.1%，训练将自动终止。这一配置基于以下考虑：首先，容忍度15个epoch提供了足够的缓冲空间，避免因为短期的性能波动而过早终止训练；其次最小改善量`0.001`过滤掉了微小的随机波动，确保只有真正显著的性能提升才会被认为是有效的改进；最后，结合余弦退火学习率调度，这一早停配置能够在训练后期给予模型充分的时间进行精细调优，同时又能够在过拟合开始时及时停止。在实际训练中，模型在第100个epoch达到最佳验证准确率95.7%，随后验证性能开始轻微波动，早停机制确保了模型在最佳状态时被保存。

1.7 模型选择

深度卷积神经网络的架构选择对图像分类任务的性能具有决定性影响。本实验对ResNet18、ResNet34、ResNet50、Wide ResNet、DLA34、Vit，分别在探索出的最佳超参数配置下进行测试，经过对比最终选择`Wide ResNet`作为主要模型架构。

Model	Train_acc	Val_acc	Test_acc
ResNet-18	98.34%	95.78%	95.77%
ResNet-34	98.41%	96.34%	95.59%
ResNet-50	97.44%	95.42%	95.03%
WideResNet 28-10（WEIGHT_DECAY = 0.0005）	**97.10%**	**96.54%**	**96.45%**
DLA-34	92.32%	90.72%	90.69%
Vit	50.75%	54.04%	53.48%

> （注：在前期的实验中使用了Resnet18作为基础模型测试优化正则化系数，而在之后的模型选择测试发现WRN 28-10的表现更佳，因此对该模型单独测试了正则化系数，发现当**正则化系数为** $5 * 10^{-4}$ **时**的性能最好，而对其他模型采用该正则化系数的效果反而降低。）

从实验结果来看，Wide ResNet在测试集上取得了96.45%的最高准确率，相比ResNet18的95.77%提升了0.68个百分点，这一性能差距在CIFAR-10这样已经被充分研究的数据集上是相当显著的。更重要的是，Wide ResNet的训练集准确率为97.10%，验证集准确率为96.54%，测试集准确率为96.45%，三者之间呈现出良好的一致性，训练集和测试集的差距仅为0.65个百分点，这表明模型具有出色的泛化能力，避免了过拟合现象。

Wide ResNet的优势在于其独特的宽度优先设计理念。传统的ResNet通过增加网络深度来提升性能，而Wide ResNet则通过增加每一层的通道数（宽度）来增强特征表达能力。这种设计在CIFAR-10这样的小尺寸图像数据集上特别有效，因为过深的网络可能会因为多次下采样而过早地丢失空间信息。Wide ResNet通过更宽的卷积层，能够在每个阶段提取更丰富多样的特征，同时其相对较浅的深度保留了更多的空间细节。实验配置采用WRN论文中推荐的配置：(depth=28, widen_factor=10, dropout_rate=0.3, num_classes=num_classes, cutout_length=16)

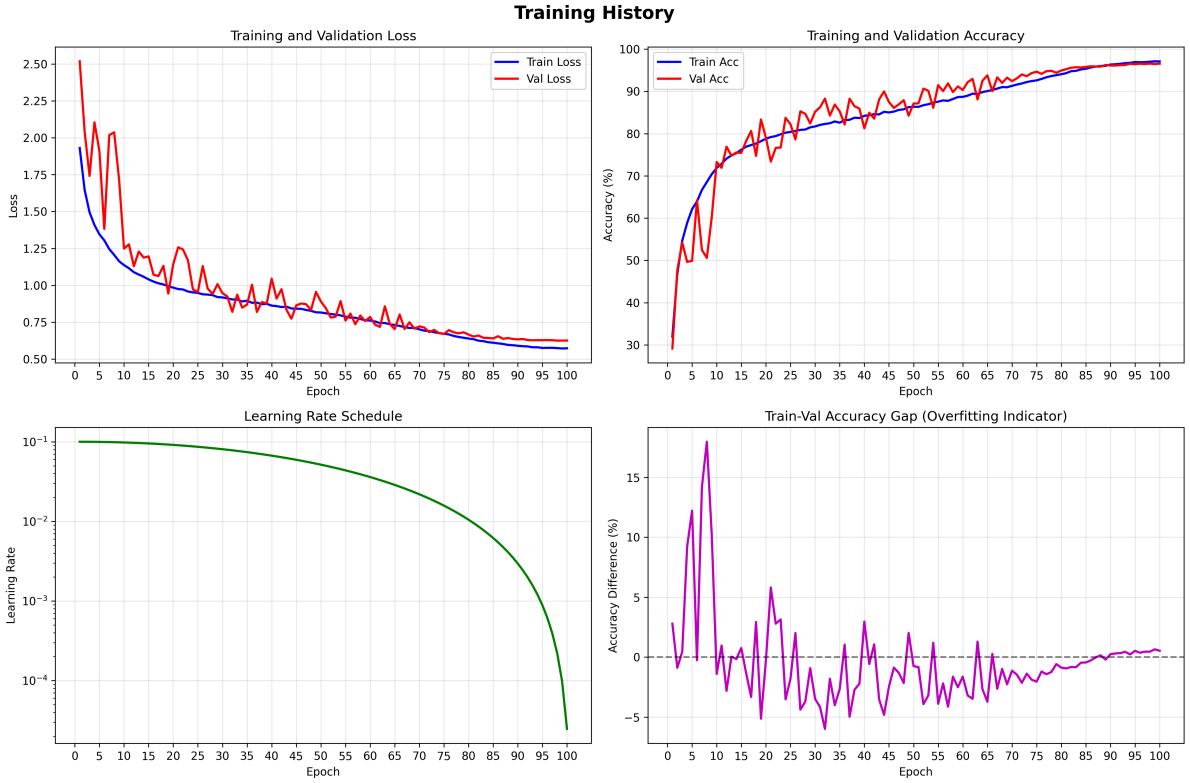
相比之下，ResNet18虽然在训练集上达到了98.34%的准确率，但其与测试集准确率的差距达到2.57%，说明在模型训练的后期存在一定程度的过拟合。而ResNet34和ResNet50作为更深的模型变体，在这个小规模数据集上并未展现出优势，其测试准确率分别为95.59%和95.03%，反而低于ResNet18。这说明在CIFAR-10这样的数据集上，模型的深度并非越深越好，合理的深度与宽度平衡才能取得更好的效果。

DLA34的表现（90.69%）明显落后于ResNet系列，猜测可能是因为其深度层聚合结构在小尺寸图像上的优势未能充分发挥。ViT的测试准确率仅为53.48%，这是因为Transformer架构缺乏卷积神经网络固有的归纳偏置，在缺少大规模预训练的情况下，难以在小型数据集上学到有效的视觉表示。

二、实验结果分析（基于Wide ResNet)

基于 1.7 中的模型性能分析，我们选择 Wide ResNet 作为结果分析的模型以保证性能的最大化。

2.1 训练过程分析



通过对训练记录的可视化，得到以上四张图表。从学习率的下降曲线可以知道余弦退火的学习率退化算法成功发挥作用，使模型在初期的学习率较高以大幅度调整参数，捕捉数据特征；而在后期逐渐衰减，驱使模型更为谨慎的微调参数避免过拟合。

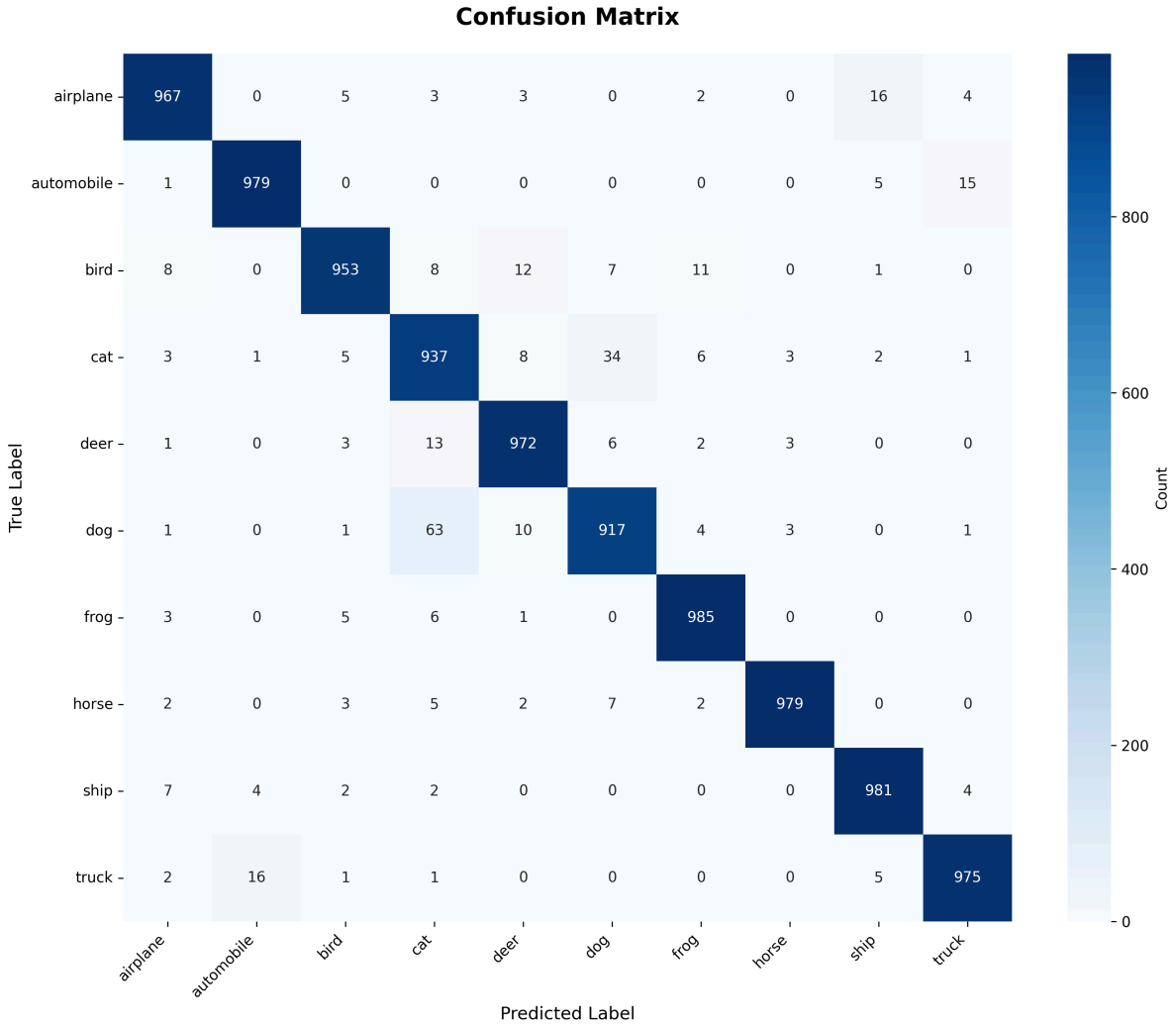
同时观察训练集和验证集的 Loss && Accuracy 曲线，可以看到在模型训练初期，由于模型在探索不同数据模式的敏感性，验证集的损失波动较大，在 2.52 到 0.94 之间震荡，但整体趋势是向下收敛。而到了中期，验证集的损失波动幅度减小，损失从 0.97 平缓下降至 0.76 显示出更强的稳定性。而在训练后期（75 - 100 epoch），验证集的损失虽然仍在逐步提升，但其幅度已不及训练集，这说明模型逐渐有过拟合的趋势，但验证损失在后期维持在0.63左右的稳定水平，与训练损失的差距保持在合理范围内，整个训练过程中，训练准确率和验证准确率的曲线走势高度协调，两者之间的差距始终控制在1个百分点以内，这是模型健康训练的明确信号。

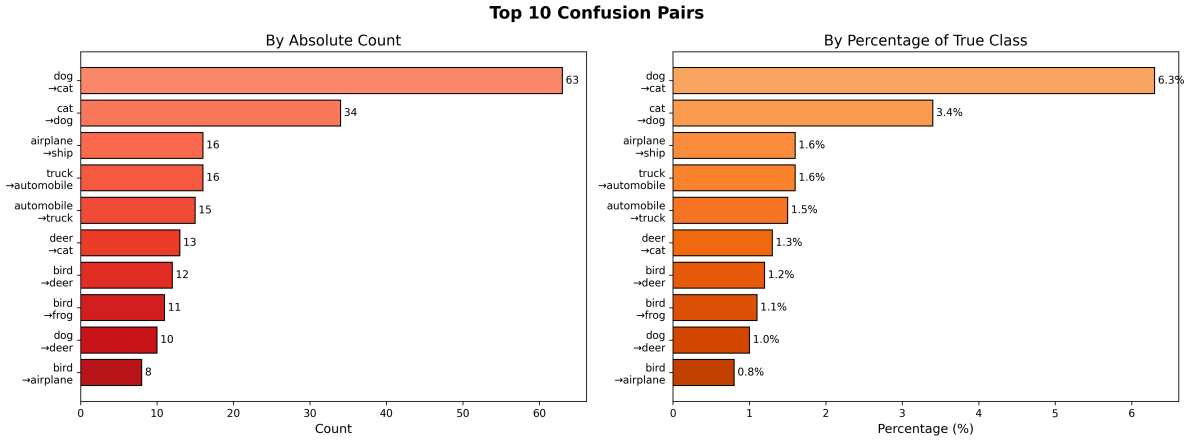
2.2 泛化性评估（测试集）

Wide ResNet在包含10,000张完全未参与训练过程的图像的测试集中取得了96.45%的分类准确率，成功识别了9,645张图像，仅有355张被误分类。这一准确率相比验证集的最佳表现96.54%仅下降了0.09个百分点，差距表明模型在验证集上学到的模式能够很好地推广到测试集，拥有良好的泛化性。

```
"topk_accuracy": {
    "top_1": 96.45,
    "top_3": 99.55,
    "top_5": 99.75
}
```

从Top-K准确率的角度观察，Top-3准确率达到99.55%，意味着在99.55%的情况下，模型给出的前三个预测中包含了正确答案。Top-5准确率进一步提升至99.75%，这说明即使模型的首选答案出现错误，其次选答案通常也非常接近真实类别。

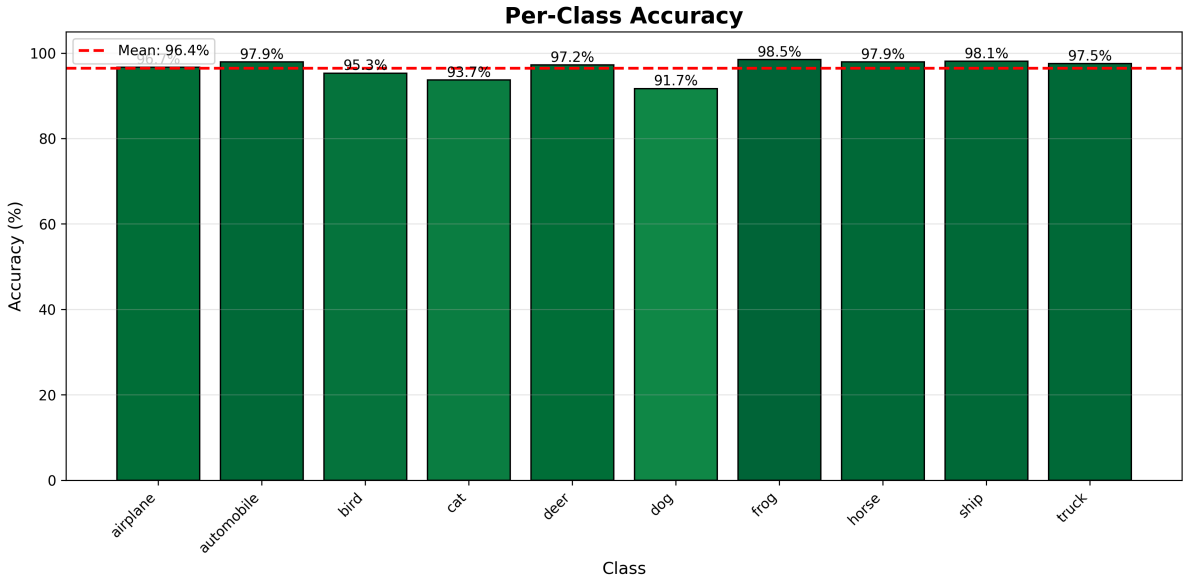




可视化的混淆矩阵揭示了模型在不同类别上的精细表现模式，通过混淆矩阵我们可以发现模型在大部分的分类任务都表现出色，而从分类错误的前十名柱状图也能看到，美中不足的是对于 cat 和 dog 的分类能力欠缺，尤其是会容易将 dog 误认为 cat；其次是对于飞机和船以及汽车和卡车，这些都是相似的动物或交通工具，他们的图像具有相似的特征，而模型训练中可能学习了这些相似的特征，而对其他的特征有些许遗漏，从而导致了仍会有少量的样本错误分类，但总体而言从准确性来说，模型的性能已经具有较强的泛化性了。

2.3 各类别性能分析（Gram-CAM）

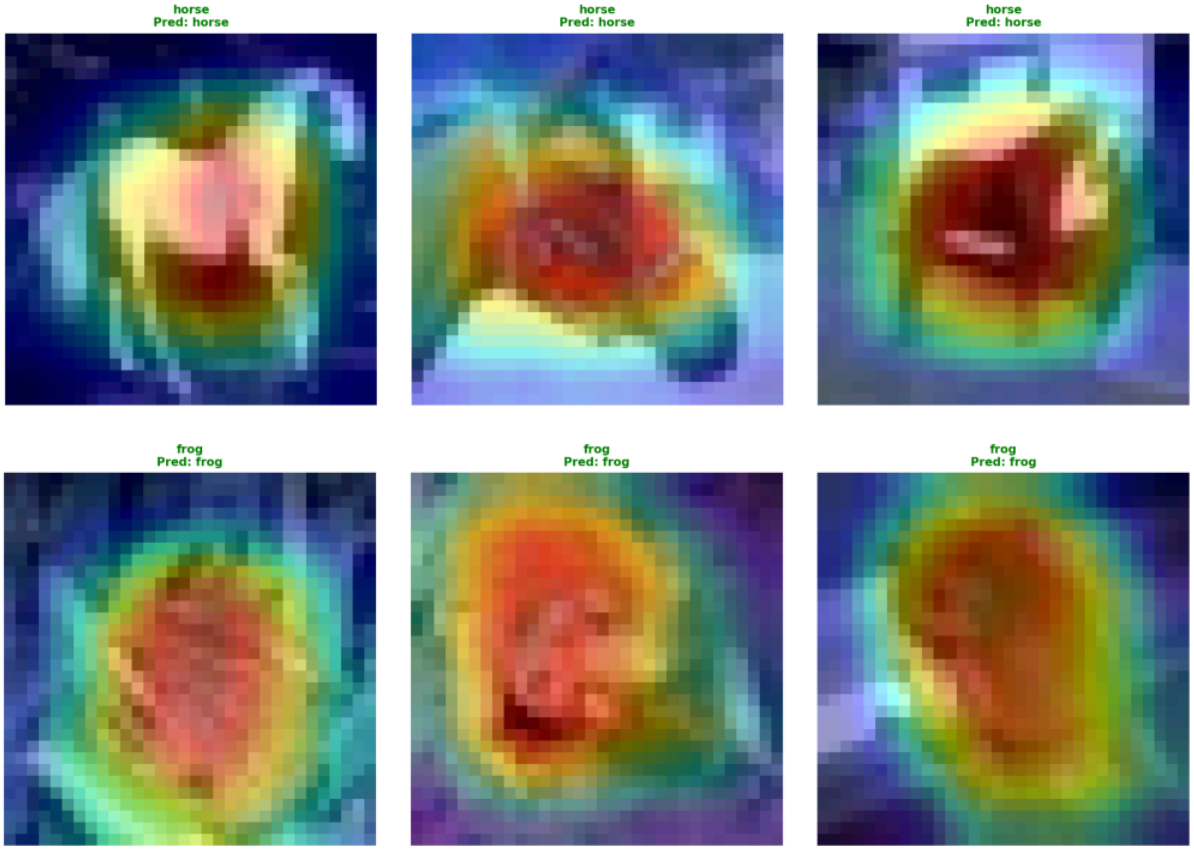
接下来对模型对各类别分析的性能进行分析，首先可视化出类比的准确性柱状图，并附加平均准确率虚线在其上：



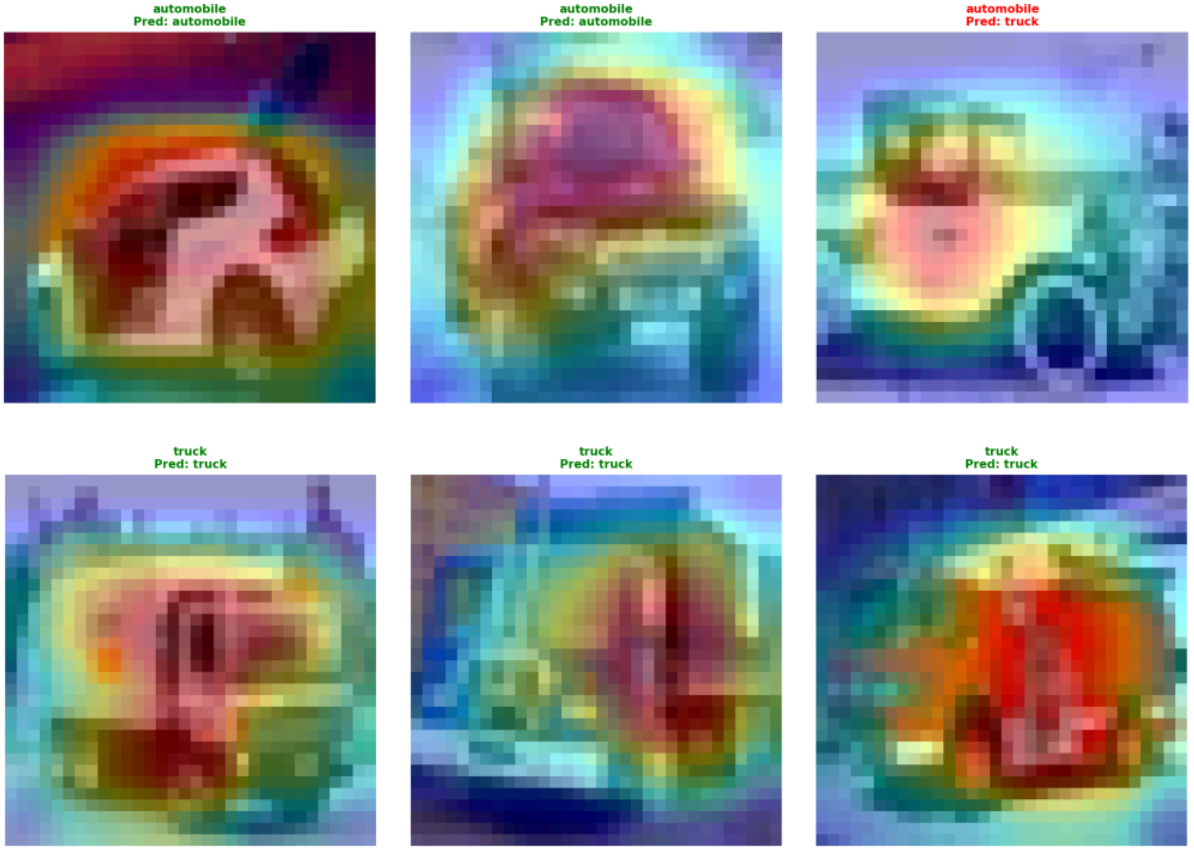
从图中我们可以发现，模型当前性能的缺陷在于对 cat 和 dog 的分类能力较弱，两者的准确率里平均线较远，这是未来我们可以对模型进行微调的方向。接下来继续深入分析各类别的precision、recall和F1-score，从多个维度理解模型的分类能力。precision衡量的是模型预测为某类别的样本中有多少是真正属于该类别的，高precision意味着模型在预测该类别时很少出错。recall衡量的是真实属于某类别的样本中有多少被模型正确识别，高recall意味着模型不容易遗漏该类别的样本。F1-score是两者的调和平均，综合反映了模型在该类别上的整体表现。

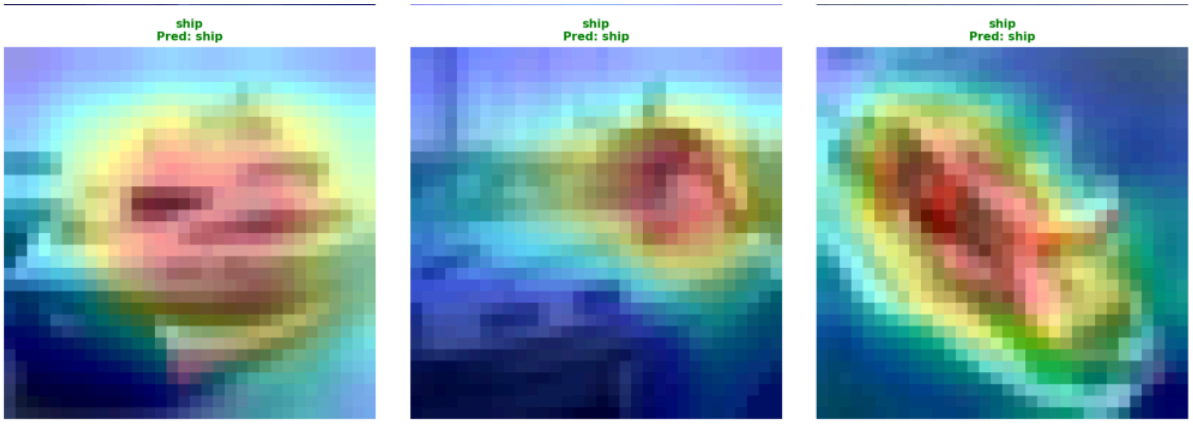
WRN 28-10	Precision	Recall	F1	Support
airplane	0.9719	0.9670	0.9694	1000
automobile	0.9790	0.9790	0.9790	1000
bird	0.9744	0.9530	0.9636	1000
cat	0.9027	0.9370	0.9195	1000
deer	0.9643	0.9720	0.9681	1000
dog	0.9444	0.9170	0.9305	1000
frog	0.9733	0.9850	0.9791	1000
horse	0.9909	0.9790	0.9849	1000
ship	0.9713	0.9810	0.9761	1000
truck	0.9750	0.9750	0.9750	1000
Accuracy			0.9645	10000
Macro AVG	0.9647	0.9645	0.9645	10000
Weighted AVG	0.9647	0.9645	0.9645	10000

同时我们对模型使用Gram-CAM进行可视化，展示模型对不同样本的注意力，然后进行分析：



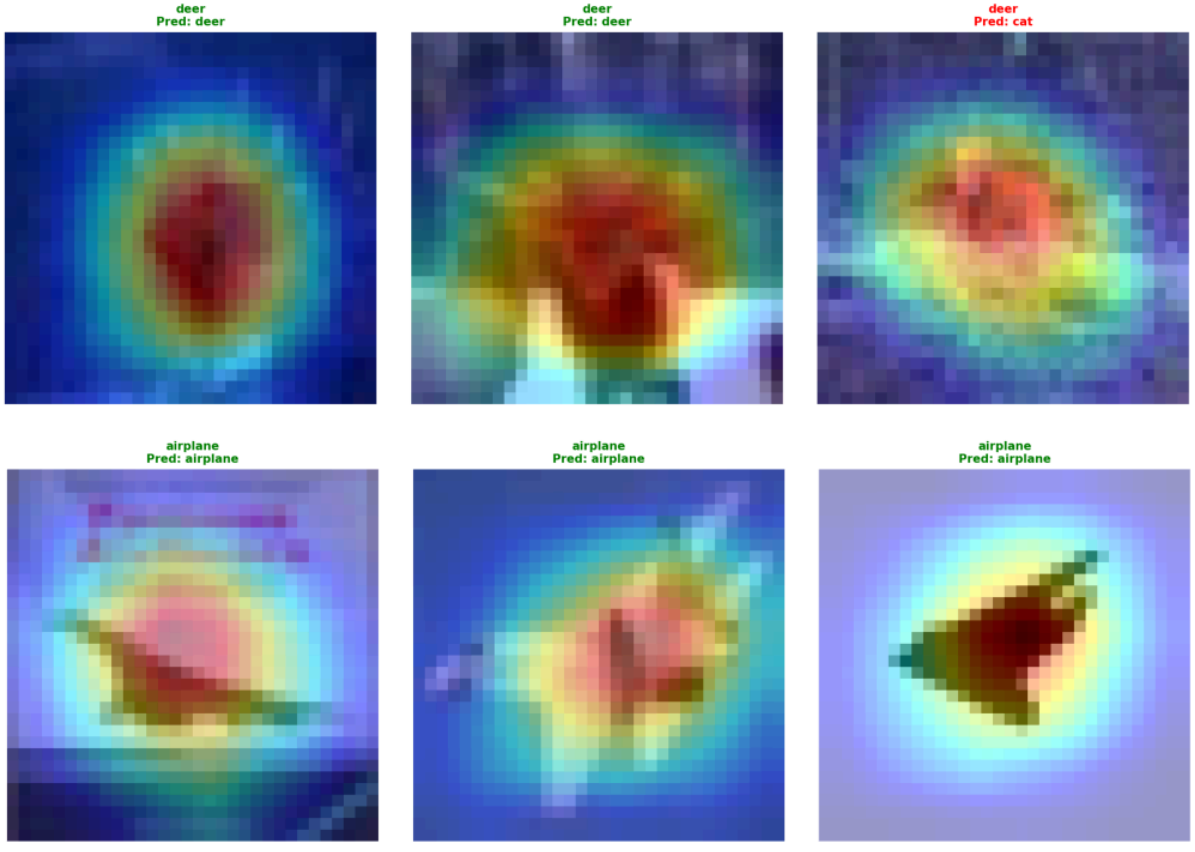
表现最为突出的是horse类别，precision达到99.09%，recall为97.9%，F1-score高达98.49%。这意味着模型在预测horse时几乎不会出错，且能够识别出绝大多数真实的马匹图像，图中能看出模型关注着马的脸、四肢和整体轮廓。frog类别紧随其后，F1-score为97.91%，其recall高达98.5%，说明模型极少遗漏青蛙样本，这得益于青蛙独特的绿色和两栖动物的形态特征。



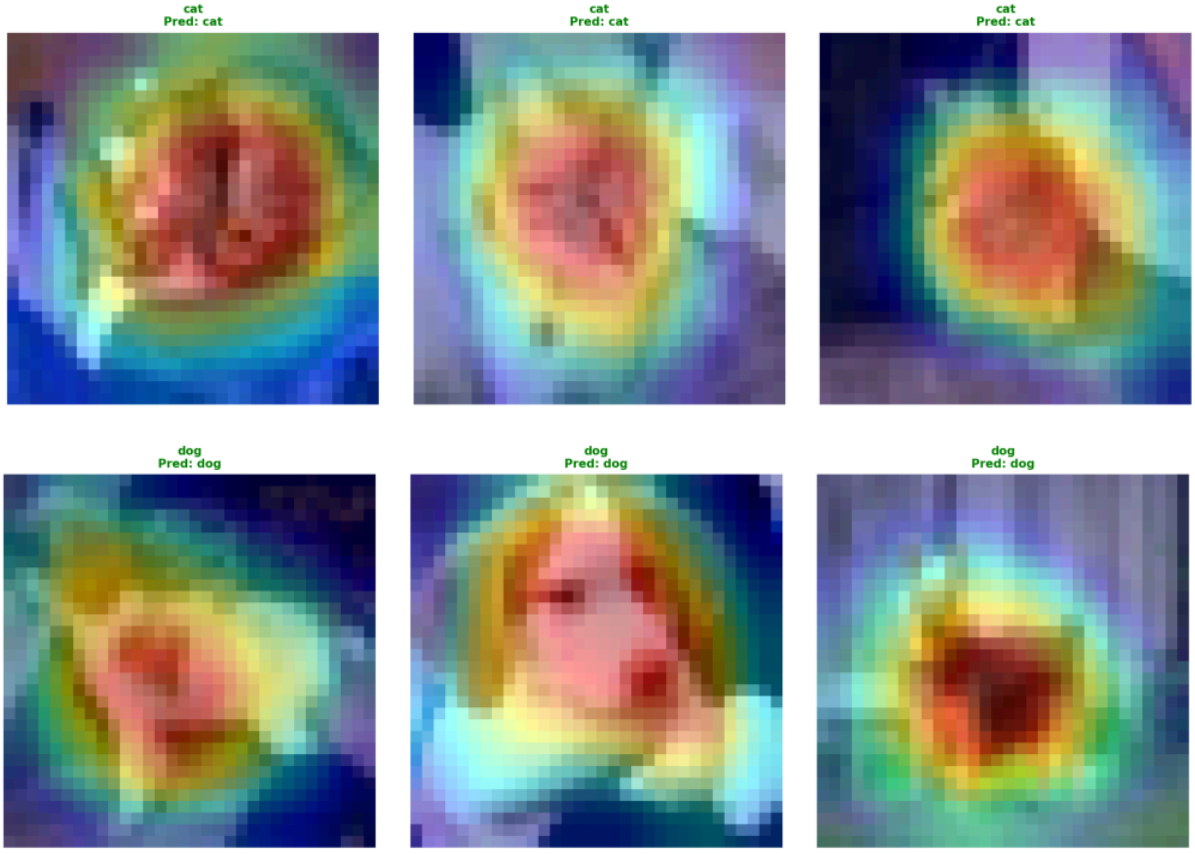


automobile和truck作为同属交通工具的类别，表现出相近的高水平性能。automobile的precision和recall均为97.9%，F1-score为97.9%，展现出完美的平衡。truck的precision为97.5%，recall为97.5%，F1-score同样为97.5%。这两个类别虽然视觉相似，但模型依然能够保持高精度识别，说明Wide ResNet成功学习到了区分轿车和卡车的细微差异。ship的F1-score为97.61%，得益于其98.1%的recall，模型对船舶图像的召回能力很强。

而纵观这三种交通工具，能发现模型都是关注交通工具的整体而非一些细节，这是因为他们的形状通常都具有独特的特征。



deer和airplane的F1-score分别为96.81%和96.94%，表现优异。deer的recall达到97.2%，表明模型很少将鹿误判为其他动物。airplane的precision为97.19%，当模型预测某图像为飞机时，这个判断的可信度很高。bird类别的F1-score为96.36%，虽然precision高达97.44%，但recall相对较低为95.3%，说明有一部分鸟类图像被模型遗漏或误判为其他类别，这与前文混淆矩阵分析中bird被误判为deer、frog和airplane的情况相吻合。

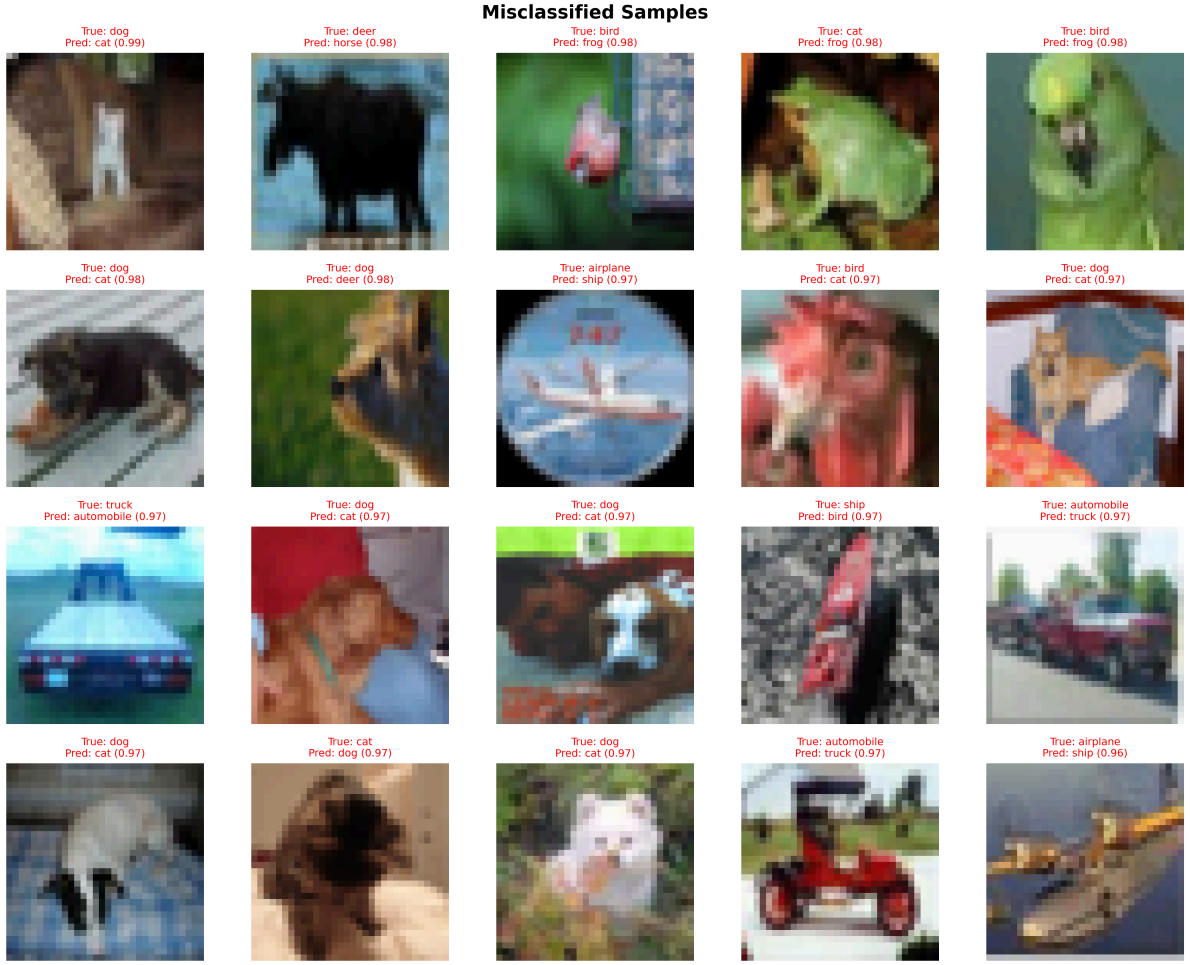


dog和cat是表现最薄弱的两个类别。dog的F1-score为93.05%，其recall为91.7%，低于precision的94.44%，这表明模型在识别狗时的遗漏率较高，有相当数量的dog样本被误判为cat或其他动物。cat的F1-score为91.95%，虽然其recall达到93.7%，但precision仅为90.27%，说明模型在预测cat时的误判率较高，会将一些其他类别的样本错误地识别为猫。cat的precision是所有类别中最低的，这与混淆矩阵中显示的许多dog、deer样本被误判为cat的现象一致。

整体而言，macro average 和 weighted average 的precision、recall和F1-score均达到96.47%左右，与整体准确率96.45%完全匹配，说明模型在各类别上的性能分布相对均衡，没有出现严重偏向某些类别而忽视其他类别的情况。这种均衡性是一个健康分类器的重要标志，表明模型在训练过程中充分学习了所有类别的特征，而不是依赖某些占主导地位的类别来提升整体准确率。

2.4 误分类样本分析

接下来可视化出误分类的样本集合：



最严重的混淆发生在dog和cat之间，这是一个双向但非对称的混淆模式。63个dog样本被误判为cat，占dog总数的6.3%，这是所有单向混淆中数量最多的一个。反向混淆的情况相对较轻，34个cat样本被误判为dog，占cat总数的3.4%。这种非对称性揭示了一个有趣的现象：当模型面对四足哺乳动物的模糊图像时，它更倾向于做出cat的预测。这可能是因为在训练过程中，cat类别的某些特征模式被网络赋予了更高的权重，或者cat类别的样本在视觉特征空间中占据了更大的区域。从视觉角度分析，当拍摄角度相似、毛色接近时，dog和cat在32×32的图像中确实难以区分。

airplane和ship之间的混淆排名第三，有16个airplane样本被误判为ship。在俯视角度拍摄的情况下，飞机的机翼和机身轮廓可能与船体的水平结构产生视觉上的相似性，尤其是在低分辨率下，两者都呈现出拉长的对称形状。此外，飞机图像的背景可能是天空或云层，在某些情况下与水面的颜色和纹理相近，进一步加剧了混淆。

truck和automobile之间存在相互混淆，16个truck被误判为automobile，15个automobile被误判为truck。作为同属地面交通工具的类别，这种混淆是可以预期的。轿车和卡车的主要区别在于车身高度、货箱结构以及整体比例，但在32×32的图像中，这些细节特征可能被严重压缩或丢失。当图像拍摄角度较为正面或背面时，两者的轮廓差异更加微妙，导致模型难以准确区分，同时这两个类别的相互混淆数量非常接近，说明模型在这对类别上没有明显的偏向性。

deer被误判为cat的情况有13例，占deer总数的1.3%。这一混淆可能源于某些deer图像中动物的姿态或背景环境与cat类似。例如，一只趴卧在地上的鹿可能与蜷缩的猫在整体轮廓上有相似之处，而鹿的棕色皮毛在某些光照条件下也可能与猫的毛色混淆。

bird类别的误判较为分散，其中12例被误判为deer，11例被误判为frog，8例被误判为airplane。bird与deer的混淆可能与自然环境背景有关，许多鸟类和鹿的图像都拍摄于树林或草地环境中，当图像分辨率降低时，背景的相似性可能掩盖了主体的差异。bird与frog的混淆则可能涉及颜色因素，某些鸟类的羽毛颜色偏绿色，在低分辨率下可能与青蛙的绿色皮肤产生混淆。bird与airplane的混淆似乎与形态有关，鸟类展翅飞翔时的轮廓可能在某些角度下与飞机的剪影相似。

dog被误判为deer的情况有10例，这可能与某些dog品种的体型和颜色接近鹿有关，尤其是大型犬种在侧面拍摄时可能呈现出与鹿相似的四肢和身体比例。

这些混淆模式共同指向一个核心问题：32×32像素的分辨率是CIFAR-10数据集的根本限制。在如此低的分辨率下，许多细节特征被抹去，类别之间的边界变得模糊。模型的误判往往不是随机的，而是基于视觉相似性做出的合理但错误的推断。**这也说明我们的模型在数据集上的训练和测试取得96.45%的准确率是一个很高的性能表现，这也得益于我们在第一章内对各类超参的调整，对多种数据增强手段的尝试，以及对多种模型架构、复杂度的测试。**

2.5 Mix-up

虽然本实验已经采用了Random Crop、Random Horizontal Flip和Cutout等数据增强技术，但仍有进一步深化的空间。本实验在初步尝试中启用MixUp，但发现在数据集上的效果并不好，反而会降低性能表现和拟合程度，因此在后续的优化中关闭了这一功能，但保留了代码以备后续实验。