
神经网络和深度学习 期中作业

钱皓然

18307110289

18307110289@fudan.edu.cn

朱笑一

18307130047

18307130047@fudan.edu.cn

梁敬聰

18307110286

18307110286@fudan.edu.cn

1 Introduction

在本次 Project 中，我们主要完成了两个任务。首先，我们使用 CNN 网络模型作为 baseline 在 CIFAR-100 上训练并测试，并且对比 Cutmix、Cutout 与 Mixup 三种方法的性能表现；其次，我们使用了两种目标检测模型：Faster R-CNN 和 YOLO V3 在 VOC 数据集上进行训练，并在测试集图像进行测试模型性能表现。我们的结果显示两种模型均能在测试图像上得到较好的检测分类效果。

在接下来的部分中，我们就将展示我们关于这个任务的成果，具体介绍顺序如下：在第二部分我们将会展示我们对于第一个问题，即使用 CNN 神经网络模型在 CIFAR-100 数据集进行图像分类的成果；在第三部分，我们会分别介绍 Faster R-CNN 和 YOLO V3 两种目标检测模型在 VOC 数据集上训练和测试的结果，最后在第四部分进行总结。¹²

2 The First Problem

在这个问题中，我们将实现一个适用于图像分类问题的 CNN 神经网络模型，并在 CIFAR-100 数据集 (Krizhevsky et al., 2009) 上训练与测试。本次实验实现的模型基于自适应 SAM (ASAM) (Kwon et al., 2021) 中用于实验的 ResNeXt 模型 (Xie et al., 2017)，各项模型与训练参数均尽量与 Kwon et al. (2021) 对齐。与此同时，在训练的过程中，我们会引入不同的图像数据增强技术，并比较它们改善模型分类性能的效果，同时也会与已有的结果比较。

2.1 CIFAR-100 数据集

本次实验采用 CIFAR-100 数据集 (Krizhevsky et al., 2009) 作为图像分类任务的训练与测试数据集。该数据集共有 50,000 张训练图像和 10,000 张测试图像，分属 20 个大类下的 100 个子类。每张图像都是 32×32 的 RGB 三通道图像，共 $3 \times 32 \times 32 = 3,072$ 个像素。为了充分体现 CNN 模型的分类性能，同时方便与既有结果比较，我们采用 100 分类标签作为待预测的标签，忽略 20 大类的标签。

¹ 实验代码请参考 <https://github.com/ljcleo/NeuNetOne>。

² 模型可以从 <https://drive.google.com/drive/folders/1JyAux4pSCsYahNlg6Ic6PmK6sXqRpG0c?usp=sharing> 下载。

根据 ASAM 的实验设定，所有模型都经过相同的训练轮次，因此原则上模型的训练过程并不需要验证集参与；但为了展示模型在训练过程中分类性能的变化，我们仍然会随机抽取训练集的 10% 作为验证集，随训练进程计算模型分类准确率，剩余的 90% 用于模型训练。此外，由于本实验引入的部分数据增强方式需要融合多个标签，因此所有的分类标签都以 one-hot 向量表示；同时我们仅在验证集和测试集上计算模型的分类准确率，而在训练集上只会计算损失函数。

2.1.1 数据增强

CIFAR-100 数据集的规模并不算大，因此利用复杂模型直接学习容易导致过拟合。数据增强可以有效缓解这一点，它能够在既有数据的基础上，通过一些图像或标签上的处理得到新的样本，从而实现数据集的扩充。

根据要求，本次实验比较了下列三种基于图像与标签的数据增强技术：

Mixup 该方法 (Zhang et al., 2017) 通过将两个图像-标签样本随机加权叠加得到新的样本，其中图像按照给定的加权比例在各个通道叠加，标签则是两个 one-hot 向量的加权和。在原始论文中，两份样本的加权比例是从 Beta(α) 分布中随机采样的，其中 α 是预先设置的参数。本次实验为了方便与其它方法比较，统一设置 $\alpha = 1$ ，即从 $[0, 1]$ 上的均匀分布上随机采样。

Cutout 该方法 (DeVries and Taylor, 2017) 通过随机将图像上的一块矩形区域全部置零得到，其中矩形区域是通过随机选取中心得到的正方形与原图像区域取交集得到的。根据原始论文的实验结果，本次实验设定正方形的大小为 8×8 (即抹去以随机选点为中心、横纵坐标 ± 4 的区域)，区域中心则在整张图像上均匀随机采样。

Cutmix 该方法 (Yun et al., 2019) 可以视为 Mixup 与 Cutout 的组合版本，是在 Cutout 的基础上，用第二份样本填充原本置零的区域。由于填充的部分包含有效信息，因此该方法不再固定区域面积，而是随机生成一个替换比（类似 Mixup 的加权比）并随机采样一个中心后，生成与原图像有相同纵横比的区域（对于 CIFAR-100 数据集而言也是正方形），再与原图像区域取交集。图像标签的加权比以实际得到的替换区域占原图像面积的比例为准。本次实验中，替换比也是从 $[0, 1]$ 上的均匀分布上采样得到的，区域中心则同样在整张图像上均匀随机采样。

图 1 和图 2 分别展示了在测试集和训练集的一个批次上应用上述数据增强技术得到的新样本，从左到右依次为原图像、Mixup、Cutout 和 Cutmix，可以看到不同的方法对图像与标签作出的修改。

另外，本次实验中所有图像都在应用上述数据增强方法之前进行了一些基础的预处理。在训练时，原图像会在填充至 40×40 后随机裁剪为 32×32 的子图像，并有一定概率在水平方向上翻转，最后在三个通道上分别准化。在测试时则分两种模式：

1. 单份：直接将原图像逐通道标准化后输入模型；
2. 十份：原图像在填充后裁剪中心和四个角落的 32×32 子图像，并水平翻转得到总共 10 张图像，标准化后输入模型，最终取 10 个输出向量的均值作为的预测结果。

我们同时测试了上述两种模式下模型的分类准确率。

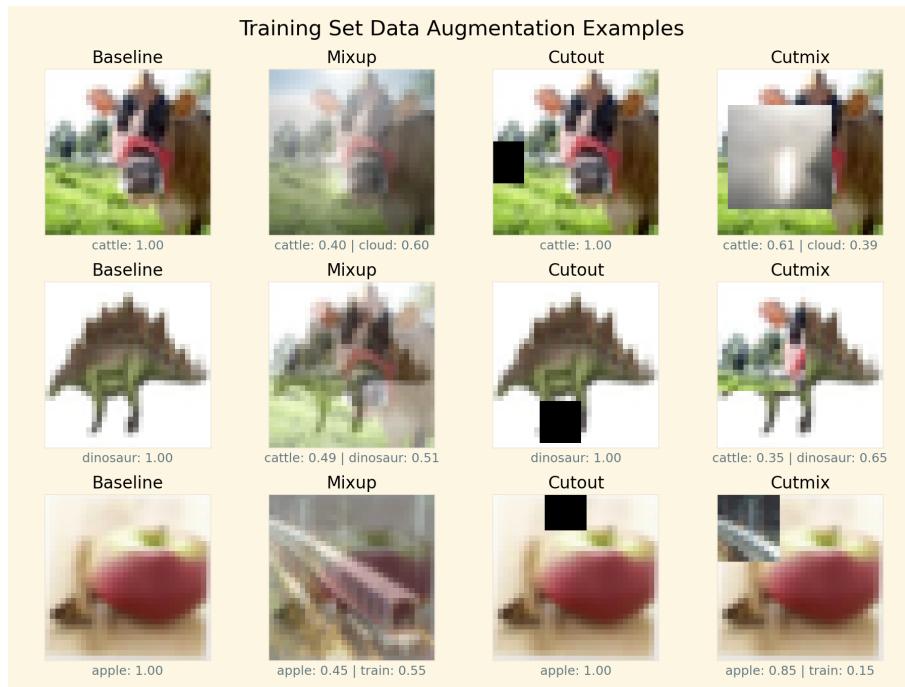


图 1: 训练集数据增强示例

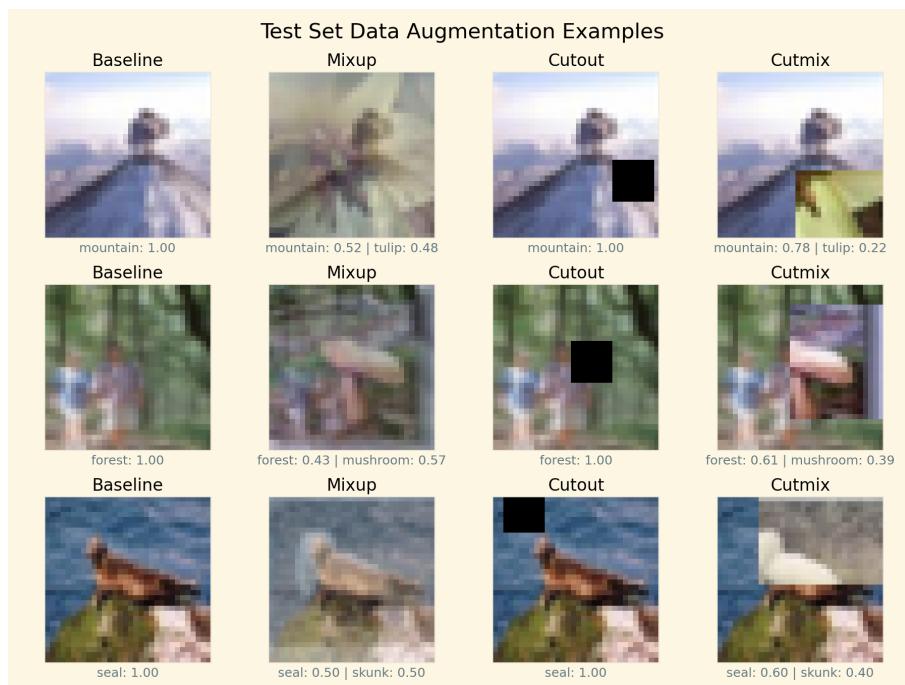


图 2: 测试集数据增强示例

2.2 网络模型

本次实验采用的 ResNeXt 模型 (Xie et al., 2017) 是 ResNet 模型 (He et al., 2016) 的一个改进版本，将原版 ResNet 中的每一个卷积层调整为分组卷积³，从而用等量甚至更少的参数实现更高维的特征提取，如图 3 和图 4 所示。除此之外，ResNeXt 与 ResNet 的部件组成基本一致，都由若干阶段组成，其中每个阶段包含数个重复的卷积层，且下一个阶段的特征数翻倍而图像边长减半，最后通过全局池化与全连接层获得最终的分类预测。

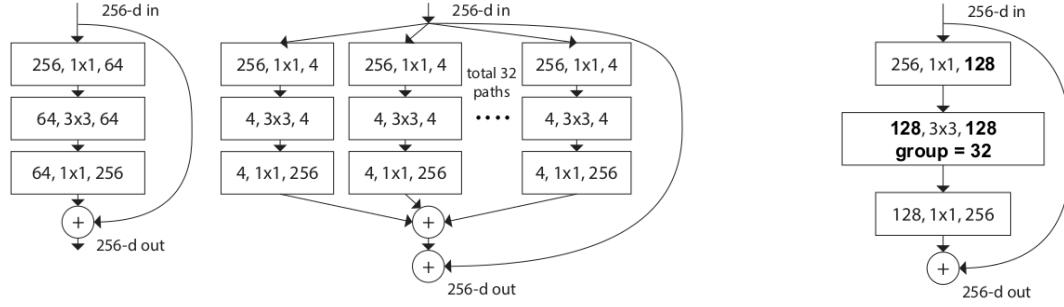


图 3: ResNet 与 ResNeXt 的卷积层

图 4: ResNeXt 卷积层的分组卷积形式

对于 CIFAR-100 而言，由于图像边长只有 32，因此网络仅包含三个阶段（接在一个 3×3 卷积层之后），其中每个阶段有 3 个分组卷积层，输出特征数分别为 256、512 和 1,024，最后接全局池化与一个全连接层，与原论文保持一致。整个网络共有 $1 + 3 \times 3 + 1 = 29$ 个参数层，因此我们跟随 Xie et al. (2017) 称该模型为 ResNeXt-29。

2.3 训练配置

本次实验特别尝试了 2021 年发表的 ASAM 优化器 (Kwon et al., 2021)。ASAM 的基础是 SAM 优化器 (Foret et al., 2020)，其在传统 SGD 的基础上考虑了邻域内函数的峰值，能够避免训练过程中陷入局部空洞，尽量选择平坦的区域前进。SAM 优化器（采用 2-范数，下同）的目标函数为

$$\min_w \max_{\|\epsilon\|_2 \leq \rho} L_S(w + \epsilon) + \frac{\lambda}{2} \|w\|_2^2$$

通过梯度近似得到的每步迭代过程则表示为

$$\begin{cases} \epsilon_t = \rho \frac{\nabla L_s(W_t)}{\|\nabla L_s(w_t)\|_2} \\ w_{t+1} = w_t - \alpha_t (\nabla L_S(w_t + \epsilon_t) + \lambda w_t) \end{cases}$$

其中 α_t 是当前的学习率。在此基础上，ASAM 为原目标函数引入了标准化算子 T_w^{-1} ，使整个优化过程满足尺度不变性。ASAM 的目标函数为

³这是 ResNeXt 模型卷积层三种等价实现方法中的一种，也是 He et al. (2016) 推荐采用的方案。

$$\min_w \max_{\|T_w^{-1}\epsilon\|_2 \leq \rho} L_S(w + \epsilon) + \frac{\lambda}{2} \|w\|_2^2$$

相应的迭代过程则修改为

$$\begin{cases} \epsilon_t = \rho \frac{T_{w_t}^2 \nabla L_s(W_t)}{\|T_{w_t} \nabla L_s(w_t)\|_2} \\ w_{t+1} = w_t - \alpha_t (\nabla L_S(w_t + \epsilon_t) + \lambda w_t) \end{cases}$$

具体到本实验中，我们跟随 Kwon et al. (2021) 的最优配置，取 $T_w = diag(|w_1|, \dots, |w_n|)$ 为逐元素标准化算子（另外增加了稳定因子 ηI_n ，实验中统一设定稳定系数 $\eta = 0.01$ ）并采用 2-范数。所有实验的初始学习率统一为 $\alpha = 0.1$ ，邻域大小 $\rho = 1$ ，权值衰减因子 $\lambda = 5 \times 10^{-4}$ ，并采用带重启的余弦衰减 (Loshchilov and Hutter, 2016) 调整学习率⁴。

其它方面，本次实验中模型损失定义为交叉熵损失，所有实验的批量大小均设为 128，并在训练集上重复训练 200 代。另外除了该标准配置外，我们还测试了用 SGD 替代 ASAM（初始学习率相同）和用阶梯衰减替代余弦衰减（在第 60、120 和 180 代衰减为 0.1 倍）的训练配置。

2.4 实验结果

所有模型的测试结果如表 1 所示。总体而言，各个模型的结果并未达到 Kwon et al. (2021) 的预期结果，这很有可能与学习率的调整方式不同，以及移除了一些其它的数据增强技术（为了比较三种图像-标签数据增强方法）有关；另外采用十份预测的结果在大部分情况下都优于单份预测。ASAM 优化器在实验中的优势不算明显，甚至在不采用图像-标签数据增强的情况下差于 SGD；不过 SGD 的方差更大，其结果具有一定的偶然性。

三种数据增强技术在不同的优化策略下效果不一：采用 ASAM 训练的模型在混合类方法 (Mixup 和 Cutmix) 下表现更好，而 Cutout 作用有限；然而替换为 SGD 时，数据增强（尤其是 Cutout 和 Cutmix 等剪切类方法）反而降低了模型的分类准确率。另一方面，采用阶梯衰减时混合类方法也削弱了模型性能，这又与采用余弦衰减的情形相反。总而言之，不同方法在不同的应用情景下存在明显的效果差异，其优劣不能一概而论。

模型	单份				十份			
	原始	Mixup	Cutout	Cutmix	原始	Mixup	Cutout	Cutmix
ResNeXt-29	69.78	71.55	69.76	71.42	70.19	72.61	70.44	72.27
ResNext-29 (SGD)	74.49	70.06	62.63	60.86	74.67	71.67	63.87	61.56
ResNeXt-29 (Step)	71.99	69.04	71.95	69.70	72.52	70.27	73.13	70.63

表 1: 模型测试集分类准确率 (%)

图 5 展示了 ResNeXt-29 模型采用四种图像数据增强方法时的训练曲线。可以看到，混合类方法的训练损失明显更高，这是因为二者生成的新样本不再是单一标签，因此更难预测；然而在验证集上，这

⁴由于 Kwon et al. (2021) 没有提供相关参数，因此我们采用 Loshchilov and Hutter (2016) 中的推荐值 $T_0 = 10$ 、 $T_{mult} = 2$

两种方法虽然会使模型在早期损失下降更慢一些，但经过充分训练后，无论是验证损失还是验证准确率都更胜一筹。相比之下，Cutout 方法从一开始就有更好的验证集损失与分类准确率，但优势不明显，最后被混合类方法超越。

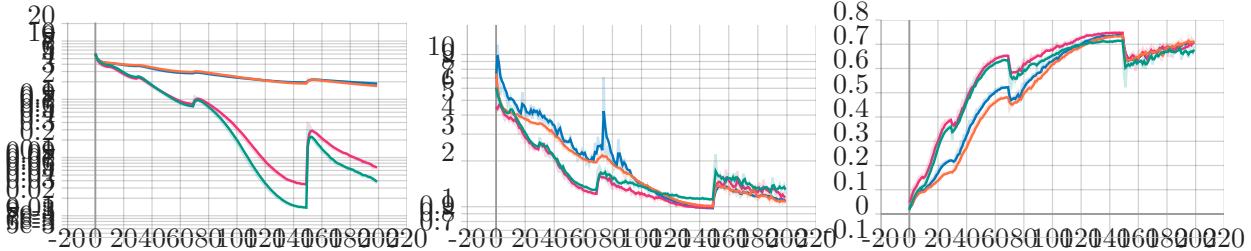


图 5: 训练损失、测试损失与测试准确率。绿色: 原始版本; 蓝色: Mixup; 粉色: Cutout; 橙色: Cutmix

3 The Second Problem

在这个问题中，我们将分别利用目标检测模型 Faster R-CNN 和 YOLO V3 在 VOC 数据集上训练。为了验证模型的性能，我们会将模型分别应用在在 VOC 测试数据集及 VOC 数据集以外但是包含有 VOC 中类别物体的图像上，考察模型在目标检测和类别预测上的效果。在我们下面的介绍中，我们会首先对 VOC 数据集进行简要的说明，然后我们分别介绍我们是如何用 Faster R-CNN 和 YOLO V3 对处理后的数据进行训练以及测试。

3.1 VOC 数据集

PASCAL VOC 挑战赛 (The PASCAL Visual Object Classes) 是一个世界级的计算机视觉挑战赛，关注于分类，定位，检测，分割，动作识别等多个任务。VOC 数据集即为该挑战赛提供的数据集，包含一共 20 个类别，具体类别名称详见图 6。

目前目标检测普遍使用的是 VOC2007 和 VOC2012 数据集。由于 VOC2012 的测试集未被公开，所以在训练 YOLO v3 模型时，我们这里使用 VOC2007 和 VOC2012 的训练集作为训练集，使用 VOC2007 和 VOC2012 的验证集作为验证集，使用 VOC2007 的测试集作为测试集。

值得注意的是，VOC 数据集中的位置标注格式与 YOLO v3 中的不同。VOC 数据集中的位置标注以左上角为原点，向右向下分别为宽度和高度的正方向，直接标注检测框在宽度和高度上的最大值和最小值；YOLO v3 的位置标注则是标注检测框的中心带你的位置，以及检测框的宽度和高度。因此在进行训练和测试之前，我们需要首先对 VOC 数据集中的数据格式进行简单的转化，具体代码可见 convert.py 文件。

3.2 Faster R-CNN

在这一部分，我们主要介绍我们如何使用 Faster R-CNN 对 VOC 数据集进行训练以及测试。注意，这里我们使用的 Faster R-CNN 代码为 potterhsu 提供的版本。

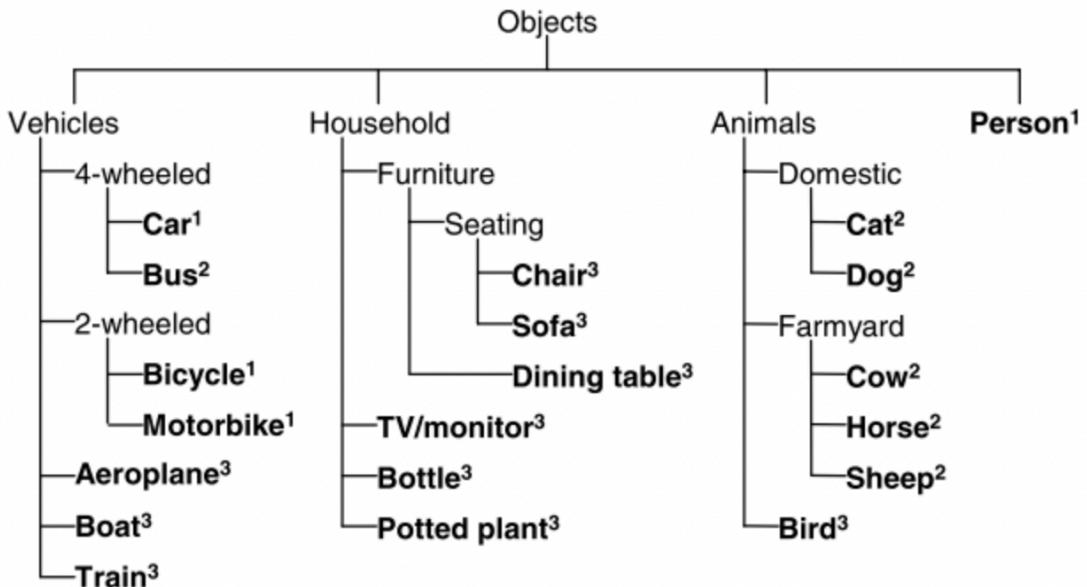


图 6: Classes of VOC dataset.

3.2.1 Introduction of the model

在使用 Faster R-CNN 对 VOC 数据集进行训练和测试之前，我们首先对 Faster R-CNN 进行大致的介绍。Ross B. Girshick 在 2015 年发表的论文《Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks》 Ren et al. (2015) 中提出了 Faster R-CNN 模型。该模型解决了 SPPnet 和 Fast R-CNN 模型中建议区域计算的瓶颈，从而使该模型计算更加迅速。

Faster R-CNN 算法的关键部分有三个：

1. 共享基础卷积层。该层用基础的卷积层 +RELU 激活函数 + 池化层提取图片的特征，这些特征被共享用于后续的 Region Proposal Networks 层。
2. Region Proposal Networks 层。该层用于生成 region proposals，并判断 anchors 为正例或负例，然后用 bounding box regression 对 anchors 进行修正，从而得到精确地 anchors。
3. Roi 池化层。利用第一步得到的特征和第二步得到的 Proposals，综合后提取出针对特定 proposals 的特征。

最终利用第三步得到的新特征，进行分类，并且再次微调 anchors。

为了实现这个网络，Faster R-CNN 使用了如图 7 所示的网络结构。该图的上半部分则为刚刚提到的共享基础卷积层，左下角的网络则为 Region Proposal Networks 层，右下角的网络则为 Roi 池化层与最终的分类器。

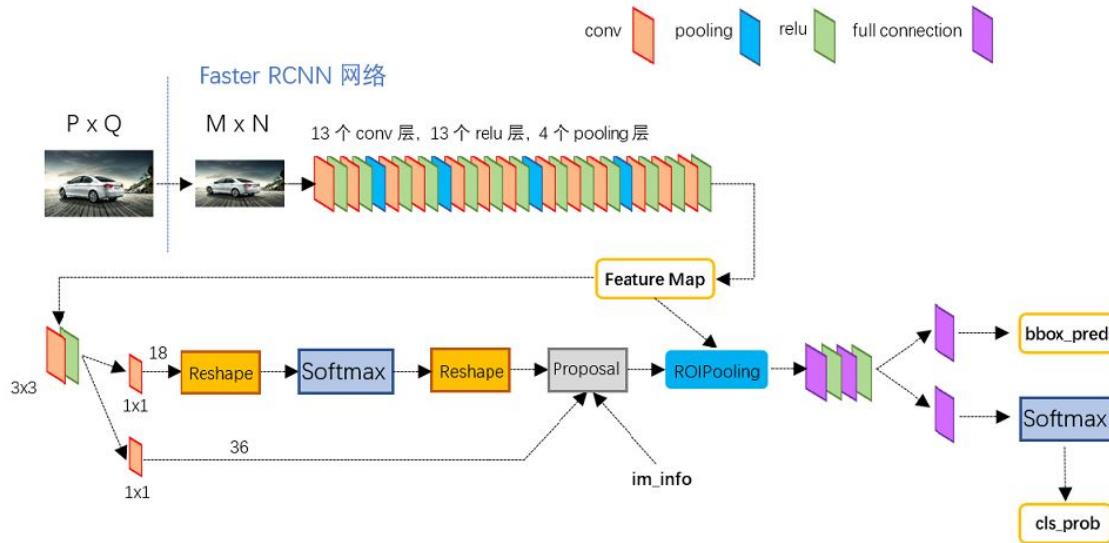


图 7: Structure of Faster R-CNN.

3.2.2 Training of the model

我们以 VOC2007 作为数据集进行训练与测试。由于模型参数量较大，加上硬件设备算力限制，我们这是 Batch Size 为 1。

初始学习率为 0.001，然后我们使用指数下降的方式减少学习率，即在第 n 个 epoch 中，学习率为 $\alpha_n = 0.001 \times 0.995^n$ ，优化器为随机梯度下降 (SGD)，总共训练 36 个 epoch。损失函数由四部分组成，详情请见下图。

图 8 从左至右分别为 Total Loss、Anchor Object Loss、Anchor Transformation Loss、Proposal Class Loss 和 Proposal Transformation Loss。可以看到大概在第 15 个 epoch 的时候测试集的各种 Loss 都几乎是最小的，继续训练下去反而会使得模型产生过拟合现象。

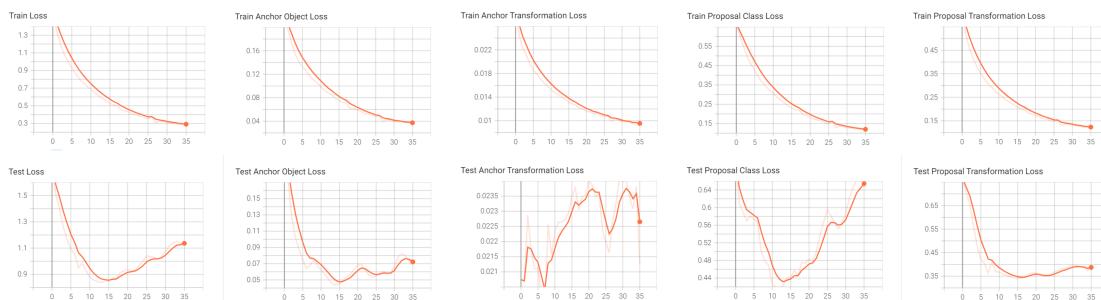


图 8: Faster R-CNN Train and Test Loss

最后，我们以 mean AP 对模型进行评价，测试集的 mean AP 如下图。

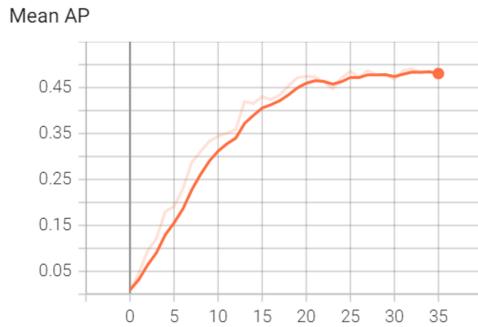


图 9: Faster R-CNN Test mean AP

3.2.3 Testing of the model

在我们完成了对于模型的训练之后，我们在 VOC2007 的测试集上进行测试以考察模型的性能。

在这里，我们展示四张测试集上的图的效果。



图 10: Detection on VOC: prediction image

在 VOC 测试集中的图像表现如图 10 所示，大部分实体都能准确的被找到，如右上角的图能完美的找到两个实体，但也有遗漏实体，如左上角的图遗漏了右侧的一小部分车实体，左下角的图遗漏了中间的植物，以及右下角的图遗漏了中间的狗，主要原因可能是实体过小而难以被识别出来。还有一些极少数的错误判别，如左上角的误识别了画框为显示器，可能是因为画框与显示器比较相似，所以误判别了，因此训练出来的 Faster R-CNN 在一些相似物体的判别上还有待一定的提升。

我们继续在 VOC 数据集意外的图像进行检验，结果如图 11 所示，在这三张图上 Faster R-CNN 表现都非常优秀，左上角的图中几乎找到了所有的人，自行车，但是遗漏了被遮挡面积非常大的骑车，而剩下的两张图都能完美的找到所有实体，特别是右上角的图，完美的找到了一只猫和三只狗，左下角的图

则能很好的识别出火车和五个人，该图的火车容易被识别成公共汽车，因此 Faster R-CNN 在这张图中表现非常出色。



图 11: Detection on custom data

3.3 YOLO V3

在这一部分，我们主要介绍我们如何使用 YOLO V3 对 VOC 数据集进行训练以及测试。注意，这里我们使用的 YOLO v3 代码为 ultralytics 提供的版本。

3.3.1 Introduction of the model

在使用 YOLO V3 对 VOC 数据集进行训练和测试之前，我们首先对于 YOLO V3 进行大致的介绍。YOLO (You Only Look Once, YOLO) 算法最初由 Joseph Redmon 等人在 2015 年提出 Redmon and Farhadi (2018)，经过两轮的迭代，在 2018 年发展出 YOLO V3 版本。相较于 R-CNN 系列算法两阶段的目标检测方式，即先产生候选区域，再做分类和位置坐标的方法，YOLO V3 作为一个单阶段预测算法，仅使用单个网络结构，在产生候选区域的同时即可预测出物体类别和位置。

YOLO V3 算法的基本思想可以分成两部分：

- 按一定规则在图片上产生一系列的候选区域，然后根据这些候选区域与图片上物体真实框之间的位置关系对候选区域进行标注。跟真实框足够接近的那些候选区域会被标注为正样本，同时将真实框的位置作为正样本的位置目标。偏离真实框较大的那些候选区域则会被标注为负样本，负样本不需要预测位置或者类别
- 使用卷积神经网络提取图片特征并对候选区域的位置和类别进行预测。这样每个预测框就可以看成是一个样本，根据真实框相对它的位置和类别进行了标注而获得标签值，通过网络模型预测其位置和类别，将网络预测值和标签值进行比较，就可以建立起损失函数。

为了实现这一思想，YOLO V3 使用了如图 12 所示的网络结构。可以看到，YOLO V3 使用了 darknet-53 的前面的 52 层作为支柱，并且大量使用残差的跳层连接。为了降低池化带来的梯度负面效果，作者

通过调整卷积层的步长来替代池化层实现降采样的效果。除此之外，为了加强算法对小目标检测的精确度，YOLO V3 借鉴了 FPN(feature pyramid networks) 的思想，采用多尺度来对不同大小的目标进行检测，越精细的网格就可以检测出越精细的物体。

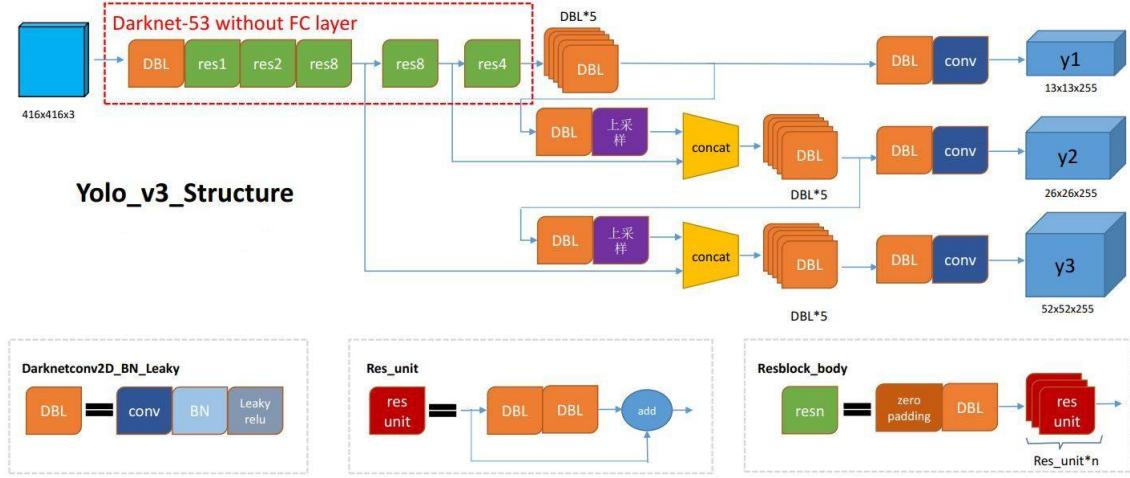


图 12: Structure of Yolo v3.

为了最后得到较好的检测和分类效果，YOLO V3 设计了如下三个损失函数：

1. 坐标误差：

$$\begin{aligned} Loss_{\text{box}} &= \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i^j)^2 + (y_i - \hat{y}_i^j)^2 \right] \\ &\quad + \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} \left[\left(\sqrt{w_i^j} - \sqrt{\hat{w}_i^j} \right)^2 + \left(\sqrt{h_i^j} - \sqrt{\hat{h}_i^j} \right)^2 \right] \end{aligned}$$

表示当第 i 个网格的第 j 个 anchor box 所产生的 bounding box 和真实目标的 box 去比较，计算得到中心坐标误差和宽高的误差。

2. 置信度误差

$$\begin{aligned} Loss_{\text{obj}} &= - \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} \left[\hat{C}_i^j \log(C_i^j) + (1 - \hat{C}_i^j) \log(1 - C_i^j) \right] \\ &\quad - \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{noobj}} \left[\hat{C}_i^j \log(C_i^j) + (1 - \hat{C}_i^j) \log(1 - C_i^j) \right] \end{aligned}$$

其中 C_i^j 表示第 i 个网格的第 j 个 anchor box 所产生的 bounding box 是否负责预测某个对象，我们使用交叉熵来计算其与真实值的误差。

3. 分类误差

$$Loss_{\text{cls}} = - \sum_{i=0}^{S^2} I_{ij}^{\text{obj}} \sum_{c \in \text{classes}} \left[\hat{P}_i^j \log(P_i^j) + (1 - \hat{P}_i^j) \log(1 - P_i^j) \right]$$

计算我们的分类预测值与真实值的误差。

我们的最终损失即把这三个部分的误差累加在一起，用以训练我们的检测模型。同时，为了评估我们最终模型的好坏，我们也使用目标检测中常用的指标：mAP@0.5、mAP@0.5:0.95 和 precision 等来体现我们模型的性能。

3.3.2 Training of the model

我们以 VOC2007 和 VOC2012 的测试集作为测试集进行训练。为了得到更好的检测效果，我们使用了预训练的权重继续进行训练，部分参数设置如表 2 所示。为了选择合适的 epoch 防止训练过长导致的过拟合，我们利用了早停法，设置耐心值为 100 进行训练，可以看到模型在第 265 轮时早停了，考察训练和验证损失可以看到如图 13 所示。我们可以看到，在 200 步开始，我们有尽管训练集上的损失还在下降，在验证集上损失逐渐增长，出现过拟合的现象，因此我们最后设定 epoch 为 200 并进行训练得到模型以用于验证测试模型的性能，此时我们有模型的准确性如图 13，可以看到随着训练步数的增加，我们的准确性不断上升。

Batch size	16	momentum	0.937
initial learning rate	0.01	epoch	200
OneCycleLR learning rate	0.1	optimizer	Adam

表 2: Part of Parameter Setting for YOLO v3

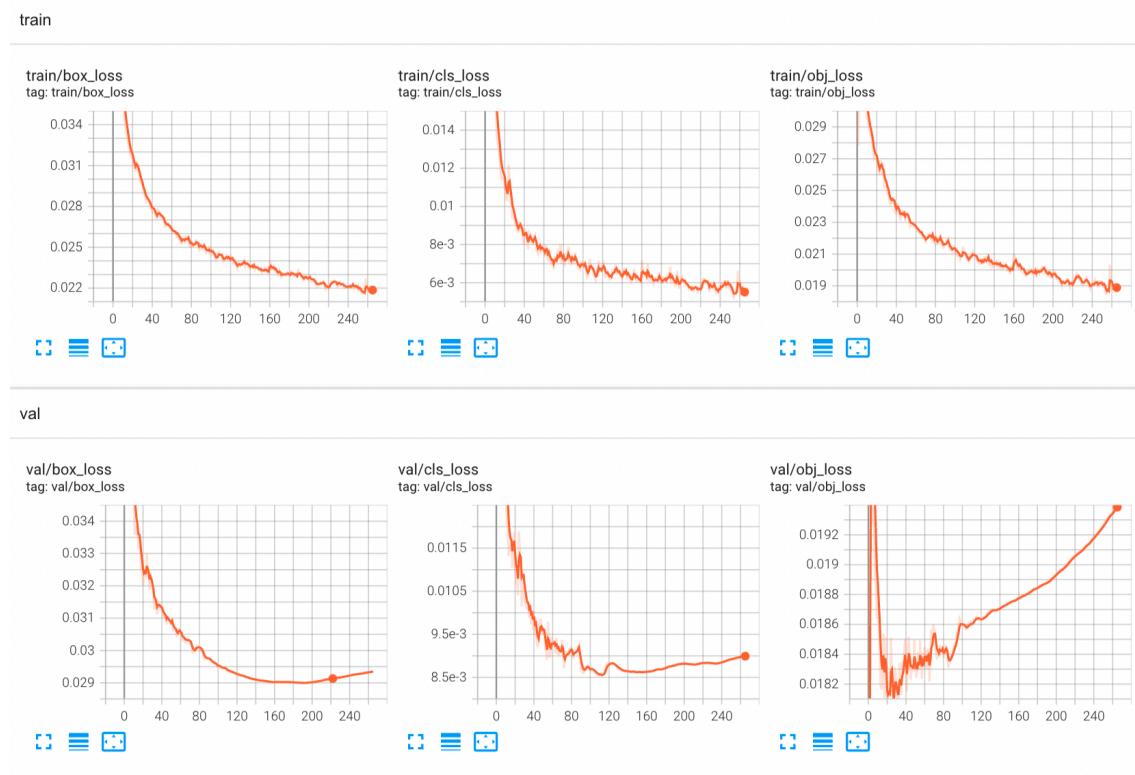


图 13: Training & Validating loss using Early Stopping.

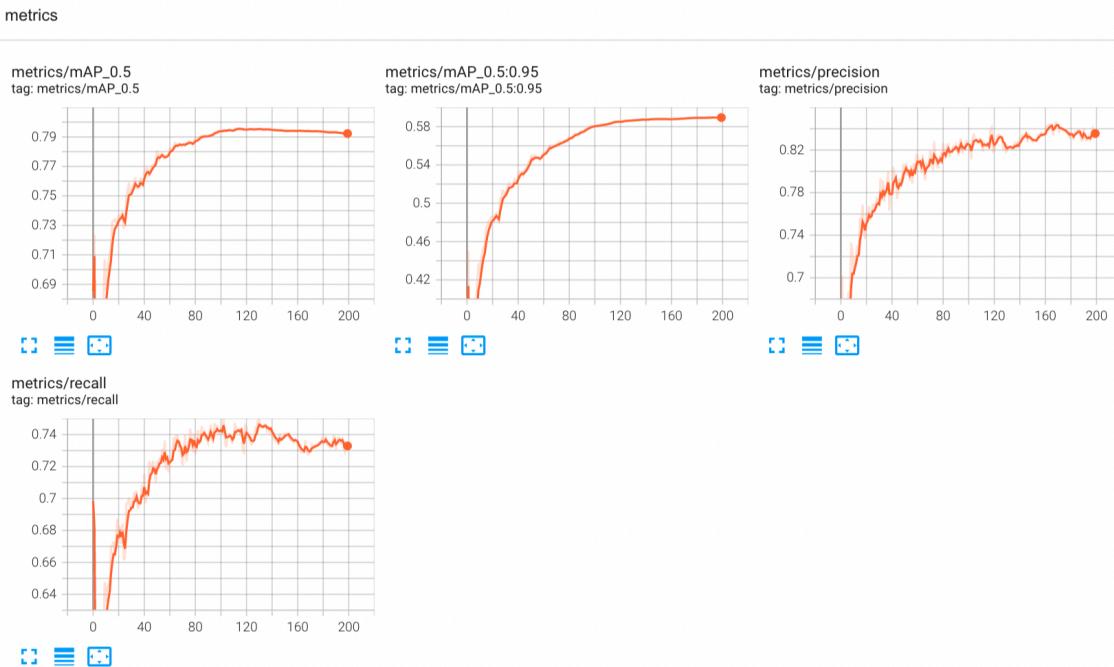


图 14: Metrics under 200 epochs.

3.3.3 Testing of the model

在我们完成了对于模型的训练之后，我们在 VOC2007 的测试集上进行测试以考察模型的性能。

首先在模型的分类上，我们得到了如图 15 所示的混淆矩阵。可以看到，大部分的值都聚集在对角线上，对角线以外的部分值较小，说明我们对大部分的物体都成功进行了正确的分类。

另一方面，为了考察我们目标检测的能力，我们分别在 VOC 测试集中选了 4 张图片，在 VOC 数据集选了三张图片进行验证。在 VOC 测试集中的图像表现如图 16 和图 17 所示。可以看到尽管大部分的图像中的物体都可以正确识别，如图右上完美的把但我们还是看到不少漏检查的物体，如图左上的汽车和屏幕中的人，左下的植物和右下的狗。同时还有极少的错误分类，如左上的图中识别出了遥控器。仔细考察我们的错误的例子，右下的图可能是因为狗只露出了一个头儿没有露出身体；左上的图也是类似的，不仅人只有部分身体，还因为光线原因出现过曝的现象导致脸几乎和背景融为一体，而车也因为拍摄时的快速移动而产生了一定的模糊。如何对这种物体的局部依然进行准确的识别是我们提高准确率的关键。

我们再继续在 VOC 数据集以外的图像进行检验，结果如图 18 所示。我们还是可以看到与之前在 VOC 测试数据类似的结论，如右上图片中猫后面的狗和左上图中被人挡住的车子，对于这种只显露局部区域的物体我们容易产生漏检的情况。另一方面，由于在左下场景中火车只给了局部的图像，使得其跟公交车的侧身类似，导致模型的错误分类。因此我们可以看到，对于只给出局部信息的物体的识别是我们进一步提升的关键问题。

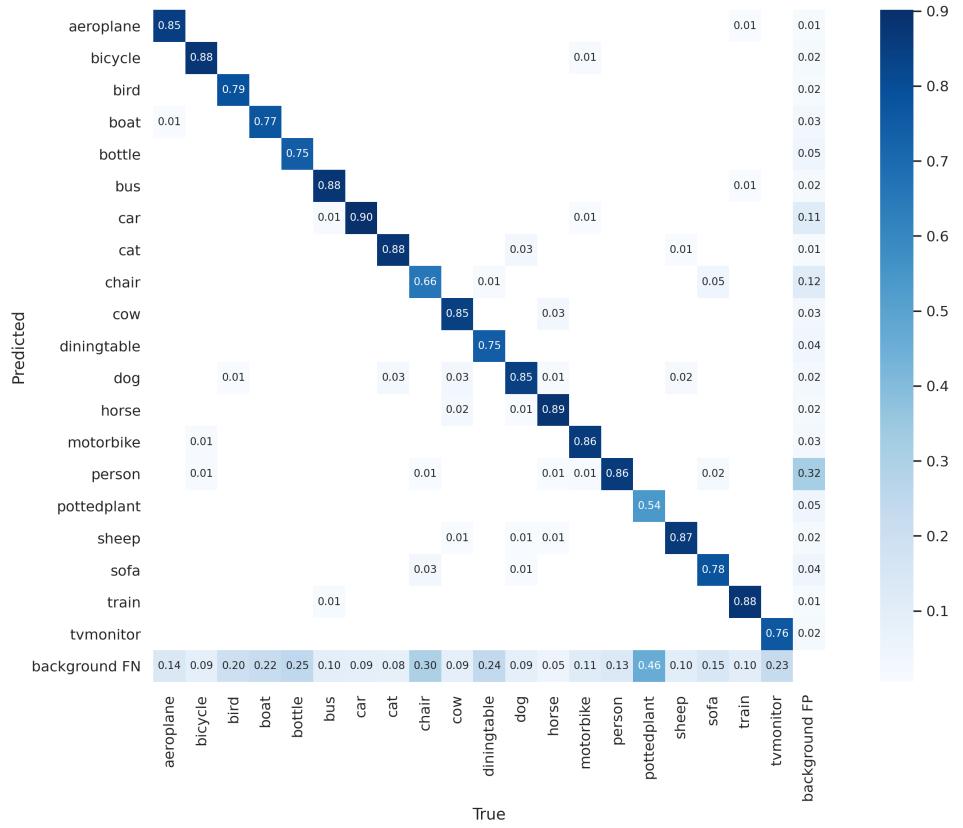


图 15: Confusion matrix on the test dataset.



图 16: Detection on VOC: label image

图 17: Detection on VOC: prediction image



图 18: Detection on custom data.

4 Conclusion

在第一个任务中，我们使用 CNN 神经网络模型在 CIFAR-100 数据集进行图像分类，并对比使用 cutmix、cutout、mixup 三种图像增强算法下模型性能的差别，在得到不错的分类准确性的同时，我们发现三种数据增强技术在不同的优化策略下效果不一，需要结合应用场景进行进一步分析；在第二个任务中，我们使用了两种目标检测技术：Faster R-CNN 和 YOLO V3 在 VOC 数据集上进行训练和测试，得到了不错的检测效果，对 VOC 训练集意外以外的图像也能得到不错的结果。我们进行了详尽的可视化来展示我们的模型的训练进展和最终的检测效果。

参考文献

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, pages 5905–5914. PMLR, 2021.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.