成　绩	
评阅人	

复 旦 大 学

研 究 生 课 程 论 文

论文题目：　Project4-基于特征融合的 Animals 图像分类

修读课程：　智能信息处理的统计方法（COMP620053）

选课学期：　2024-2025 学年第一学期

选课学生：　宋安洋（24210240291）

完成日期：　2024.1.6

项目名称	Project4-基于特征融合的 Animals 图像分类		
完成时间	2024/1/6	指导教师	路红
小组成员 （姓名学号）	专业学院		成员分工 （任务/在报告中的作用）
宋安洋 24210240291	计算机科学技术学院		尝试对深度学习方法改进并尝试特征融合。首先基于 PJ3 优化了部分超参并进行了对比实验。复现了 RandAugment 的自动数据增强方法尝试提升表现并进行了对比实验。复现了 Point Fusion、GLIP、Decision Fusion、MVC 的融合思想尝试提升表现并进行了对比实验。完成了实验对应的报告内容以及引言、混淆矩阵分析、总结、参考文献、附录。
李天佑 24212010019	软件学院		尝试对机器学习方法改进并尝试特征融合（基于 PJ2 首先使用颜色直方图提取全局特征和 sift 提取局部特征，之后在刘喆和邓晨欣帮助下进行特征融合）。完成了实验对应的报告内容。
刘喆 24210240239	计算机科学技术学院		尝试完成了一个简易的图像分类综述（3.1、3.2）。机器学习方法中，帮助李天佑进行特征融合；深度学习方法中，复现了 WFCG 的融合思想尝试提升表现并进行了对比实验。尝试对基分类器预测结果用 hard voting 提升表现并进行了对比实验。完成了实验对应的报告内容。
邓晨欣 24210240144	计算机科学技术学院		尝试完成了一个简易的图像分类综述（3.3、3.4）。机器学习方法中，帮助李天佑进行特征融合；深度学习方法中，尝试对 fusion 分类器和部分基分类器预测结果用 hard voting 提升表现并进行了对比实验。完成了实验对应的报告内容。

目　　录

1 引言

在 PJ2、PJ3 中我们分别使用机器学习方法和深度学习方法简单完成了 Animals-10 图像分类任务。在这个任务中，一共有 6 个类别，分别为：蝴蝶、猫、鸡、牛、狗、羊。训练集一共有 6000 张图像，每个类别分别有 1000 张图像。测试集一共有 1800 张图像，每个类别分别有 300 张图像。令人高兴的是，数据集的分布是高度平衡的。对于机器学习方法，在 PJ2 的基础上，我们首先进行早期融合和后期融合，早期融合：在特征提取阶段，将全局特征（颜色直方图）和局部特征（如 SIFT + BoW）合并为一个联合特征向量。后期融合：训练单独的分类器分别处理全局和局部特征，最后通过概率加权的方式融合分类结果。对于深度学习方法，在我们首先尝试了对超参数进行合理的改进，我们使用了线性衰减的 lr scheduler，并在前 5%的 steps 中 warmup，训练 3 个 epoch 来实现恰好基本完全收敛与最佳的验证效果，PJ4 中，vit_b_16 表现最佳，首先在这个过程中，准确率从 **97.28%** 提升到了 **98.28%**。然后我们尝试复现了 **RandAugment** 的数据增强方法，vit_b_16 的准确率从 98.28% 提升到了 **98.61%**。接着我们尝试复现了 5 种深度学习特征融合的方法，当需要两个 backbone 时，采用表现最好的 vit_b_16 和 densenet201，最终的实验结果是采用 **GLIP** 的交叉注意力+残差思想表现最佳，相比于 vit_b_16 的 98.61%，准确率再一次提升，提升到了 **99.06%**。对于每次尝试，我们都有详尽的完整的实验对比，也通过混淆矩阵对结果进行了一定的分析。最后我们还尝试了进一步用 hard voting 的方式对各分类器预测结果进行集成，但是没有超过 99.06%，我们指出，想要进一步提高表现，或许只能尝试进一步扩大数据增强规模，或是只能选用更加强大的 backbone。

2 PJ2与PJ3实验结果回顾

PJ2 与 PJ3 的实验结果如表 2.1 所示。

模型	ACC(%)	F1(%)
决策树	44.72	40.40

SVM	45.50	44.47
随机森林	45.22	44.28
XGBoost	49.28	48.87
LightGBM	50.94	50.45
lenet5	51.56	51.30
resnet152	97.22	97.23
densenet201	**97.44**	**97.44**
alexnet	89.89	89.93
googlenet	90.28	90.28
vgg19_bn	97.28	97.28
vit_b_16	97.28	97.29

表 2.1 PJ2 与 PJ3 实验结果回顾

PJ2 与 PJ3 的实验结果如表 2.1 所示，这是已修正的结果。需要强调的是，负责 PJ2 的同学在 PJ2 PPT 中的给出的结果不正确，已在 PJ3 报告时修正。PJ3 报告由于时间紧张 googlenet 那一行 90.28 误写成了 97.28,其余没问题。在 PJ4 中，我们对关键结果已反复核对，并且在代码压缩包中，我们留存了包括 tfevents 在内的各种 logs，预测结果也留存为了 npy 文件，对于最佳模型参数的 checkpoints，如有需求可以联系我们获取。

3 图像分类综述

3.1 图像分类研究背景

图像分类一直是图像领域的重要研究方向,主要应用于环境监测和社会经济领域。影像分类过程复杂，受多种因素影响，例如地形复杂性、数据质量和分类算法的选择。

尽管已经发展出许多高级分类方法，但由于研究区域景观的复杂性以及不同数据源的差异，将图像数据分类为主题图仍然面临挑战。本论文通过总结现有的高级分类方法和提升分类精度的技术，探讨了图像分类研究的主要进展和前景。

3.2 分类方法与技术

3.2.1 传统分类方法

传统分类方法包括 ISODATA、K 均值和最大似然分类器（MLC）等，这些方法已在诸多教科书中被详细讨论。这些方法通常基于像元的光谱特征进行分类，虽然在某些情况下能够提供有效的分类结果，但它们在处理复杂的图像数据时面临一些固有的局限性。

首先，ISODATA 和 K 均值等无监督分类方法通过迭代过程将像元划分到预定义的类中，依赖于初始类中心的选取，并且容易受到数据噪声的影响。其次，最大似然分类器（MLC）作为一种有监督方法，假设每个类别的光谱分布符合正态分布，并通过估计每个类别的统计参数来计算像元属于某一类别的概率。然而，这一方法在实际应用中面临的挑战包括类别的光谱重叠、训练样本不足以及在复杂地形或多变环境下的分类不准确性。

3.2.2 非参数分类方法

近年来，非参数分类方法如人工神经网络、决策树分类器和支持向量机的应用越来越广泛。这些方法无需假设数据的正态分布，因此更适合处理复杂景观和多源数据 。例如，支持向量机（SVM）、随机森林（RF）、神经网络（如深度学习）等方法被提出并应用于图像的分类任务。这些方法不依赖于传统的统计假设，能够更好地应对复杂的非线性关系和数据的高维特性。其中，支持向量机（SVM）通过寻找最优超平面进行数据分类，能够有效地处理小样本、高维度数据，且对异常值具有较强的鲁棒性；而随机森林通过集成多棵决策树的预测结果，可以减少过拟合，提高分类精度。

此外，神经网络尤其是深度学习方法（如卷积神经网络 CNN）在处理复杂图像数据时，显示出显著的优势。这些方法能够自动从数据中提取特征，减少人工特征工程的依赖，尤其在大规模数据和高分辨率影像的分类中表现出色。

然而，尽管新兴的机器学习方法在分类精度和应用范围上取得了显著进展，但它们的计算资源消耗较大，且对训练数据的质量和数量有较高要求。此外，许多先进算法的可解释性较差，这使得它们在一些应用中难以得到广泛接受。因此，如何在精度和计算效率之间找到平衡，并结合传统分类方法与先进机器学习技术，成为当前图像图像分类领域的一个研究热点。

3.2.3 结合分类方法

结合多种分类器的方法被证明可以显著提高图像图像分类的精度。这类方法通常利用不同分类器各自的优势，减少单一分类器可能存在的局限性，从而取得更好的分类结果。例如，将最大似然分类器（MLC）与决策树分类器（如 CART 或 ID3）结合使用，可以通过两者互补的特性提升分类精度。

最大似然分类器是一种基于统计学原理的有监督分类方法，它假设每个类别的像元值遵循特定的概率分布，并根据贝叶斯规则计算像元属于某个类别的后验概率。然而，MLC 的表现会受到训练样本的数量、类别之间光谱重叠及数据质量等因素的影响，尤其是在复杂的环境中，可能无法有效区分那些光谱上相似但在空间或时间特征上有所不同的类别。

另一方面，决策树分类器则通过递归地划分数据空间来进行分类，其优点在于易于理解和解释，可以处理非线性关系，并能够处理离散和连续特征。在实际应用中，决策树可以帮助克服 MLC 在高维空间中遇到的一些困难，特别是对于一些具有明显层次结构的地物类别，决策树能够更好地捕捉其中的规律和特征。

将最大似然分类器与决策树分类器结合，通常可以采取以下几种方式：一是通过“先分类后融合”的方法，将两种分类结果进行融合，利用决策树分类器对最大似然分类器的输出结果进行修正或重分类；二是通过“并行融合”的方式，在同一数据集上分别使用最大似然分类器和决策树分类器，再通过投票机制或加权平均的方式将两者的分类结果合并，以此提高整体分类精度。例如，可以为每个分类器分配不同的权重，根据其在特定数据集上的表现进行加权融合，从而获得更为精确的分类结果。

这种基于多个分类器的集成方法已在多个图像应用中获得了显著的成效。研究表明，将决策树与最大似然分类器结合，能够充分利用决策树在非线性分类中

的优势和最大似然分类器在处理光谱数据时的精确度，从而实现更为准确的地物分类。例如，有研究表明，基于集成学习的分类方法（如随机森林）与最大似然分类器结合，能够显著提高森林覆盖度分类的精度。类似地，有研究也提出，结合不同算法的分类结果，可以有效减少分类误差，特别是在复杂的地形或异质性较强的区域。

除了传统的分类器组合外，近年来，机器学习方法，如支持向量机（SVM）与决策树结合，或者神经网络与其他统计方法结合，也被广泛研究并应用于图像图像的多类分类任务。这些集成方法通过自动学习不同数据特征，能够自适应地调整各分类器的权重和参数，从而进一步提高分类精度，尤其是在大数据和高分辨率图像的处理过程中表现尤为突出。

3.3 提升分类性能的技术

3.3.1 利用多源数据

集成多种传感器数据（如光学数据与雷达数据）能够有效提高分类精度。这种方法通过融合多源数据的优势，增强分类的鲁棒性和精度 。

3.3.2 使用多时相数据

利用多时相数据能够捕捉植被和作物的不同物候特征，从而改进分类结果。例如，通过整合春季和秋季的图像数据，可以显著提升湿地分类的准确性。通过融合不同源的数据，可以充分发挥各自的优势，弥补单一数据源的不足，进而提升分类的鲁棒性和精度。这种多源数据融合方法，尤其在处理复杂地形、环境条件差异大或目标类别多样的场景中，展现了极大的潜力。

数据融合通过结合不同传感器的数据，可以有效克服单一数据源的局限性，进而提升分类精度。常见的数据融合方法主要可以分为三类：像素级数据融合、特征级数据融合、决策级数据融合。

像素级融合能够提高图像细节的表现，尤其是在纹理信息和边界检测方面。例如，将 SAR 数据与高分辨率光学数据（如 SPOT 或 QuickBird）结合，能够增强地物的边界和纹理特征，提高分类的精度。

特征级融合是通过从光学和雷达数据中提取特征（如纹理、形状、光谱特征

等），并将这些特征合成新的特征向量进行分类。与像素级融合相比，特征级融合能更好地利用不同数据源的互补信息。

在决策级融合中，光学数据和雷达数据分别经过独立分类，得到各自的分类结果。然后，通过投票、加权平均或其他决策规则合成最终的分类结果。这种方法的优点是可以最大程度地利用不同分类器的优势，适应不同数据源对分类精度的贡献。

3.3.3 数据变换技术

数据变换技术如主成分分析、光谱混合分析等，能够减少数据冗余并提取关键特征。这些方法被广泛用于高光谱数据分类，尤其是在植被和土地覆盖分类中表现出色。

主成分分析（PCA）是一种线性数据降维技术，常用于图像数据的预处理阶段。其基本思想是通过将高维数据投影到一个新的坐标系中，保留数据中方差最大的方向，从而减少数据的维度。通过 PCA 转换，多个高相关的特征可以被合并成少数几个不相关的主成分，其中大多数信息都保留在前几个主成分中。这一过程有效地减少了数据冗余，简化了后续的分类工作。

3.4 挑战与未来研究方向

尽管图像分类技术已取得显著进展，但仍面临一些挑战。首先，混合像素问题在低分辨率图像中仍然影响分类精度，尤其是在多类地物混合的情况下。其次，分类器融合规则的设计尚不完善，如何最大化各分类器的优势仍是一个难点。此外，分类精度评估方法需要进一步改进，以适应不同应用场景的需求。

未来的研究应聚焦于几个方面：一是开发更加高效的非监督分类算法，减少对标注数据的依赖；二是加强多源数据融合，特别是异质数据的整合，提升分类精度；三是优化分类后处理技术，减少分类噪声并提高实际应用效果。这些研究将有助于推动图像分类技术在各领域的应用和发展。

4 机器学习特征融合

4.1 特征融合的背景与意义

特征融合是机器学习与图像分类领域的重要研究方向，其核心思想是将多种特征表示形式结合，从而最大化利用每种特征的优势，提高分类精度和鲁棒性。相比单一特征方法，融合特征能够更好地处理数据复杂性与高维特性，在遥感影像、医学图像、以及环境监测等应用中具有广泛前景。

特征融合通常分为早期融合与后期融合两种主要形式：

早期融合：在特征提取阶段将多种特征连接为一个联合特征向量，作为分类器的输入。

后期融合：训练多个分类器分别处理单一特征，并在决策阶段对分类结果进行加权平均、投票或其他规则组合。

随着深度学习的兴起，特征融合技术进一步发展，通过多模态数据的融合（如光学数据与雷达数据）或多任务学习的结合，有效提升了分类性能。

4.2 特征融合方法

4.2.1 早期融合

早期融合将多种特征直接连接为一个联合特征向量。其优点是可以充分利用多种特征的相关性，在单一分类器内完成分类任务。例如，将颜色直方图（全局特征）与 SIFT 特征（局部特征）拼接后输入到支持向量机（SVM），可以显著提高分类精度。

然而，早期融合也存在一定局限性：

当特征维度较高时，联合特征可能导致维度灾难，增加计算复杂度。

不同特征之间可能存在相关性冗余，影响分类器性能。

4.2.2 后期融合

后期融合方法通过独立训练多个分类器分别处理单一特征，再将分类结果综合为最终决策。常见的后期融合方法包括：

加权平均：根据每个分类器的性能分配不同权重，对分类概率进行加权求和。

投票机制：通过分类器输出的多数表决决定最终类别。

最大概率选择：选择概率最高的分类器输出。

后期融合的优势在于灵活性，可同时利用多种分类器和特征的优势，但其效

果依赖于分类器和融合规则的选择。

4.3 特征融合实验与分析

本文实验中实现了早期融合与后期融合的特征分类方法，并与基于单一特征的分类方法进行了比较。

实验设置：

特征种类：颜色直方图（全局特征）和 SIFT + BoW（局部特征）。

分类器：支持向量机（SVM）。

数据集：包含 6 个类别的图像数据集。

结果对比：

全局特征分类准确率：50.97%

局部特征分类准确率：50.78%

早期融合分类准确率：55.44%

后期融合分类准确率：64.67%

结果分析：

单一特征的分类性能较为接近，局部特征在捕捉纹理信息方面稍显优势。

早期融合由于维度提升及特征冗余问题，其分类性能未显著提升。

后期融合通过综合全局和局部特征的分类信息，取得了显著的准确率提升。

4.4 未来研究方向

尽管特征融合技术已展现出显著优势，但仍面临一些挑战：

高维数据的有效融合：如何处理大规模高维数据的冗余与相关性问题。

融合规则的优化：设计更智能的融合策略，以最大化利用多特征和多分类器的优势。

多模态数据的深度融合：结合光学、雷达、文本等多模态数据，提升分类器的鲁棒性。

可解释性：为特征融合方法提供更好的模型解释能力，以便在实际应用中获得广泛认可。

特征融合的研究与实践将进一步推动机器学习在复杂图像分类任务中的应

用。

5 超参数调整

我们首先尝试了对超参数进行合理的改进，我们使用了线性衰减的 lr scheduler，从 5e-5 不断衰减到 0，并在前 5%的 steps 中 warmup 不断从 0 线性增长到 5e-5,然后训练 3 个 epoch 来实现恰好基本完全收敛与最佳的验证效果。实验结果如表 5.1 所示。其中 vit_b_16 最终表现最好，googlenet（inception v1）上升幅度最大。

模型	ACC(%)	F1(%)
resnet152	98.11(↑0.89)	98.11(↑0.88)
densenet201	98.22(↑0.78)	98.22(↑0.78)
alexnet	90.94(↑1.05)	90.98(↑1.05)
googlenet	93.89(**↑3.61**)	93.89(**↑3.61**)
vgg19_bn	97.39(↑0.11)	97.39(↑0.11)
vit_b_16	**98.28**(↑1.00)	**98.28**(↑0.99)

表 5.1 超参调整后各深度学习模型表现（lenet 不考虑，后续同理。括号中提升值基于 PJ4 的结果）

6 数据增强

在深度学习领域，数据增强(Data Augmentation)是一种非常重要的技术。它通过对训练数据进行一系列的变换，生成新的训练样本，从而扩充训练集的规模。这对于提高模型的泛化能力，减少过拟合现象具有重要意义。尤其是在训练数据量较小的情况下，恰当地使用数据增强可以显著提升模型性能。

传统的数据增强方法，如随机裁剪(Random Crop)、随机翻转(Random Flip)、随机旋转(Random Rotation)等，虽然可以一定程度上扩充数据集,但仍然存在一

些局限性:

（1）增强方式单一：传统方法通常只使用一种或几种固定的变换，难以覆盖真实场景中的多样性。

（2）参数选择困难:不同的数据集和任务，可能需要不同的增强参数。手动调整这些参数非常耗时且需要领域知识。

（3)增强强度不够:为了避免过度失真,传统方法的变换幅度往往较为保守，导致增强效果有限。

因此 2018 年，google 便提出了一种 AutoAugment[10]的方式，用一个极其简单的分类器，辅以网格搜索数据增强方法的方式，而其缺点也是明显的:（1）大规模采用这样的方法会增加训练复杂性、加大计算成本。（2）无法根据模型或数据集大小调整正则化强度。

于是 2019 年，google 提出了改进的方法 RandAugment[9]，研究表明只需要包含的总的增强方法够多够广，进行随机增强的效果反而会比 AutoAugment 中提出的方法更好。而且这样是十分高效的。具体来说，一个高维度视角的代码可以如图 6.1 所示。

```
transforms = [
'Identity', 'AutoContrast', 'Equalize',
'Rotate', 'Solarize', 'Color', 'Posterize',
'Contrast', 'Brightness', 'Sharpness',
'ShearX', 'ShearY', 'TranslateX', 'TranslateY']

def randaugment(N, M):
"""Generate a set of distortions.

   Args:
     N: Number of augmentation transformations to
         apply sequentially.
     M: Magnitude for all the transformations.
"""

  sampled_ops = np.random.choice(transforms, N)
  return [(op, M) for op in sampled_ops]
```

图6.1 高维度视角的 RandAugment 代码

总体来说就是，我们总共包含了 14 种数据增强方法。可以通过超参数 N 来设置一张图片使用 N 个增强方法的组合，可以通过超参数 M 来设置数据增强方法的强度。具体实验中，我们采用 RandAugment 在 imagenet 实验结果中推荐的 N=2、M=9。然后我们对训练集扩充了一倍，每张图像都增强了一张，即从 6000 张扩充到了 12000 张。或许进一步扩大规模对实验结果可能会有进一步的提升，时间有限，我们便仅扩充了一倍。数据增强对比图如图 6.2 所示，数据增强后的实验结果在表 6.1 中。其中 vit_b_16 表现最好，googlenet 上升最明显。

图6.2(a) 数据增强前

图6.2(b) 数据增强后

图 6.2 RandAugment 数据增强示例

模型	ACC(%)	F1(%)
resnet152	98.28(↑0.17)	98.28(↑0.17)
densenet201	98.44(↑0.22)	98.44(↑0.22)
alexnet	91.67(↑0.73)	91.70(↑0.72)
googlenet	95.83(**↑1.94**)	95.84(**↑1.95**)
vgg19_bn	97.83(↑0.44)	97.84(↑0.45)

vit_b_16	**98.61**(↑0.33)	**98.61**(↑0.33)

表 5.1 数据增强后各深度学习模型表现（括号中提升值基于第 5 节超参调整的结果）

7 深度学习特征融合

我们复现了作业要求中的全部方法的融合思想，其中[8]和[4]本质是完全一致的，使用交叉注意力+残差的方法，没有重复复现。我们对的融合总体上来说，使用两个 backbone encode 的特征融合时，我们选取第五节中表现最好的 vit_b_16 和 densenet201。特别地，比如 decision fusion 类似挂 adapter 的思想把 decision 模块融合进去，这种就只需要一个 backbone，我们就只选取 vit_b_16。首先，这里先汇报复现的最终结果，然后在接下来的小节中我们将一一讲解细节，最后我们将汇报尝试通过 hard voting 尝试进一步集成的实验。复现结果如表 7.1 所示。

模型	ACC(%)	F1(%)
Point Fusion	98.67(↑0.06)	98.67(↑0.06)
GLIP	**99.06(↑0.45)**	**99.06(↑0.45)**
Decision Fusion	98.61(-0.00)	98.61(-0.00)
WFCG	98.50(↓0.11)	98.50(↓0.11)
MVC	98.17(↓0.44)	98.17(↓0.44)

表 7.1 深度学习特征融合方法表现（括号中变化值基于融合中采用的最好的 backbone 即 vit_b_16 在第 6 节中的实验结果）

7.1 Point Fusion

Point Fusion[3]是一个用于目标检测任务的模型。他将 3D 点云图像和 RGB 图像分别提取的特征融合起来，其中融合的过程相对简单，只是一个简单的拼接。

如图 7.1 所示。

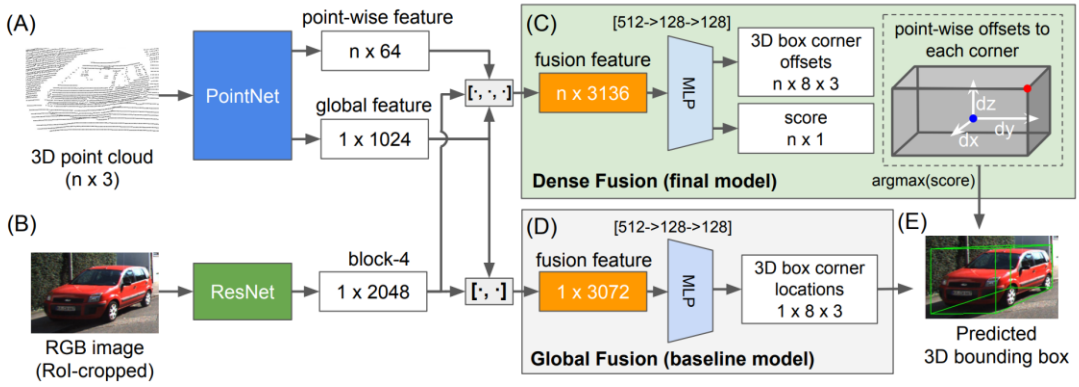


图7.1 Point Fusion 模型结构图

我们仿照它的思想，将 vit_b_16 和 densenet201 在输出层前一层的输出作为 vision encode 的结果向量，然后直接拼接来实现融合。然后后面仿照其思想接一个 MLP。MLP 输入维度 1920+768，输出维度 6，中间层维度我们尝试了 768、512、256。中间层激活函数我们尝试了 silu（swish）、GELU、SwiGLU。最终实验表明中间层维度 512、激活函数 GELU，可以带来最好的结果，准确率相比单独的 vit_b_16 上升了 0.06%。根据经验，CV 中往往使用 swish，传统 NLP 如 BERT 中往往使用 GELU，现代大模型预训练中往往使用 SwiGLU，会比较好。也许注意力机制的结构搭配 GELU 是最好的选择，而我们这里使用到了 vit，可能因此导致 GELU 带来了最好的效果。中间层维度也是一个关键所在，在这里输入输出维度差别很大，一个平稳的下降，可以取得最好的拟合效果。在后续的融合中，我们采用的 MLP 均基于本实验的结果。

7.2 GLIP

GLIP[4]是一个目标检测，并通过提示词完成自然语言标注的任务。如图 7.2 所示。

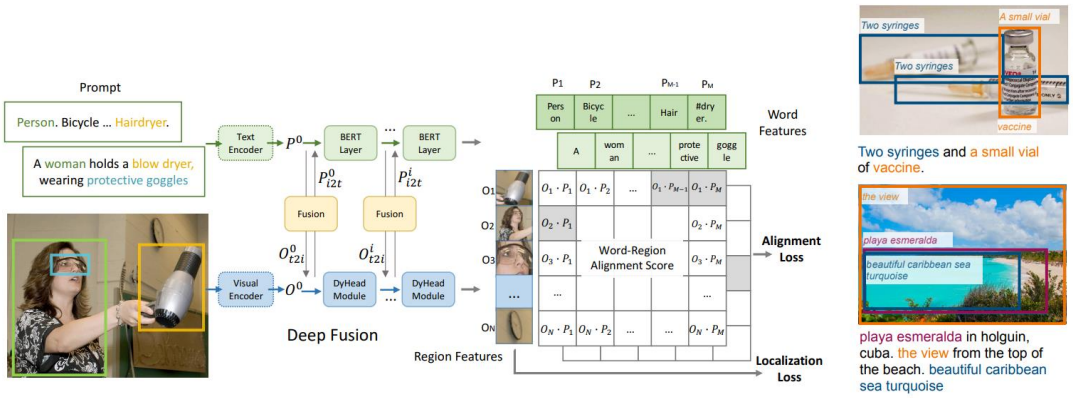


图7.2 GLIP 模型结构图

图中 text encoder 和 visual encoder 是自然语言/视觉分别的两个 backbone，然后 encode 的结果通过交叉注意力+残差互相融合，在图中明确用 Fusion 标注。然后它在后续的 BERT layer/Dynamic Head 的每一层的结果都做这样一个融合。由于任务不同，以及规模较小，我们这里就对 vit_b_16 和 densenet201 的 encode 结果做一次交叉注意力+残差后直接拼接并接一个 MLP。

对于交叉注意力的过程，以 GLIP 为例，具体来说如下公式所示：

计算交叉注意力的过程：

$$O^{(q)} = OW^{(q,I)} \tag{7.1}$$

$$P^{(q)} = PW^{(q,L)} \tag{7.2}$$

$$Attn = O^{(q)}(P^{(q)})^{\top}/\sqrt{d} \tag{7.3}$$

$$P^{(v)} = PW^{(v,L)} \tag{7.4}$$

$$O_{\text{t2i}} = \text{SoftMax}(Attn)P^{(v)}W^{(out,I)} \tag{7.5}$$

$$O^{(v)} = OW^{(v,I)} \tag{7.6}$$

$$P_{\text{i2t}} = \text{SoftMax}(Attn^{\top})O^{(v)}W^{(out,L)} \tag{7.7}$$

应用到各层我们可以简单写作：

$$O^{i}_{\text{t2i}}, P^{i}_{\text{i2t}} = \text{X} - \text{MHA}(O^{i}, P^{i}), i \in \{0,1,..,L-1\} \tag{7.8}$$

然后将交叉注意力残差连接：

$$O^{i+1} = \text{DyHeadModule}(O^i + O^i_{\text{t2i}}), O = O^L \tag{7.9}$$

$$P^{i+1} = \text{BERTLayer}(P^i + P^i_{\text{i2t}}), P = P^L \tag{7.10}$$

本质上 MLZSL[8]一文中在交叉注意力层面上，融合方法是完全一致的。MLZSL 一文是在 zero-shot，无标签的前提下，将各种各样的标签 embedding 并且引入噪声，然后通过交叉注意力机制进行融合实现图像类别的标注。其中 Oa 和 Of 的融合本质和 GLIP 一文是完全一致的。MLZSL 的结构图如图 7.3 所示。

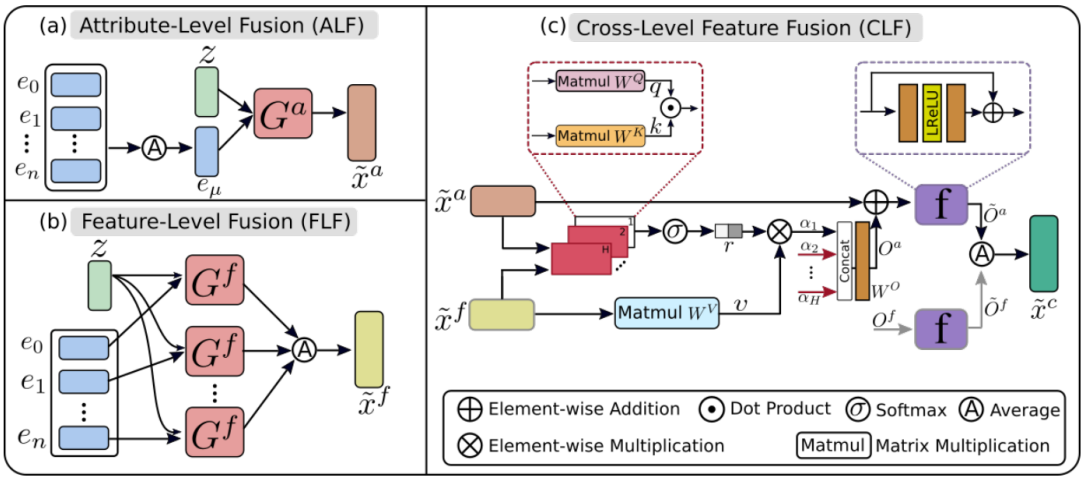


图7.3 MLZSL 模型结构图

交叉注意力近些年在各种各样的文章中广泛应用，确实是一个效果极佳的常见融合方法。任何时候都值得尝试，在本实验中，该方法也是取得最好表现的方法，在单独的 vit_b_16 的基础上再度提升准确率 0.45%达到了极高的 99.06%。

7.3 Decision Fusion

Decision Fusion[5]有些类似挂 adapter，它提出一个 Decision Fusion Module（DFM）的辅助网络，在 backbone 的每个 block，我们可以立刻对特征做 global average pooling（GAP）得到池化向量然后立刻做一个类似输出层的行为，映射到 num_classes 维，称作 make a decision，然后在立刻 expand

到和原特征高宽一致，在通道维拼接回去，继续传给后续的层。形象化来看，如图 7.4 所示

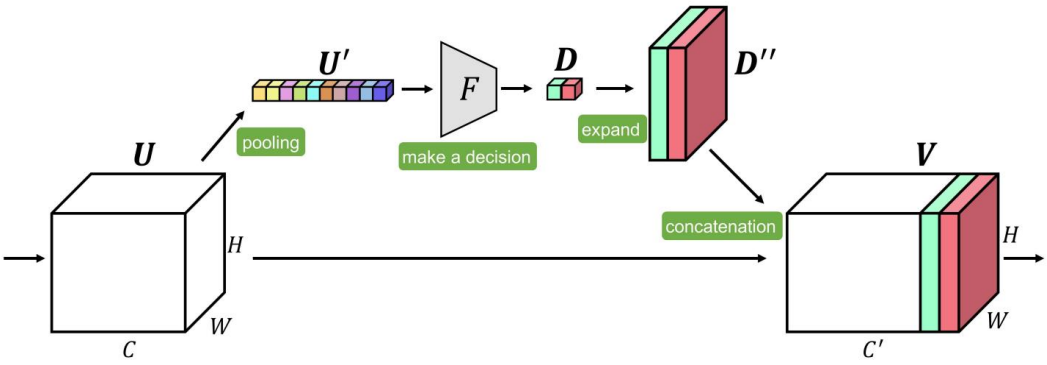


图7.4 Decision Fusion Module 结构图

在本实验中，具体来说我们对 vit_b_16 引入这种辅助网络。在 12 层 encoder layer，每一层我们取出[CLS]位置的向量，然后用一个 MLP make a decision 然后 expand 并立刻拼接到 length 维（或说 channel 维）的末尾，继续传给下一层 encoder layer。更具体地，如果你了解 BERT 的话，我们仿照 bert pooler output 的思想，本质上要区分[CLS]处的输出和其他位置处的输出，bert pooler output 中等特征长度前馈并采用 Tanh 激活以区分，这是我们这里 decision 中 MLP 的处理。当然，对于输出层的 MLP 我们依然采用 GELU 激活，因为只使用[CLS]处的输出向量，不涉及其他位置的使用与区分。

但是十分可惜的是，Decision Fusion 的实验结果并没有带来任何提升，但是也没有下降。

7.4 WFCG

如图 7.5 所示是 Weighted Feature Fusion of Convolutional Neural Network and Graph Attention Network (WFCG)[6]的模型结构图。它将 GNN 和 CNN 的特征进行了加权融合。

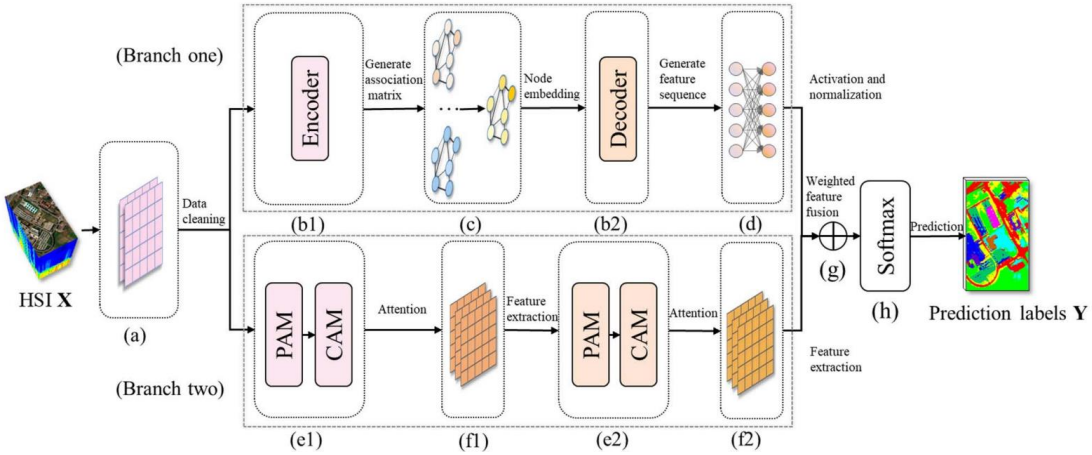
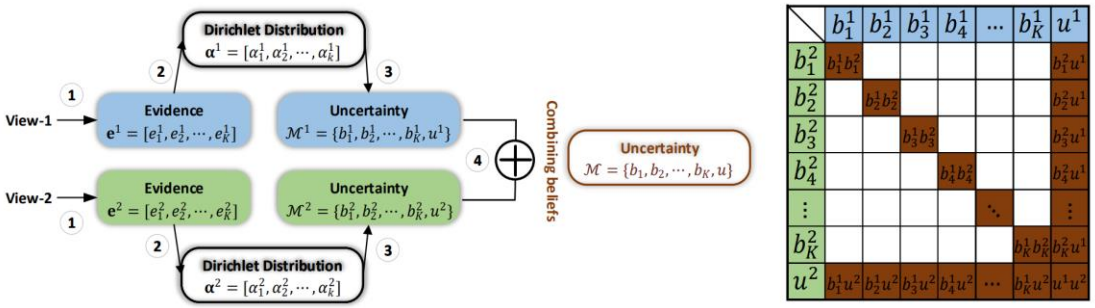


图7.5 WFCG 结构图

这个模型中，融合的思想实质非常简单，只是引入了一个超参数 w，两个 backbone 的 vis encode 分别乘以 w 和 1-w 进行加权。在本实验中，我们从一个更加简单的角度和方式完成这项任务，我们定义了一个 w=nn.Parameter()，初始化为 0.5，然后来通过梯度下降来自动优化这个 w，实现本任务中的加权。但是很不幸的是，融合的测试结果，准确率比 vit_b_16 要低，恰好是 densenet201 和 vit_b_16 的平均值。我们认为这是一个非常中庸的方法，意义不是很大，甚至未必比直接拿着多个模型预测结果投票要好。

7.5 MVC

Trusted Multi-view classification（MVC）[7]是一个非常有意思也有实现难度的网络，它的思想是通过狄利克雷分布和 Dempster-Shafer 证据融合理论的结合，将证据置信度作为打分的预测方式。具体来说，它的结构图如图 7.6 所示。



(a) Overview of the trusted multi-view classification (b) Combining beliefs

图7.6 MVC 结构图

接下来我们来具体解释一下。什么是狄利克雷分布。当只有两个类别时，它退化为 Beta 分布，我们首先理解什么是 Beta 分布

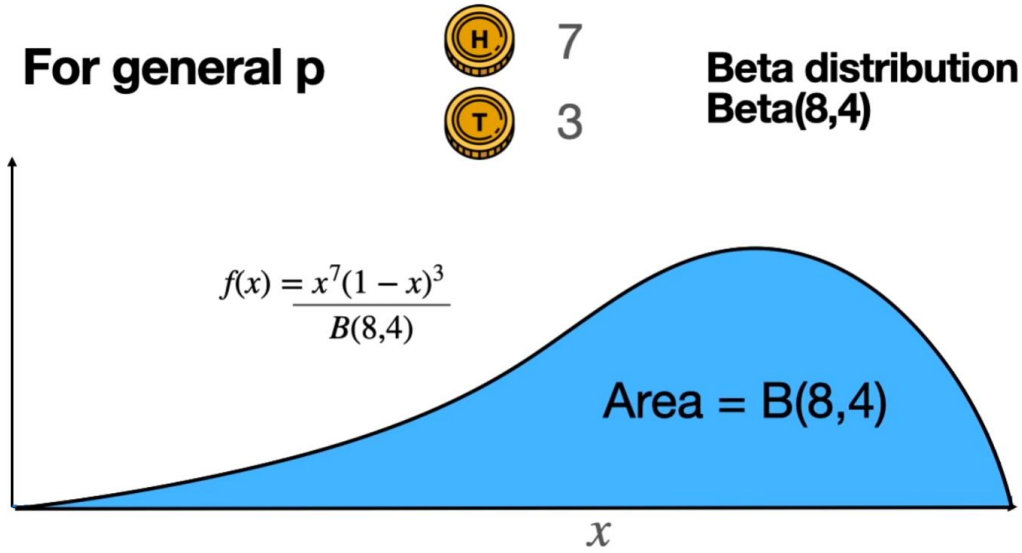


图7.7 Beta 分布示例图

例如。假设扔硬币，有两个类别，即正面朝上和反面朝上，设其概率分别为 x 和 1-x，那么，假设 7 个正面 3 个反面，这时候概率就是$x^7(1-x)^3$。如果我们固定 x，就会看成扔出几次正面几次反面是概率最佳的，指数也就成为了一种置信度，实质上它可以是>1 的正数，也可以是<1 的正数甚至是负数。但是这种情况下，比如，当我们扔硬币次数足够高，因为指数计算，那概率必然是足够低，但这并不代表我们不该有这样高的置信度。这时候我们就是，对任意 x 在[0,1]，可以求取特定置信度下的积分，那么就可以通过除以这个积分来归

一化的方式，让指数真正具有置信度的能力。这就是 Beta 分布。而狄利克雷分布，就是扩展到多个类别的情况，就比如二项分布和多项分布的差异。狄利克雷分布相比于普通的概率，从单纯形上来看，普通的概率是单纯形上的一个点，而狄利克雷则为单纯形上的每个点赋予了概率密度。

在本文中，我们进一步将 Dempster-Shafer 证据融合理论进行结合。置信度 b（belief）、不确定度 u（uncertainty）如下所示：

$$b_k^v = \frac{e_k^v}{S^v} = \frac{\alpha_k^v - 1}{S^v} \quad \text{and} \quad u^v = \frac{K}{S^v} \tag{7.11}$$

类别数是 K， $\sum b_k + \mathrm{u} = 1$。然后对于多视图的融合，则是完全使用 Dempster-Shafer 证据融合理论，比如两个视图，置信度和不确定度如下公式所示：

$$b_k = \frac{1}{1-C}(b_k^1 b_k^2 + b_k^1 u^2 + b_k^2 u^1), \quad u = \frac{1}{1-C} u^1 u^2 \tag{7.12}$$

其中，$C = \sum_{i \neq j} b_i^1 b_j^2$表示冲突程度。简单解释一下，冲突程度就是两个试图在各不同类别置信之积的总和，然后置信度和不确定度的计算中，用 C 进行了正则。对于置信度的计算，简单来说第 k 类的置信度，也就是两个试图都置信第 k 类的置信度之积加其中一方不确定一方置信的积的和。不确定度就是两个试图不确定度的积。

然后相比于一般的交叉熵，这里的

$$\mathcal{L}_{ace}(\boldsymbol{\alpha}_i) = \int \left[\sum_{j=1}^{K} -y_{ij} \log(p_{ij}) \right] \frac{1}{B(\boldsymbol{\alpha}_i)} \prod_{j=1}^{K} p_{ij}^{\alpha_{ij}-1} d\mathbf{p}_i = \sum_{j=1}^{K} y_{ij} \left(\psi(S_i) - \psi(\alpha_{ij}) \right) \tag{7.13}$$

简单解释一下就是，做了积分，不关心概率是多少，而是概率为单纯形上任意一点，在单纯形上进行积分，关键在于置信度，用置信度进行分类。但是这样的约束能够使得每个样本正确类别产生更多的证据，不能约束其他类别产生更少的证据，因此原文还加入了 KL 散度的惩罚。并且引入了一个超参数，随 epoch 增加从 0 线性增大直至到 1，防止网络在训练初期过多关注 KL 散度。

实验中我们直接将 densenet201 和 vit_b_16 的 encode 前馈到 numclasses=6 维，这样作为两个视图的输入 α，然后采用 MVC 中的方式。训练

过程中，初期 loss 很大，并且出现了非常大的梯度值，让我们一度怀疑不会收敛。然而幸运的是，它确实有效，能够收敛，并且测试准确率也很不错，虽然相比融合前其实是略微下降的。这个是一个偏向机器学习理论的方法，我想和这些视觉计算很强的 backbone 进行融合或许并不是一个合适的做法。我们认为 MVC 这个方法应该是很有效的，但它或许适合更多更小的分类器视图的融合，或许能够带来非常好的效果。

7.6 Hard Voting

我们对各模型实验预测的结果进行了 hard voting 集成尝试进一步提高准确率。对于有多个最多票类别的情况，我们在同票的这几个类别中固定 seed=42 进行随机分类。详细完整实验结果可以看 PJ4 代码中的 voting.ipynb。我们首先对融合前的各个模型尝试，包括 alexnet、googlenet、vgg、resnet、densenet、vit。这里仅汇报包含 vit+densenet 的集成结果。如表 7.2 所示。

模型	ACC(%)	F1(%)
vit+densene+resnet	**98.67(↑0.06)**	**98.67(↑0.06)**
vit+densene+vgg	98.50(↓0.11)	98.50(↓0.11)
vit+densene+googlenet	98.50(↓0.11)	98.50(↓0.11)
vit+densene+alexnet	98.22(↓0.39)	98.22(↓0.39)
vit+densene+resnet+vgg	**98.67(↑0.06)**	**98.67(↑0.06)**
vit+densene+resnet+vgg+googlenet	98.50(↓0.11)	98.50(↓0.11)
vit+densene+resnet+vgg+googlenet+alexnet	98.44(↓0.17)	98.44(↓0.17)

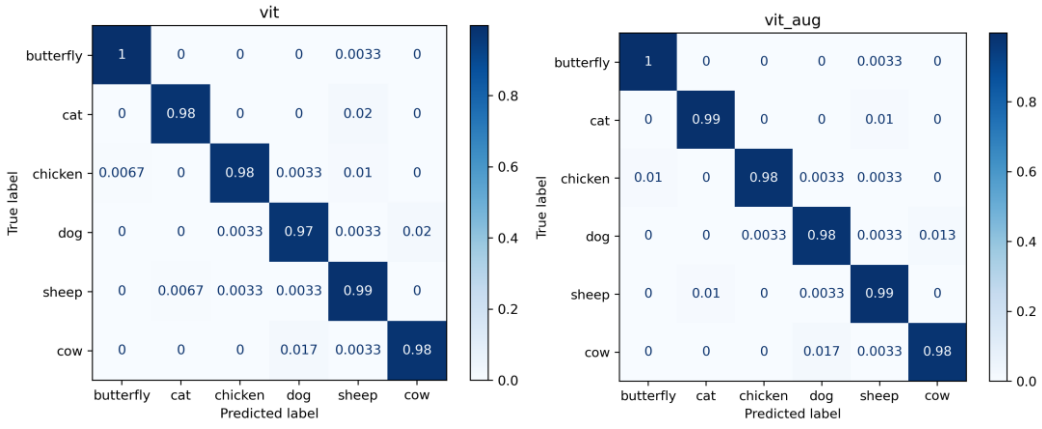
表 7.2 融合前模型 hard voting 结果（仅汇报包含 vit+densenet，其中包含二者的三个模型集成结果全部汇报，四至六个仅汇报最佳结果。变化值相比于 vit 在第 6 节的结果）

实验结果表明用多个表现同样尽可能好的分类器 hard voting 往往会带来更好的表现。表现明显更差的分类器在本实验中会拉低表现最好分类器的表现，尽管有时并不一定，如果能带来新的有价值的不同视角，有时也许可能提高表现。实验中组合的最好结果是：vit + densenet + resnet，98.67。vit + densenet + resnet + vgg，98.67。均达到了 Point Fusion 的表现。

基于上述结果，三到四个分类器组合就是充足的，以及考虑到我们的 fusion 方法已经用到 vit、densenet。所以尝试对 fusion 方法们和 vit、densenet、resnet、vgg 中，尝试组合。由于组合量过大，这里不做展示。详细内容在 voting.ipynb 中。这种情况下集成偶尔确实有一点作用，对某些方法确实会带来一定表现的提升，但总体来说效果并不是很好。各分类器往往总是拉低 GLIP 的表现，集成过后，只有 GLIP + Point Fusion + densenet 的组合达到了和 GLIP 一样的 99.06%，没有能超过 GLIP 的表现。因此本项目最终的准确率最高就是 99.06% 了。或许在我们的分类器范围内，有 0.94 的样本极难分类，所有分类器几乎都做错了。想要进一步提高表现，只能尝试扩大数据增强规模或者选用更加强大的 backbone。

8 混淆矩阵分析

我们这里对 VIT 数据增强前后以及和 GLIP 的实验结果，做了混淆矩阵分析与对比，如图 8.1 所示。



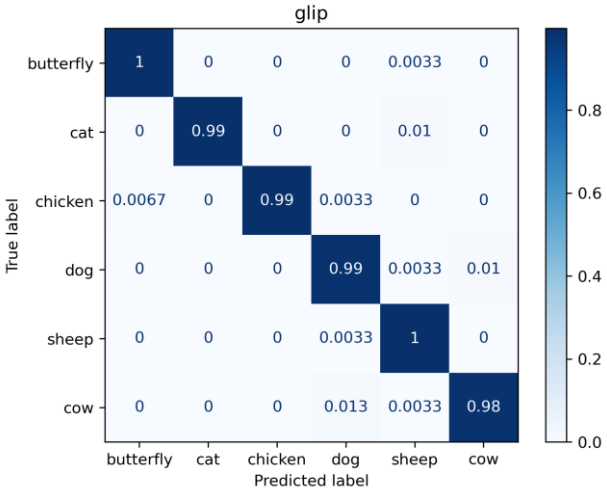


图8.1 VIT 数据增强前后、GLIP 混淆矩阵分析

最初极难分类的牛和狗，在数据增强后有了一定改善，在 GLIP 特征融合后更是有了明显改善。对羊和鸡的误分类上，也得到了一定明显改善。但是牛和狗依然是分类最大难点。

9 总结

在 PJ2、PJ3 中我们分别使用机器学习方法和深度学习方法简单完成了 Animals-10 图像分类任务。在 PJ4 中，对于机器学习，我们通过多种方法探索了图像分类优化的可能性。在 PJ2 中，基于颜色直方图提取图像的全局特征，捕捉整体的颜色分布信息；同时，利用 SIFT（尺度不变特征变换）提取图像的局部特征，并通过 Bag-of-Words（BoW）模型生成描述符，刻画图像的局部纹理和形状特性。在此基础上，在 PJ4 中我们尝试了特征融合，包括将全局与局部特征拼接的早期融合，以及基于分类概率加权平均的后期融合方法。通过训练支持向量机（SVM）分类器，我们验证了这些特征及融合方法的有效性，并发现后期融合可以显著提升分类精度，为进一步提升机器学习在图像分类中的表现提供了基础。对于深度学习，我们先后通过对超参数的合理优化、复现了 RandAugment 的数据增强方法、以及尝试特征融合，最终 GLIP 的交叉注意力+残差的融合方式融合 vit_b_16 和 densenet201 取得了最好的 99.06%的准确率表现。经过分析我们认为在这些 backbone 和数据增强方法的前提下，已经基本

提升到了极限，未来或许更大规模扩充数据以及换用更加强大的 backbone 或许可以进一步提高在该任务中的表现。

参 考 文 献

[1] Chaoqun Ma, Xiaoguang Hu, Li Fu, Guofeng Zhang, An Improved ORB Algorithm Based on Multi-feature Fusion, IEEE International Symposium on Industrial Electronics (ISIE) 2018.

[2] O. Melnik, Y. Vardi, and C.-H. Zhang, Mixed group ranks: preference and confidence in classifier combination, IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(8): 973-981, Aug. 2004.

[3] Danfei Xu, Dragomir Anguelov, Ashesh Jain, PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation, IEEE International Conference on Computer Vision and Patter Recognition, 2018.

[4] Li, Liunian Harold, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang and Jianfeng Gao, Grounded Language-Image Pre-training, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021): 10955-10965.

[5] Tang K, Ma Y, Miao D, et al. Decision fusion networks for image classification[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022.

[6] Dong Y, Liu Q, Du B, et al. Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification[J]. IEEE Transactions on Image Processing, 2022, 31: 1559-1572.

[7] Zongbo Han, Changqing Zhang, Huazhu Fu, Joey Tianyi Zhou, Trusted Multi-view Classification, ICLR2021.

[8] Gupta, Akshita, Sanath Narayan, Salman Hameed Khan, Fahad Shahbaz Khan, Ling Shao and Joost van de Weijer, Generative Multi-Label Zero-Shot Learning, ArXiv abs/2101.11606 (2021).

[9] Cubuk E D, Zoph B, Mane D, et al. Autoaugment: Learning augmentation

strategies from data[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 113-123.

[10] Cubuk E D, Zoph B, Shlens J, et al. Randaugment: Practical automated data augmentation with a reduced search space[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020: 702-703.

附　　录

提交代码说明

这里我们给出对提交代码的说明。

PJ2 的代码为 PJ2.py。

PJ3 的代码为 PJ3 文件夹中 Animals-10 文件夹。

PJ4 的机器学习部分代码为 PJ4.1.py。

PJ4 的深度学习部分代码为 PJ4.2 文件夹中 Animals-10 文件夹。

PJ4.1 在 PJ2 的基础上进行的改造。PJ4.2 在 PJ3 的基础上进行的改造。

这里对 PJ4.2 的项目结构进行说明。

由于我目前主做的方向是大模型相关的。使用的训练框架是 huggingface 的 Trainer，尽管 Trainer 暴露的接口十分糟糕，不如手写框架，甚至不如 pytorchlightning，但是它是有学习价值的，通过使用 hf 的 Trainer，熟悉它的一些参数，有助于帮忙掌握 trl.SFTTrainer，大模型的有监督微调还是用 hf 的框架更为方便许多。另外 hf 的 datasets 既方便又高效，它可以类似 numpy 绕开 GIL 锁的限制，并且会缓存一个高效的.arrow 文件，性能上本质已经取代了 torch.utils.data 的传统 Dataset，建议使用。接下来具体讲解项目结构。

train.py 是训练脚本。data_utils.py 中完成的内容是在 datasets.map 中使用的数据预处理函数。evaluate.py 是 load_best_model_at_end=True 并执行 evaluate.py 打印 best model 在测试集上的 acc 和 macro_f1。config_utils.py 是我的一个习惯，由于 parser 的 args 没有自动提示，因此我便定义一个名为 Args 的 dataclass 将 parser 的 args 解包到其中。analyse.py 是执行混淆矩阵分析保存图片。然后模型的定义全部都在 models 文件夹下。logs 文件夹下包括所有各种 log，有文本形式的记录还有 tfevents，可以运行 tensorboard 查看。outputs 文件下下是实验的参数保存（提交的代码已删除参数）和标签、预测结果、混淆矩阵分析的保存。原数据集在 data 文件夹下也 copy 一份放到 data_aug 中，rand_augment.py 是复现的 Rand Augment 方法，直接执行来实现增强，每个图片增强一次，扩充一倍训练集数据量。voting.ipynb 是对 hard voting 的

尝试，通过 dfs 网格搜索的方式组合各分类器预测结果。