基于 PCA 和 SVM 方法的猫狗分类

Jianhua Guo

摘要

实验使用 8000 张猫狗图像数据集进行二分类，在实验室电脑上实现了约 75% 的分类准确率，且运行时间不超过 4 分钟。由于实验是在非笔记本的环境下运行的，如果切换到笔记本上实际运行时间过长，可以只使用 2000 张训练集，每个分类各 1000 张图片（从实际运行结果来看，使用 8000 张图像会比只使用 2000 张图像的训练效果更好一些，但仅使用 2000 张图像也能够达到 70%-74% 之间的准确率，所以使用 2000 张同样足够了）。整个实验实现了数据加载，特征提取、PCA 降维和 SVM 分类的完整流程，其中 PCA 和 SVM 都基于 numpy、scikit-image 和 matplotlib 自定义实现。

1. 模型的基本流程和框架

基本流程步骤
1. 数据加载
2. 传感器与预处理
3. 特征提取
4. 特征拼接
5. PCA 降维
6. 训练分类器
7. 模型评估与预测

2. 为什么要改进

基本的训练模型虽然看起来很完整，但在实际的实验运行中却发现存在一些对准确度提升基本没有任何帮助、同时会大幅增加运行时间的冗余操作，例如提取 SIFT 特征或其他额外特征等：由于进行一次数据增强会将需要特征提取和训练的数据集成倍增大，所以为了满足低耗时的实验要求，我们应该提取真正有用的特征，而不能随意的提取任意多的特征，也不应该提取那些对准确率提升效果不大，但提取特征却耗时较长的特征，如 SIFT 特征，即便 SIFT 不是密集特征提取器，但一张图像可能仍有几百上千个关键点，提取 SIFT 带来的准确率提示效果无法平衡它的运行耗时，所以最好的选择是不提取 SIFT 特征。

3. 改进的思路

基本的改进思路：先从流程的每一个步骤入手，在一些细节上对算法进行优化，此外，还可以从整体上对实验框架和流程进行优化，比如使用一些常规的训练手段达成提高准确率的目的。

4. 具体改进

4.1. 数据加载阶段

传统图像加载通常将图像调整为固定尺寸，但往往忽略抗锯齿处理，导致调整后的图像可能出现锯齿状边缘或细节失真，影响图像质量。代码中使用 transform.resize 抗锯齿技术，通过设置 anti_aliasing=True，在调整图像尺寸时平滑像素间的过渡，减少失真，从而显著提升图像质量。

此外，代码在加载图像时引入了随机水平翻转增强，具体通过以下逻辑实现：

```
if np.random.rand() > 0.5:
    img_flipped = np.fliplr(img)
```

每次加载图像时，有 50% 的概率生成一张水平翻转的图像，并将其加入数据集。这种实时增强增加了数据的多样性，有效减少模型过拟合的风险。

4.2. 特征提取阶段

传统方法中常使用耗时的 SIFT 特征，而本方法选择更加简单易处理的 HOG、LBP 和颜色直方图特征，并将三种特征拼接为一个多维特征向量：

$$\text{np.concatenate}([\text{hog\_feat}] + \text{lbp\_feats} + [\text{color\_hist}])$$

这种多特征融合充分利用了图像的不同方面（结构、纹理和颜色），提升了分类器的判别能力。

传统 HOG 特征提取通常使用默认参数，且忽略颜色信息。改进后通过优化超参数，在多个细节上提升了特征的质量，例如增加方向细粒度，采用 L2-Hys 归一化，指定 channel_axis=-1，在 RGB 三通道上分别计算特征。

对于 LBP 特征，使用 RGB 多通道计算。选择超参数 P=16, R=2 和 method='uniform' 平衡特征丰富性和计算复杂度。

最后，颜色直方图捕捉全局颜色分布信息，弥补了 HOG 和 LBP 在颜色信息上的不足。

4.3. 新增：标准化处理阶段

由于特征提取后，不同特征的数值范围可能差异很大，需要进行标准化处理消除这种差异，于是我们新增一个标准化处理函数，对特征矩阵进行标准化。

4.4. PCA 降维阶段

手动实现 PCA 降维功能，在多个方面进行了改进，具体如下：

- **超参数选择**：
  - variance_ratio=0.99 提供自适应主成分数量选择，相比直接指定维度，这种方式更直观地反映了降维的质量。
- **算法逻辑**：
  - 使用 np.linalg.eigh 提高特征值分解的效率和稳定性。
  - 降序排序确保主成分对应最大方差方向，通过 np.argsort(eigvals)[::-1] 实现高效排序。
  - 结合标准化输入，确保协方差矩阵计算的公平性，避免特征尺度差异对 PCA 结果的干扰。

4.5. SVM 训练阶段

Algorithm 1 SVM 伪代码及实验主流程

```
类 LinearSVM:
    初始化 lr=1e-4, batch_size=128, epochs=100, patience=10
    function fit(X, y, X_val, y_val)
        for 每个 epoch do
            分批次更新 W 和 b 使用 SGD
            if 验证准确率未提升 patience 次 then
                提前停止                          ▷ 早停机制
            end if
        end for
    end function
    // 主流程
    function main
        加载数据
        提取特征（HOG + LBP + 颜色直方图）
        标准化
        PCA 降维
        K 折交叉验证训练 SVM
        测试并可视化
    end function
```

手动实现线性 SVM 分类器，在多个方面进行了改进，具体如下：

- **超参数选择**：
  - lr=1e-4、reg=1e-3、batch_size=128、epochs=100 和 patience=10
  - 使用线性核（无核超参数）gamma，通过交叉验证从 {1e-3, 1e-2, 1e-1, 1} 中选择。
  - SVM 的正则化参数 reg=1e-3 控制模型复杂度和泛化能力，学习率 lr=1e-4 较小以确保收敛稳定性，迭代次数 epochs=100 结合早停机制动态调整。

表 1. 消融实验：准确率以及训练和测试的耗时对比

实验设置	学习率	图片数量	测试准确率	耗时 (秒)
未优化	5×10^{-5}	2000	0.5185	17.94
	1×10^{-4}	2000	0.5285	17.93
	5×10^{-5}	8000	0.5125	71.42
	1×10^{-4}	8000	0.5680	71.48
优化后 (5 折交叉验证)	-	2000	0.7379	103.40
	-	8000	0.7522	204.23

- **算法逻辑**：
  - 引入早停机制，动态调整训练轮数，防止过拟合。当验证集准确率连续 patience 次未提升时停止训练。
  - 使用 SGD 和 mini-batch 优化，提高计算效率。相比传统全批量梯度下降，mini-batch 方法降低了计算复杂度。
  - 精确计算 hinge loss 梯度，减少无效更新，仅对违反间隔条件的样本更新梯度。
  - 简化预测逻辑，提高推理效率，通过 X @ self.W + self.b > 0 直接实现二分类。

5. 实验结果

为了证明改进的有效性，设计了四组未优化的消融实验，以及两组优化后的对比试验，实验结果如表一所示，所有实验均在 5 分钟运行结束，且优化后的实验中，无论数据集大小是 2000 还是 8000，测试集上的准确率均达到了 0.7 以上。

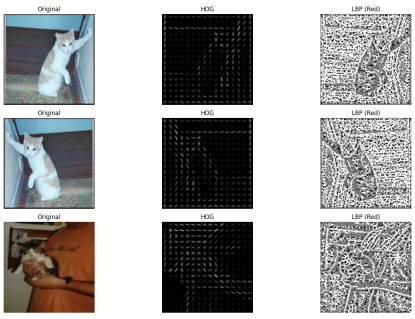

图 1. HOG 特征和 LBP 特征

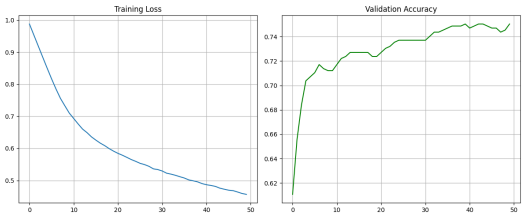

图 2. 训练集为 1000 张图像的损失和准确率曲线

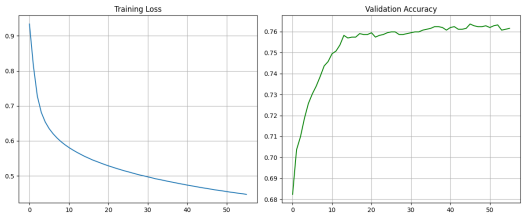

图 3. 训练集为 8000 张图像的损失和准确率曲线

6. 结论：仍存在的局限性

当前代码仅使用手工设计的特征（HOG、LBP、颜色直方图），这些特征虽然能捕获图像的某些局部信息（如梯度、纹理、颜色分布），但无法提取更高层次的语义特征。此外，线性 SVM 只能学习线性决策边界，而猫狗分类任务中，特征空间可能高度非线性。这些局限性导致模型性能未达到最优，优化后的测试准确率仍有提升空间。