数据挖掘课程设计报告

姓名: 姚远　　学号: 162150109

日期：2023 年 11 月 10 日

1 任务概况

1.1 数据集

本课设数据集来自于 Kaggle, 为树叶分类数据集. 该数据集包含 176 类不同的树叶, 18353 张训练样本 (每种树叶至少 50 个样本, 长尾效应并不明显) 以及 8800 张测试样本. 分辨率皆为 224×224. 可以发现这是一个简单的图片分类任务.

1.2 评测指标

Kaggle 官方直接采用 Classification Accuracy 作为竞赛评测标准. 一般在数据类别均衡的情况下, 模型的 acc 越高, 说明模型的精度越好. 本数据集由于类别很多, 数据分布较为平均 (长尾比约 3:1), 因此不考虑通过计算每个类别的 recall 值来评测模型.

1.3 解决方案

事实上这是一个课程网站的 project, 该课程是深度学习相关课程, 因此如果你去查看 Code 就会发现清一色的 Resnet 与其它深度学习方法.

本课设大致分为传统机器学习和深度学习两种方案来解决问题.

传统机器学习解决的过程大致包括：

1. 训练集验证集划分: 考虑到传统机器学习方法性能较优, 故采用 5-Fold Cross Validation, 训练验证比 4:1;
2. 数据预处理: 包括特征提取 (SIFT、HOG etc.)、动态聚类 (Kmeans)、降维 (PCA) 等, 将数据转化为统一的向量形式;
3. 模型分类: 通过采用不同的机器学习分类模型 (KNN、SVM etc.) 来测试分类效果.

深度学习的过程大致包括:

1. 找张 GPU: 使用 Autodl 上的 NVIDIA GTX 2080ti(11GB) 与校内 GPU 服务器的 NVIDIA RTX A5000(24GB) 训练深度学习模型;
2. 训练集验证集划分: 性能原因, 不考虑使用 k-fold, 直接划分. 训练验证比 4:1;

3. 数据预处理: 包括一系列的数据增强 (随机翻转、标准化) 以及采用跨图片增强 (训练过程中体现) 的 trick: MixUp、CutMix;
4. 训练验证模型: 对比采用不同的跨图片增强 trick 训练出来的深度学习模型 (resnet 及其变种) 在验证集上的效果.

1.4 DummyClassifier

直接采用 DummyClassifier 作 5 折交叉验证, 预测结果为训练样本中出现次数最多的类, 得出最优的 acc 为 0.036.

2 传统机器学习模型

2.1 特征提取

2.1.1 SIFT 特征提取

SIFT(Scale-invariant feature transform) 中文名为尺度不变性特征变换, 在传统的 CV 算法中拥有很高的地位. 作为一个特征点的提取算法, 它对尺度、旋转、关照等变化不敏感, 导致其效果较优.

首先我们有必要弄明白特征点: 对于平滑的区域一般变化不大, 这不是我们关心的地方, 我们关心的是纹理复杂的地方, 例如边缘、点、角之类的区域, 而这些灰度值变换大的地方就是我们要的特征点.

算法的第一步是数据预处理. 由于彩色图是三通道的, 我们需要先将其转化为灰度图, 此时灰度图为单通道, 灰度值应该在 0-255 之间分布.

算法的第二步是构建多尺度 DoG 空间. 这里的多尺度代表对不同分辨率的图片作空间构造. 对于单个分辨率的图片, 分别作六次不同方差高斯模糊后的图像:

$$\sigma, k\sigma, k^2\sigma, k^3\sigma, k^4\sigma, k^5\sigma.$$

对于特定的方差 σ, 高斯模糊后的图像的空间灰度函数为:

$$G(x_i, y_i, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x-x_i)^2 + (y-y_i)^2}{2\sigma^2}\right)$$

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y).$$

然后需要获取高斯差分函数 (DoG). 高斯差分图像是某一相同分辨率的相邻图像作差值得出, 然后与原图像 $I(x, y)$ 作卷积得到 DoG 函数:

$$\begin{aligned} D(x, y, \sigma) &= [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}$$

这样, 对于不同的分辨率, 我们有 5 个 DoG 函数. 事实上这就是高斯金字塔表达, 对于不同的尺度 (分辨率), 从下到上依次降采样, 长的很像一个金字塔.

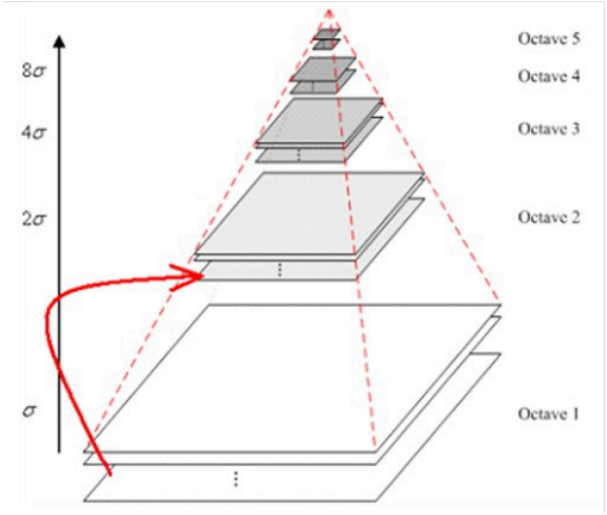

图 1: 高斯金字塔

算法的第三步是极值点检测. 寻找 DoG 函数的极值点, 最简单的想法是直接和附近点相比较. 在 SIFT 算法中我们还需要对同一组中上面和下面不同的 DoG 函数的点相比较, 这样一共需要比较 $8+9\times 2=26$ 次, 而对于每组 5 个 DoG 函数的情况, 我们只能对中间 3 个函数进行操作.

然而这种方法找到的点的位置都是在整数空间上的. 换句话说, 这些极值点都是离散的. 因此 SIFT 算法考虑通过插值法对离散的 DoG 函数进行曲线拟合, 进一步对方程求偏导, 得到精确的极值点.

此外算法还提到了删除边缘效应的点的概念, 这里不予赘述.

算法的最后一步是产生特征点描述. 上面的过程产生的特征点只有位置信息, 我们需要通过这些特征点附近区域获取特征. 算法在检出的特征点为中心选 16×16 的空间作为特征提取空间, 然后将这些区域均分为 4×4 个子区域.

对于每个 4×4 的子区域, 计算 8 个方向的梯度方向直方图. 首先我们计算每个像素的梯度的幅值和方向, 然后对子区域进行统计, 统计 8 个方向的幅度. (类似 HOG 的特征提取)

幅度公式:

$$m(x,y)=\sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2}.$$

方向公式:

$$\theta(x,y)=\arctan\left((L(x,y+1)-L(x,y-1))/(L(x+1,y)-L(x-1,y))\right).$$

然后将所有子区域的幅度直方图连接起来, 即可获得特征向量, 其维度为 $4\times 4\times 8=128$ 维.

2.1.2 HOG 特征提取

HOG(Histogram of Oriented Gradient) 中文名为方向梯度直方图, 相比于晦涩难懂的 SIFT, 这个算法更加直接, 直接在原图的基础上产生特征向量. HOG 通过计算并统计图像局部区域内的梯度方向直方图 (SIFT 部分已经介绍过了) 来形成特征, 这导致它对图像几何和光学的变化不敏感.