CUB-200图像分类任务书

一、总体框架

1.读取数据集：生成10个样本集的特征张量和标签

2.预处理: Canny Gray图 边缘检测

Gabor Gray图 提取纹理特征

HSV HSV 1通道 提取颜色特征

将以上3种图像特征拼接成特征图，用Haar矩形窗口在特征图上卷积生成特征张量

3.分类器模型

Classifier类: 随机选择特征，根据阈值分类，存储feature\_mask、阈值和自身权重

AdaBoost类: 弱分类器串行集成强二分类器

Detector：实现端到端的功能，输入一张图，输出概率最高的类索引

4. Torch-convolution 与机器学习算法对比

二、各模块原理及拆分函数

1. 读取数据集

1.1创建SingleClass类：

根据路径读取图片->保存每类高斯滤波后的HSV、gray数据

1.2创建Dataset类

1.2.1初始化：

生成10个SingleClass的特征张量

读取全部数据->归一化处理->Canny，Gabor，HSV拼接成特征图->harr生成特征张量

1.2.2生成采样集：

10个鸟类集互为正负样本

负样本从其余类中抽取->保存正负样本的特征向量和标签->打乱

1.2.3数据流：

类图像数据:

|  |  |
| --- | --- |
| image\_HSV | batch\_size\*1\*128\*128 |
| Canny处理灰度图 | batch\_size\*1\*128\*128 |
| image\_gray | batch\_size\*128\*128 |
| Gabor处理灰度图 | batch\_size\*4\*128\*128 |
| HSV 与上述数据拼接 | batch\_size\*6\*128\*128 |
| Haar输出特征张量 | batch\_size\*6\*8\*16\*16 |

2. 预处理

2.1 Canny边缘检测

2.1.1原理：

灰度图局部梯度变化大的点即为边缘点

2.1.2实现：

高斯滤波降噪（读取数据时已处理）->sobel算子卷积计算x、y方向梯度->NMS过滤非边缘点->八联通连接强弱边缘->批处理

2.1.3数据流：

批灰度图 batch\_size\*128\*128 -> 边缘特征图 batch\_size\*1\*128\*128

2.2 Gabor提取灰度图纹理信息

2.2.1原理：

2维Gabor滤波器是一个正弦平面波和高斯核函数的乘积，正弦平面波将图像的灰度分布函数变换为图像的频率分布函数，为了提取局部信息，又引入了时间局部化的高斯核函数。通过设置函数参数，可以在频域的不同尺度、不同方向上提取相关特征。

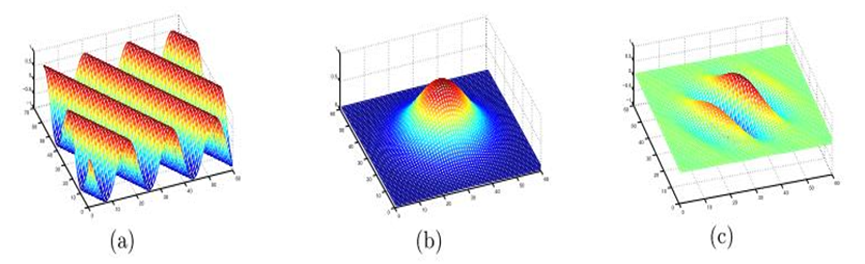
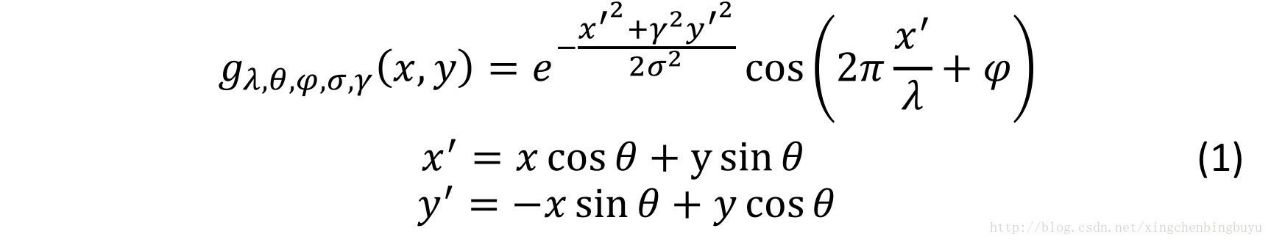


图1 Gabor函数示意图



2.2.2实现：

Gabor核函数->设置尺寸和核方向参数生成4种Gabor滤波器->卷积->批处理

2.2.3数据流：

批灰度图 batch\_size\*128\*128 -> 纹理特征图 batch\_size\*1\*128\*128

2.3 HSV提取颜色特征

2.3.1原理：

HSV即色相、饱和度、明度，HSV是一种将RGB色彩模型中的点在圆柱坐标系中的表示法。

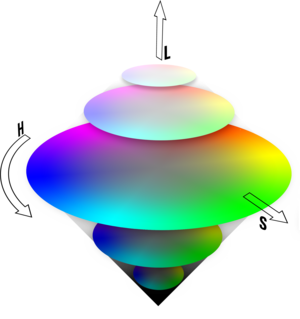
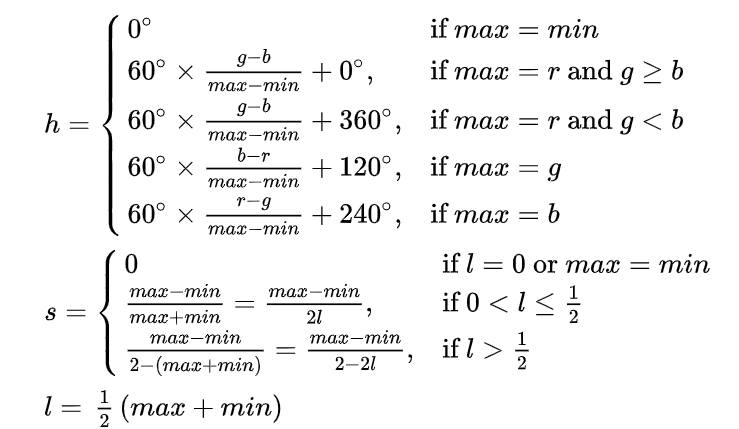
 

图2 HSV示意图

2.3.2实现：

在读取数据时convert.’HSV’,保存图像的HSV三通道数据

2.3.3数据流:

图像HSV数据128\*128\*3->加和1\*128\*128->批输出 batchsize\*1\*128\*128

将Canny，Gabor，HSV拼接输出batch\_size\*6\*128\*128大小的特征图

2.4 Haar

2.4.1原理:

Haar特征分为边缘特征、中心特征和对角线特征。特征模板内有白色和黑色两种矩形，定义该模板的特征值为白色矩形像素和减去黑色矩形像素和,该特征值反映了图像的灰度变化情况。例如：眼睛要比脸颊颜色要深，鼻梁两侧比鼻梁颜色要深。

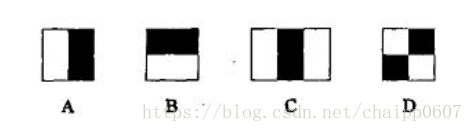
矩形特征对边缘、线段较敏感，下图是描述特定走向（水平、垂直、对角）的常用结构。

图3 Haar特征示意图

2.4.2实现：积分图法/卷积法算法

积分图法: 计算每个点包围的像素和生成积分数组->harr矩形滑动，特征值调用积分数组做加减法->不同位置不同模板遍历后生成长长的特征向量

数据流：特征图batch\_size\*6\*128\*128 -> 特征向量 batch\_size\*27530

卷积法: 设计8种Haar滤波器（尺寸8，16）->卷积->池化上采样（步长为8）

数据流：特征图batch\_size\*6\*128\*128 -> 特征张量 batch\_size\*6\*8\*16\*16

3. 分类器模型

3.1 Classifier类

3.1.1原理：

不断生成随机mask，直到获得一个有一定区分度的弱分类器

3.1.2实现：

类属性：特征掩码、阈值向量、分类类型（> <阈值）、分类器权重

特征张量经过特征掩码变为特征向量->根据阈值分类->计算错误率、分类器权重和下一个分类器的样本权重矩阵

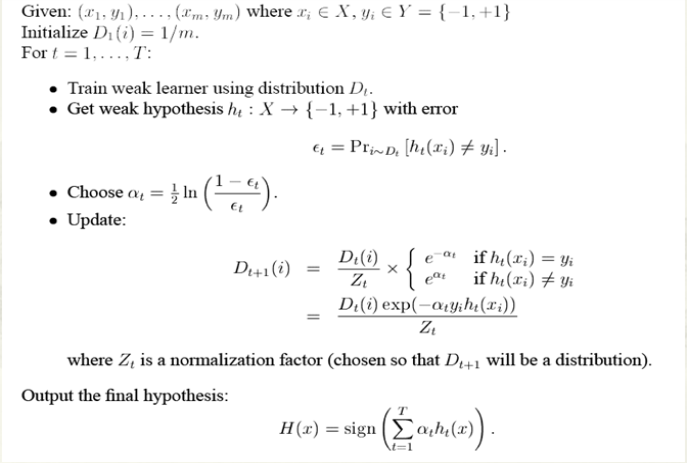
3.1.3数据流：

特征张量batch\_size\*6\*8\*16\*16-> 特征向量batch\_size\*48

3.2 AdaBoost类

3.2.1原理：

前一个基本分类器被错分类的样本的权值会增大，而正确分类的样本的权值会减小，并再次用来训练下一个基本分类器。同时，在每一轮迭代中，加入一个新的弱分类器，直到达到某个预定的足够小的错误率或达到预先指定的最大迭代次数才确定最终的强分类器。



3.2.2实现：

类属性：弱分类列表、弱分类器权重、弱分类器结果、分类结果、错误率

初始化权重矩阵 -> 生成弱分类器，弱分类器属性已包括自身权重和分类结果-> 计算新的权值分布传入下一个弱分类器 -> 按弱分类器权重组合各个弱分类器得到最终结果并计算错误率

Pred函数用于预测test图片，调用弱分类器对象的classify函数

3.2.3训练AdaBoost：

每个AdaBoost迭代10轮，将错误率最低的模型用joblib保存下来

3.3 Detector：

实现端到端的图片分类功能，输入一张图片，自动经过读取AdaBoost模型、图像预处理、将特征张量送入不同的AdaBoost进行预测、所有结果经过softmax输出概率最大的类索引

4.其他模块：

Function里面包含了常调用函数:归一化、卷积、图像显示、高斯平滑、softmax等

Others文件包含了尝试过的其他算法，如SVM、Haar的积分算法、AdaBoost算法

三、算法与改进

1.Dataset考虑点

1.1由于Haar处理两维数据，故将HSV三通道数据相加变为1通道，保留特征相关性

1.2 SingleClass类使用\_\_call\_\_(self, mode)，mode=1返回HSV，mode=2返回gray

1.3初始读入就高斯平滑去噪，全部鸟类数据放在一起做归一化，保留数据集之间的差异

1.4先生成所有图的特征张量，生成样本集时直接抽取，不用重复运算

1.5保证正负样本数近似相等

1.6为避免list和array的切换造成数据类型混乱，Dataset里所有的添加数据均采用concat方式，而非list的append方式

2.Canny

2.1梯度计算：左图为普通的f '(x) = f(x + 1) - f(x)，右图为改为sobel算子求取梯度的边缘图。可见sobel对噪声具有平滑作用。

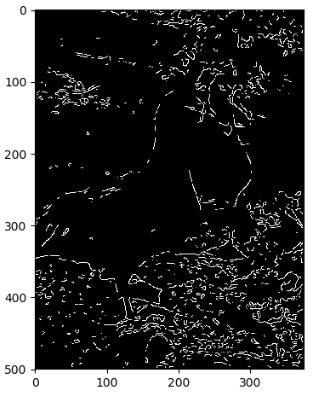
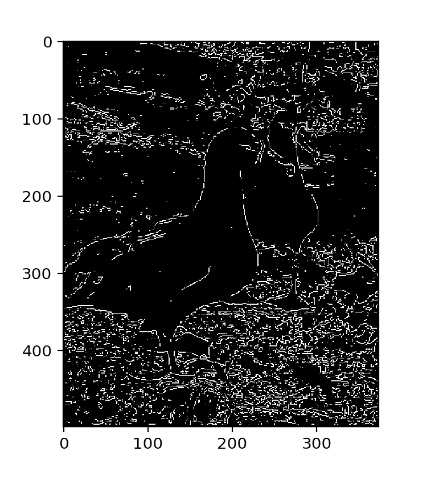


图4 梯度结果对比图

2.2非极大值抑制（NMS）：一开始设定局部梯度最大规则：

1）比周围8个点的梯度要大 2）沿着梯度方向最大

做完实验发现条件1损失边缘过多，故只采用条件2。

根据grad\_x、grad\_y的大小和方向分成4种情况插值法求取梯度方向最大值

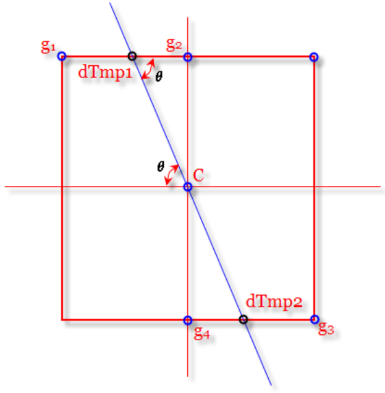


图5 梯度NMS示意图

如图情况是 grad\_y>grad\_x 且 grad\_x\*grad\_y> 0 蓝线代表梯度方向

dTmp1 = weight\*g1 + (1-weight)\*g2 ;

dTmp2 = weight\*g3 + (1-weight)\*g4 ;

这里: weight = abs(grad\_x)/abs(grad\_y)

若梯度值> dtemp1、dtemp2,则保留，否则置0

2.3软性八联通

1.5\*grad： 超过高阈值 + 连通 原算法 1

1.2\*grad： 超过高阈值 + 不连通 1

1.0\*grad： 超过低阈值 + 联通 1

0.0： 超过低阈值 + 不连通 0

0.0： 未超过阈值 0

增强强边缘点的同时保留了图像梯度特征。左图为原算法效果，右图为改进后的灰度图。

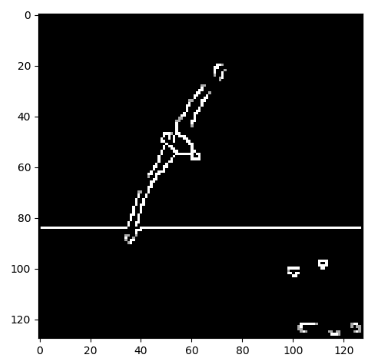
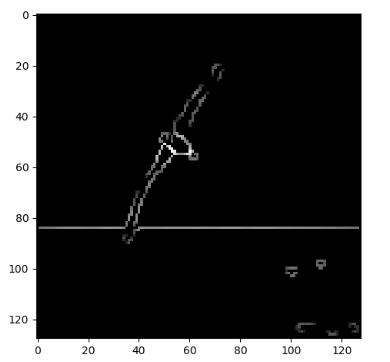
 

图6 八联通结果对比图

3.Gabor

生成4种尺寸、方向不同的Gabor滤波器与灰度图卷积

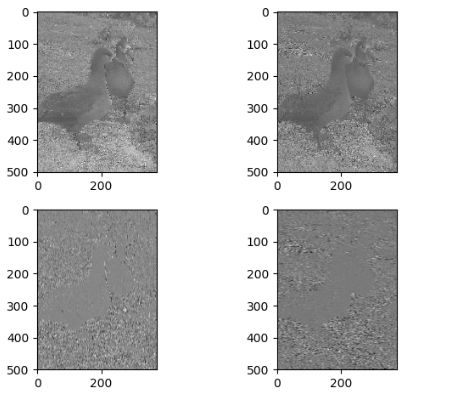


图7 Gabor结果示意图

4.Haar

实现采用两种算法。

4.1积分图法:

计算每个点包围的像素和生成积分数组->harr矩形滑动，特征值调用积分数组做加减法->不同位置不同模板遍历后生成长长的特征向量

数据流：特征图batch\_size\*6\*128\*128 -> 特征向量 batch\_size\*27530

4.2卷积法:

设计8种Haar滤波器（常规矩形尺寸8，16）->卷积->池化上采样（步长8）

数据流：特征图batch\_size\*6\*128\*128 -> 特征张量 batch\_size\*6\*8\*16\*16

比较：积分图法用于实时检测，速度快；卷积算法简洁直观，方便最大池化采样。由于本任务不追求快速，且希望为后续选取有效特征做准备。最终决定采用卷积法实现。下图是最大池化后的效果图。

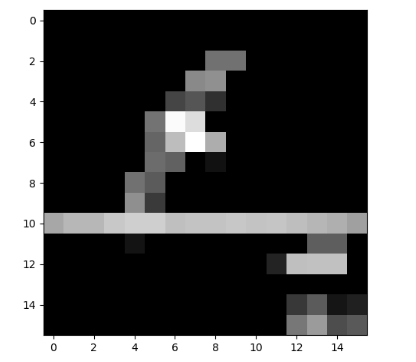


图8 Haar结果图

5. Classifier

5.1张量变向量：

将16\*16的特征张量与随机生成矩阵相乘的和作为这张特征图的特征值

曾尝试过

1）在特征图上随机选取n个点加和（n=16）

2）将特征张量reshape成（48\*256）pca降维到48\*16再随机选点（意义不大且耗时） 3）生成random.rand16\*16矩阵与特征张量点乘加和（由于代码实现方便选取）

5.2阈值和分类类型选取:

正负样本的特征向量均值作为阈值，>、<阈值分别计算错误率，将较低的错误率作为弱分类器的错误率、比较方式作为分类类型

5.3更新样本权重矩阵：

放在弱分类器里实现是为了避免在AdaBoost里再次调用弱分类器的分类结果，显得代码冗余。采用了normal\_adaboost和fast\_adaboost两种算法：

normal\_adaboost:如AdaBoost原理图所示，增大分错样本的权重，将样本权重与检测矩阵（分错为1，分对为0）点积的值作为该分类器的错误率,进而影响它在AdaBoost里的权重

fast\_adaboost：采用了0，1矩阵的模式，上一个弱分类器分对的样本权重直接为0，下一个分类器只学习分错误的样本，训练速度很快。由于本任务时间紧迫、不追求正确率且弱分类无需训练，故采用这种方式构造强二分类器

6.AdaBoost

pred函数调用Classifier的classify，此时就看出将特征掩码、阈值向量、分类类型作为分类器属性的好处。在训练时，为加快训练速度，将所有样本集的特征张量和标签保存到参数文档，直接调用。

四、Torch-convolution 与机器学习算法对比

1.深度学习要求数据量大，像这种小数据集很容易过拟合。

2.本任务的预处理和深度学习的预处理一样，所以为了以后的使用这部分画的心思很多。其中canny的梯度计算、gabor、haar均采用了卷积的方法提取特征，另外，我还借鉴了卷积层的一些操作，如 canny的NMS、软性八联通类似激活函数的作用，haar的最大池化等。

3.在弱分类器那一块，特征提取对后来的正确率影响很大，所以以往的机器学习都需要专家来设计特征。而深度学习自己学习特征，但正确率相对更高。本任务分类之前的操作几乎全部是线性变化，其实可以合成一步实现。而深度学习有很多非线性变换，其拟合特征的能力更强。

4.一开始弱分类器我采用的是线性分类的方法，它和全连接层相比少了激活函数，但由于随机梯度下降法违反题目要求放弃；后尝试SVM的简易SMO算法，其清晰的数学推理和深度黑箱学习大不相同。