

华中科技大学

模式识别课程设计报告

题号 (13 号)

题	目	材质识别
专	业	自动化
学	生	李义, 单文俊, 杜宇之, 武家声
学	号	U201514158,U201514405
学	号	U201514234,U201514399
日	期	2019-1-13

自动化学院

目录

第 1 章 背景与现状	4
第 2 章 题目概述	4
第 3 章 课题难点	5
第 4 章 任务基本要求	5
第 5 章 需求分析	5
第 6 章 研究方案设计	6
6.1 对数据的观察	6
6.2 算法设计	6
6.2.1 特征提取-LBP 编码	6
6.2.2 分类器选择-SVM	8
6.2.3 算法分析	9
第 7 章 算法实现	9
7.1 算法流程图	9
7.2 算法实现方案	10
7.3 程序功能介绍	10
7.4 结果	11
第 8 章 改进	12
8.1 破坏类别平衡	12
8.2 扩充数据集	12
8.3 添加 Gaussian noise	14
8.3.1 效果展示	14
8.4 算法的改进	15
8.4.1 特征提取与数据预处理	15
8.4.2 使用其他的学习算法	15
8.5 可以改进的地方	18
8.6 Bayesian computational framework	18
第 9 章 总结	20
9.1 课设结果总结	20
9.2 个人心得体会	20

模式识别课程设计报告

摘要

本次课程设计, 是完成对纹理的识别分类任务, 在给定的 6 种图片数据集下 (81*200*200) 进行多分类. 要求给出分类正确率以及标准差. 实验中, 采用 LBP+SVM 的基本方法完成了分类任务, 结果正确率为 $98.7\% \pm 0.08\%$ 达到的参考文献中的结果. 然后破坏类别平衡条件, 扩充 CUREt 数据集, 通过调整噪声方差的方法添加噪声, 观测训练模型的结果. 在破坏类别平衡条件下, 只需扩充数据集即可完成, 在扩充数据集为 CUREt 数据集下, 结果基本和原来的保持一致, 但是我们发现噪声对实验结果由很大影响, 尝试使用 Uniform LBP 编码等一系列数据预处理, 特征提取方法, 发现效果不明显, 因此我们采用其他的学习算法例如, KNN 分类, PCA+KNN, Bayesian method: LDA topic model(未完成), 以及 Deep filter bank(未完成) 对原先的方法进行改进.

摘要

In this paper, we use state of art texture recognition method: LBP+SVM to classify 6 classes image data. The classification accuracy on the given dataset is : $98.7\% \pm 0.08\%$. Then we break the class-balance condition, add some Gaussian noise on the datasets, use some other datasets i.e. CUREt to implement the same method. The main problem we found is that when we added some Gaussian noise, the accuracy of classification drop rapidly. Hence we try to implement some other feature extraction techniques such as uniform LBP, SIFT, HOG and other machine learning methods such as KNN, PCA+KNN, Bayesian framework classifier: LDA topic model(todo), deep filter bank to overcome these problems(todo).

第 1 章 背景与现状

材质识别是视觉识别的一个重要方面。我们每天与各种材料互动，并不断评估它们的外观。例如，在结冰的人行道上确定安全的行走路线，或在农贸市场购买新鲜农产品，或决定皮疹是否需要去看医生时，物质的质量会影响我们的决定。因此，建立一个能从图像中推断材料性质的视觉识别系统是非常有价值的。我们感兴趣的是从一个单一的图像表面识别材料类别，如玻璃、金属、织物、塑料或者木材等。从材料的视觉纹理中识别材料有许多应用，例如，它有助于图像检索和目标识别。对于这门学科，最近的发展集中在能够识别来自各种姿势和不同的照明条件的材料上。从照片中识别材料的问题主要是在反射率估算的背景下解决的。表面的视觉外观取决于几个因素——照明条件、表面样品在几个空间尺度上的几何结构和表面反射特性，通常以双向反射分布函数及其变体为特征。在光照、几何和材料特性的限制性假设下，已经开发出许多技术，可以从一组照片中估算 BRDF 模型的参数。为了识别材料，已经开发了许多低阶图像特征。人们发现图像亮度直方图的形状与人类对表面反照率的判断有关，并将其用于对光泽度、哑光、白色、灰色等球体进行分类。

第 2 章 题目概述

给出如下 6 种材质的数据

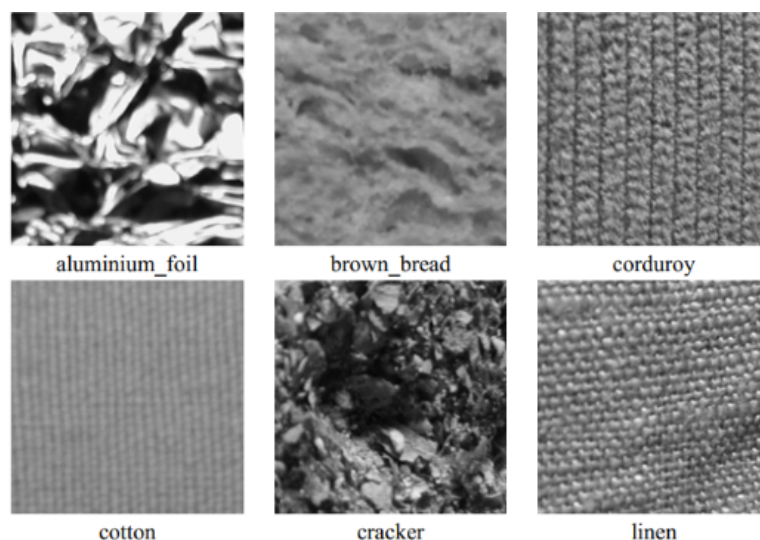


图 2.1:

需要设计算法, 对这 6 种材质进行分类. 输出准确率与标准差.

第 3 章 课题难点

根据外观将材料分类是一个很有挑战性的问题，尤其是在允许光照和姿态条件改变的情况下：由 3D 结构引起的高光和阴影可以从根本上改变样本的视觉纹理。与计算机视觉中的其他视觉识别任务不同，由于材料可以显示的外观差异很大，很难找到能够区分材料类别的良好、可靠的特征。通过阅读 [2] [4] [1], [5], [6] 中的方法，了解当前纹理识别的一些常用方法，然后选取合适的方法进行实现。

第 4 章 任务基本要求

使用: E. Hayman, B. Caputo, M. Fritz, J.-O. Eklundh, “On the significance of real- world conditions for material classification” , in Proc. European Conference on Computer Vision (ECCV), 2004, pp. 253–266.

中的算法, 完成相应的任务.

第 5 章 需求分析

本次课设需要对 6 种材质的图片进行识别与分类，每种材质的样本包中均有 81 张图，选取 40 张进行训练，剩余 40 张进行测试并得到分类的准确率。首先，直观上看，对于这种数量的样本容量并不大，因此可以不用神经网络进行训练与分类；需要分出的种类也不是很多，所以用 SVM 就可以，并且 python 的 sklearn 中已经有了完整的使用 SVM 训练与分类的功能，所以这一部分可以直接调用已有函数。而根据已给论文可知，对于图片进行 lbp 编码提取特征向量具有旋转不变性和灰度不变性等显著优点，而且提取的特征是图像的局部纹理特征，因此，对图像的处理可以使用 lbp 提取特征向量。

第 6 章 研究方案设计

本次课设的分为两部分，一是图像处理（去噪、提取特征向量等），二是训练与分类。前者我们的重点是通过 lbp 提取特征向量，后者我们通过 SVM 进行训练与分类。补充部分尝试使用不同的特征提取与预处理方法，不同的学习模型对原问题进行解决。

6.1 对数据的观察

一个视觉任务的关键，是数据集。数据集好，效果就会好。这里我们先观察一下数据集是怎么样。所给数据集过于好了：从图像质量上，光照均与，尺度相近…也就是说我们设计的分类器是非常容易骗的，在实际应用中一定会过拟合。

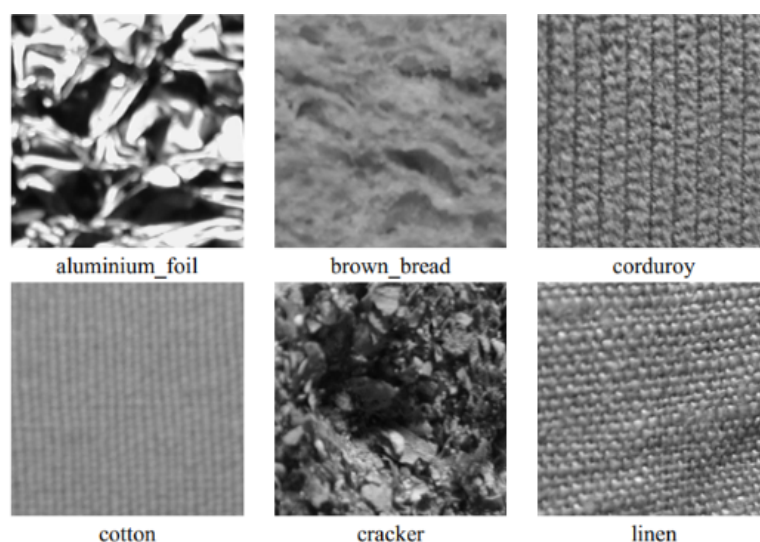


图 6.1: 六类材质数据

6.2 算法设计

使用 [2] 中的方法，即 LBP+SVM。

6.2.1 特征提取-LBP 编码

LBP 简介:LBP (Local Binary Pattern, 局部二值模式) 是一种用来描述图像局部纹理特征的算子；它具有旋转不变性和灰度不变性等显著的优点。它是首先由 T. Ojala, M.Pietikäinen, 和 D. Harwood 在 1994 年提出，用于纹理特征提取。而且，提取的特征是图像的局部的纹理特征

原始的 LBP 算子定义为在 3×3 的窗口内，以窗口中心像素为阈值，将相邻的 8 个像素的灰度值与其进行比较，若周围像素值大于中心像素值，则该像素点的位置被标记为 1，否则为 0。这样， 3×3 邻域内的 8 个点经比较可产生 8 位二进制数（通常转换为十进制数即 LBP 码，共 256 种），即得到该窗口中心像素点的 LBP 值，并用这个值来反映该区域的纹理信息。如下图所示：

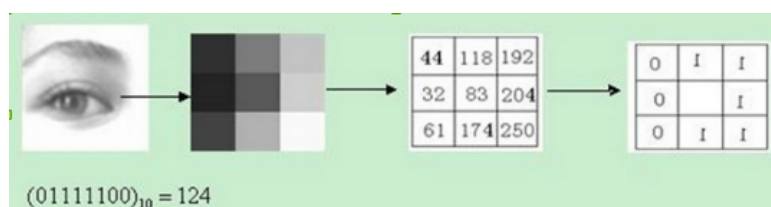


图 6.2: 图片来源: 网络

LBP 的改进: Uniform LBP: 一个 LBP 算子可以产生不同的二进制模式，对于半径为 R 的圆形区域内含有 P 个采样点的 LBP 算子将会产生 2^P 种模式。很显然，随着邻域集内采样点数的增加，二进制模式的种类是急剧增加的。例如： 5×5 邻域内 20 个采样点，有 $220 = 1,048,576$ 种二进制模式。如此多的二值模式无论对于纹理的提取还是对于纹理的识别、分类及信息的存取都是不利的。同时，过多的模式种类对于纹理的表达是不利的。例如，将 LBP 算子用于纹理分类或人脸识别时，常采用 LBP 模式的统计直方图来表达图像的信息，而较多的模式种类将使得数据量过大，且直方图过于稀疏。因此，需要对原始的 LBP 模式进行降维，使得数据量减少的情况下能最好的代表图像的信息。

为了解决二进制模式过多的问题，提高统计性，Ojala 提出了采用一种“等价模式”(Uniform Pattern) 来对 LBP 算子的模式种类进行降维。Ojala 等认为，在实际图像中，绝大多数 LBP 模式最多只包含两次从 1 到 0 或从 0 到 1 的跳变。因此，Ojala 将“等价模式”定义为：当某个 LBP 所对应的循环二进制数从 0 到 1 或从 1 到 0 最多有两次跳变时，该 LBP 所对应的二进制就称为一个等价模式类。如 00000000 (0 次跳变)，00000111 (只含一次从 0 到 1 的跳变)，10001111 (先由 1 跳到 0，再由 0 跳到 1，共两次跳变) 都是等价模式类。除等价模式类以外的模式都归为另一类，称为混合模式类，例如 10010111 (共四次跳变) (这是我的个人理解，不知道对不对)。

通过这样的改进，二进制模式的种类大大减少，而不会丢失任何信息。模式数量由原来的 2^P 种减少为 $P(P-1)+2$ 种，其中 P 表示邻域集内的采样点数。对于 3×3 邻域内 8 个采样点来说，二进制模式由原始的 256 种减少为 58 种，这使得特征向量的维数更少，并且可以减少高频噪声带来的影响。

显而易见的是，上述提取的 LBP 算子在每个像素点都可以得到一个 LBP “编码”，那么，对一幅图像（记录的是每个像素点的灰度值）提取其原始的 LBP 算子之后，得到的原始 LBP 特征依然是“一幅图片”（记录的是每个像素点的 LBP 值）。例如：

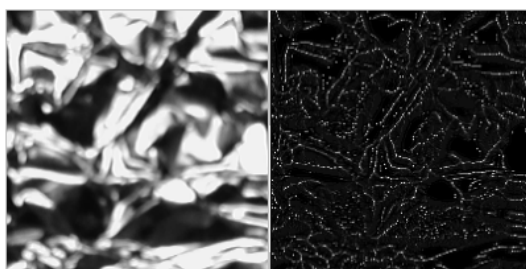


图 6.3: 左边是原图, 右边是 *LBP* 特征

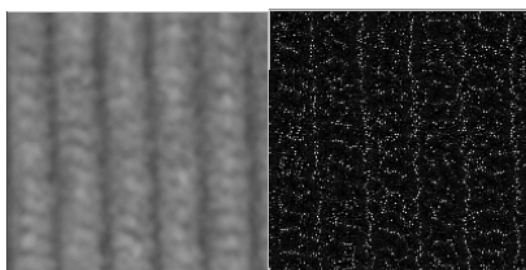


图 6.4: 左边是原图, 右边是 *LBP* 特征

此外, LBP 的应用中, 如纹理分类、人脸分析等, 一般都不将 LBP 图谱作为特征向量用于分类识别, 而是采用 LBP 特征谱的统计直方图作为特征向量用于分类识别。

6.2.2 分类器选择-SVM

SVM 简介: 大家应该对 SVM 已经比较熟悉了, 这里对其做一个简要的介绍. SVM 原始情况下是一个线性分类器, 这样依赖, 模型的容量就不会很大的 [3], 因此很多时候是欠拟合的, 我们在这里使用的 SVM 是使用核方法的, 核方法的好处是可以使 SVM 做不是线性分类器可以完成的工作. 通常使用 rbf 核.

sklearn 接口:

使用 sklearn 中的包来完成相应的训练任务, 具体的接口如下:

```
1 SVC(C=1.0,kernel='rbf',degree=3,gamma='auto_deprecated',coef0=0.0,shrink_
  ↳ king=True,probability=False,tol=0.001, cache_size=200,
  ↳ class_weight=None,
  ↳ verbose=False,max_iter=-1,decision_function_shape='ovr',
  ↳ random_state=None)
```

对数据的标准化: 采用 $x' = \frac{x - \min}{\max - \min}$ 对数据进行标准化.

多分类方式:OvO, 即两两进行二分类.OvO 的方法如下图所示 [5]

分类算法参数的确定:SVC 参数: $C = 32$, rbf kernel $\gamma = 0.5$

因为只有两个参数, 采用 uniform grid method 优化参数 (交叉验证方法优化参数).

6.2.3 算法分析

feature extraction complexity: $\mathcal{O}(9N^2KC)$

这是因为, 每次卷积需要计算 9 个位置的信息, 而每一类有 $K = 81$ 张图片, $C = 6$ 为一共 6 个类别, N 为图像的长和宽, 因此最终计算量级大概在 $2e9$.

Training complexity : $\mathcal{O}(dN^3)$

Prediction complexity :for rbf kernel $\mathcal{O}(d^2)$ - $1e4$ 的维度一下差不多都是毫秒级询问.

因此交叉验证一次的复杂度为: $\mathcal{O}\left(kd\left(\frac{k-1}{k}N\right)^3\right)$ - $1e9$ 量级

其中 k 为进行交叉验证的次数, d 为数据的维数, N 为样本点数. 因此计算一对参数点下的 loss 大概在 $1e9$ 量级. 也就是说按分钟来进行计时的. 如果将 grid search 看做常数, 那么就是几秒-几百秒的样子.

参考链接: <https://stats.stackexchange.com/questions/88975/impact-of-data-dimensionality-on-computational-complexity-of-prediction>
<https://stats.stackexchange.com/questions/94596/computational-complexity-of-prediction>

第 7 章 算法实现

7.1 算法流程图

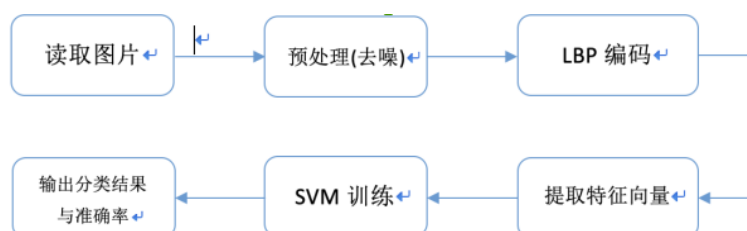


图 7.1: 首先数据预处理, 然后特征提取, 最后训练模型, 分类

7.2 算法实现方案

采用常用的方法: 特征提取使用 C++, 模型训练使用 python.

这是因为 C++ 的运行效率在特征提取上略高.

主要分为三的部分: 读取数据, 特征提取, 模型训练与结果输出

选用 C++ 完成特征提取是常用的方法, 这是因为 C++ 的效率会比 python 高一些.

7.3 程序功能介绍

feature extraction.cpp

```
1 //LBP编码,tmp为图像数据
2 int LBP_encode(int tmp[][maxn],int i,int j,int N,int M)
3
4 //数据归一化
5 inline void trans_feature(double *feature,int m,int M)
6
7 //添加Gaussian noise
8 void Guassain_Smooth(IplImage *inm,IplImage *outm,double mean,double
   ↪ stddev,int cnt)
9
10 //平滑滤波
11 void imfilter(IplImage *inm,IplImage *outm,int sz)
12
13 //完成特征提取
14 void feature_extraction(string path)
```

SVM training.py

```
1 #从path 读取数据
2 def read_data(path)
3
4 #x为数据,y为标签,cas记录是第几组
5 def svm_solve(x,y,cas):
6
7 #从r读数据,写到wpath中
```

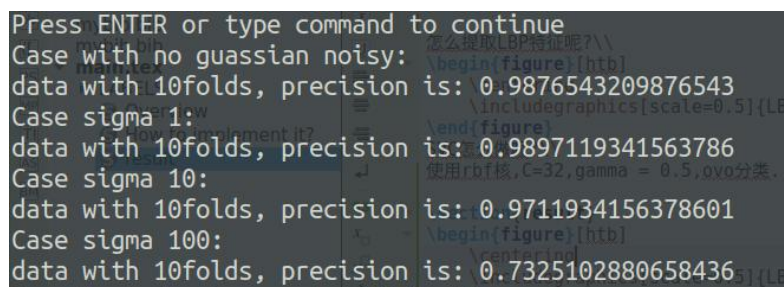
```
8 def get_data(rpath,wpath):
9
10 #写特征,标签到文档
11 def write_data(feature,label,path):
12
13 #提取图像特征
14 def get_feature(pic):
15
16 #进行LBP编码
17 def encode(lbp):
```

PCA KNN.cpp

```
1 //定义数据结构-kd tree
2 class Kdtree
3
4 //建树方法,l,r是当前递归位置标记,rt是当前节点标号,dep表示当前递归深度
5 void build(int l,int r,int rt,int dep)
6
7 //rt为当前节点标号,dep为当前深度,q是待询问点,M表示询问M近邻
8 void query(int rt,int dep,Point q,int M)
9
10 //PCA将LBP特征降维,maxd表示降维后的维度.
11 void PCA(int feature[][maxc],int reduc[][maxd])
```

7.4 结果

结果为 $98.7\% \pm 0.08\%$ 如下图所示:



```
Press ENTER or type command to continue
Case with no gaussian noisy:
data with 10folds, precision is: 0.9876543209876543
Case sigma 1:
data with 10folds, precision is: 0.9897119341563786
Case sigma 10:
data with 10folds, precision is: 0.9711934156378601
Case sigma 100:
data with 10folds, precision is: 0.7325102880658436
```

图 7.2:

因此达到了基本的要求, 接下来我们做一些改进

第 8 章 改进

我们对原实验进行了三个扩展:

- 破坏类别平衡条件
- 扩充数据集: 具体的我们扩充数据集为 CURET 数据集, 大约有 5600 张 200×200 的图片
- 添加噪声: 我们使用了一种十分粗糙的添加噪声的方法, 即在原图像上加上加性 Gaussian noise, 参数 (μ, σ) 其中 $\mu = 0, \sigma = 0, 20, 40, 60, 80, 100$.

8.1 破坏类别平衡

这种情况简单的对数据进行 augment 即可, 对实验结果没有什么影响

8.2 扩充数据集

我们扩充数据集使用经典的纹理识别数据集:CURET 数据集¹.
数据如下图所示:

¹<http://www.cs.columbia.edu/CAVE/software/curet/>

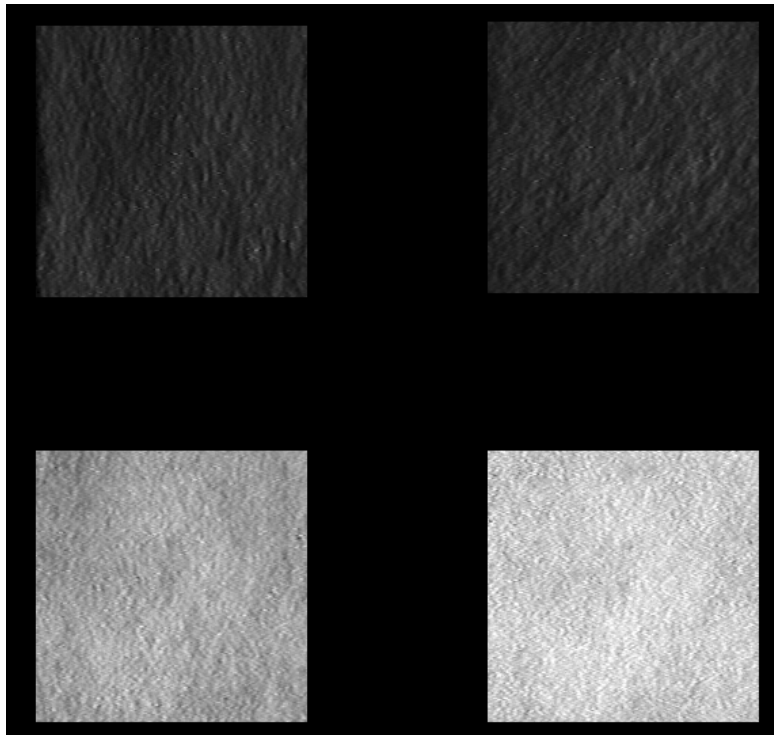


图 8.1: 注意到同一类别的数据有明显的光照差异, 但是发现对最终的分类结果没有太大的影响, 这是因为 *LBP* 特征是比较的相对关系, 跟整张图片的光照强弱是没有太大的关系的.

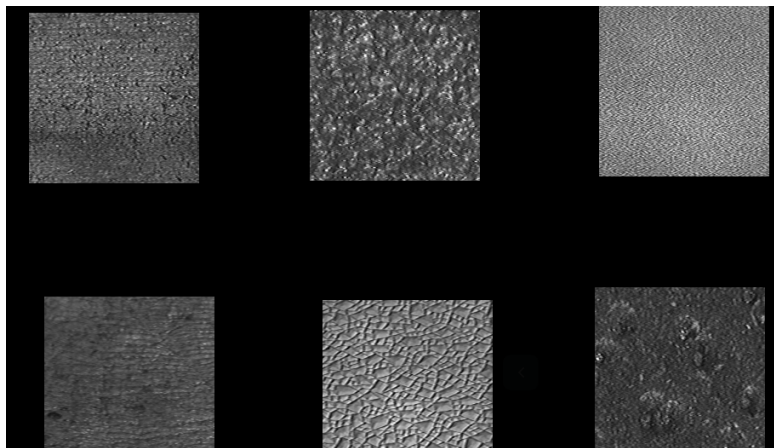


图 8.2: 可以发现不同的纹理之间有比较明显

我们仍然使用先前的方法, 发现准确率在 97.9% 也就是说准确率并没有明显的降低, 这说明

8.3 添加 Gaussian noise

8.3.1 效果展示

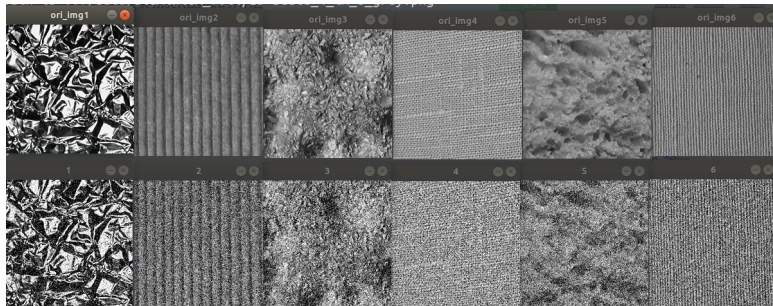


图 8.3: 其中上面一排的图片是原始的 6 类图片, 下面一排的图片是加了 $\sigma = 50$ 时候的图片, 可以发现, 纹理还是可以比较清晰的分辨出来的, 同样, 使用 $LBP+SVM$ 分类没有太大的准确率的降低.

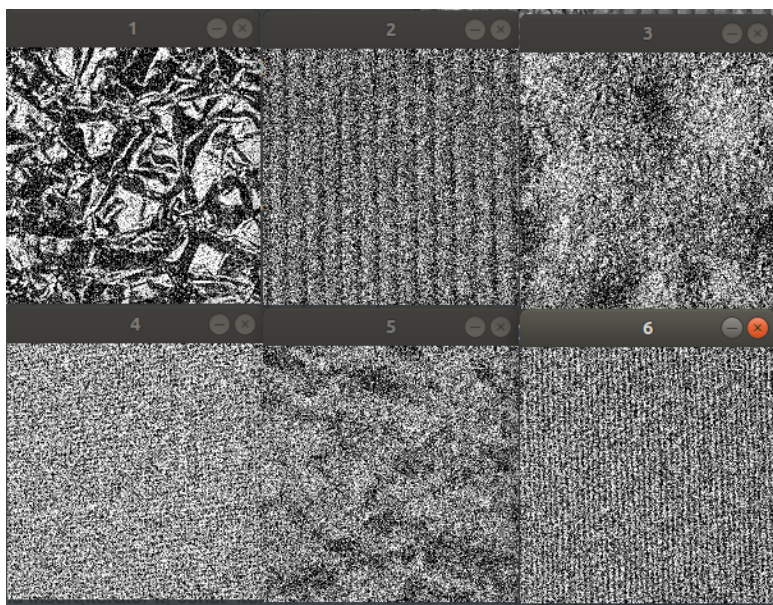


图 8.4: 可以发现, 和上面的图像相比, 下面一排的第一张和第三张已经比较难以区分了

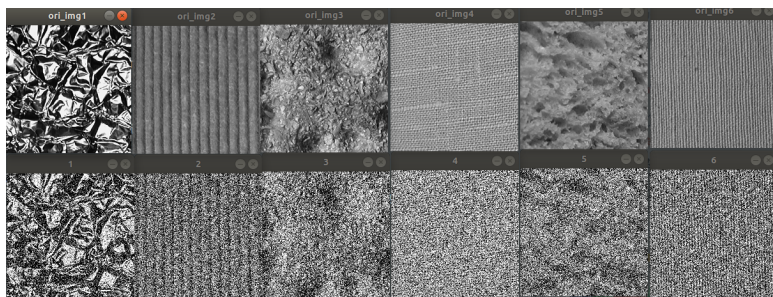


图 8.5: 上面一排为原始图像, 下面一排为加了噪声时候的图像, 可以发现下面一排的第 4 张和第 6 张图像已经淹没在了噪声中. 但是放大了看, 可以发现, 一个是竖条纹, 一个是横条纹

我们发现, 和视觉观看的效果类似, 当 $\sigma = 80\%$ 的时候, 精度已经降到 80% 一下, 这说明原先的 SVM+LBP 算法是存在问题的. 因此需要进行一些改进

8.4 算法的改进

8.4.1 特征提取与数据预处理

HOG 特征: 使用 HOG 特征准确率不到 60%

使用平滑滤波: 我们取滤波的半径为 $r = 3, 5, 7$ 发现分类的精度并没有太大的变化, 甚至当 $r = 7$ 的时候分类精度会有所下降, 这是因为较大半径的平滑滤波使得图像产生的失真反而不容易进行分类.

使用 Uniform LBP 算子: 发现和原始的 LBP 算子一样, 精度几乎没有什么变化, 因此也不能解决问题

在经过了上面的一些数据预处理以及特征提取方法的改进之后, 我们意识到, LBP 可能是最适合做纹理识别的一种算子, 他几乎提取的图像纹理的绝大部分信息, 因此, 在使用 LBP 算子的条件下, 应该尝试采用其他的学习分类策略来改进当前结果.

8.4.2 使用其他的学习算法

首先注意到, 这个图像的数据非常的好, 光照明显, 无太大的尺度变化, 同一类别的图像基本是一样的, 这提示我们, 可能需要更弱的学习器去完成这个在噪声条件下的分类任务, 这是因为, 图像数据集非常的弱, 用稍微强一点的学习器可能对噪声, 也就是变化会变得敏感, 也就是所谓的过拟合, 因此尝试使用弱学习器是解决问题的一种方法. 我们尝试一下 KNN.

KNN 在实际的应用场合中也是经常见到的. 这是因为他的查询效率非常的高他的复杂度在十分理想的情况下是 $\mathcal{O}(N \log N)$ 的, 我们接下来做一个简单的介绍:

例如我们现在有 $(3, 6)$, $(17, 15)$, $(13, 15)$, $(6, 12)$, $(9, 1)$, $(2, 7)$, $(10, 19)$ 这么多个点, 最终形成了一颗 BST 如下:

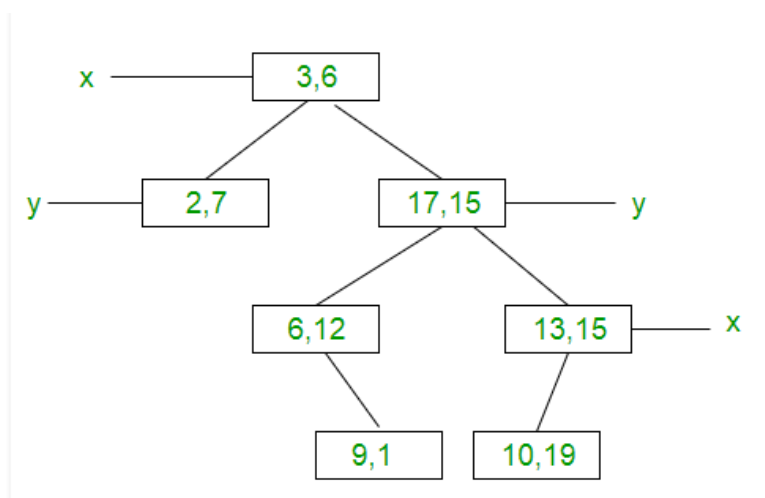


图 8.6: 这是 $K=2$ 的情形

最后静态 KD 树对平面划分的情况如下图所示:

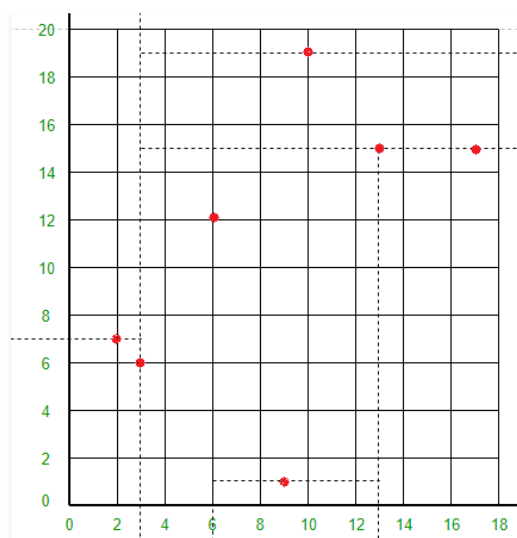


图 8.7: 注意到, 这个划分是与上面的 BST 对应的, 其中根节点划分 x , 接下来一层划分 y 再接下来一层划分 x , 如此反复, 然后每次都是按照堆的 *insert* 方法进行节点的插入来建树

BST 有一个有点就是他的深度是比较短的, 因此只需要 $\mathcal{O}(\log N)$ 的询问就可以找到一个点.

接下来是询问 K 邻近, 询问 K 邻近的方法是使用一个优先队列维护最近的节点, 我们在这颗 BST 上搜索的方法是先进入最优可能离当前点最近的点, 如果我们走到底仍然没有找到这个点, 就沿着原路返回, 因此最坏情况的时间都浪费在了返回上面. 返回的方法如下图所示:

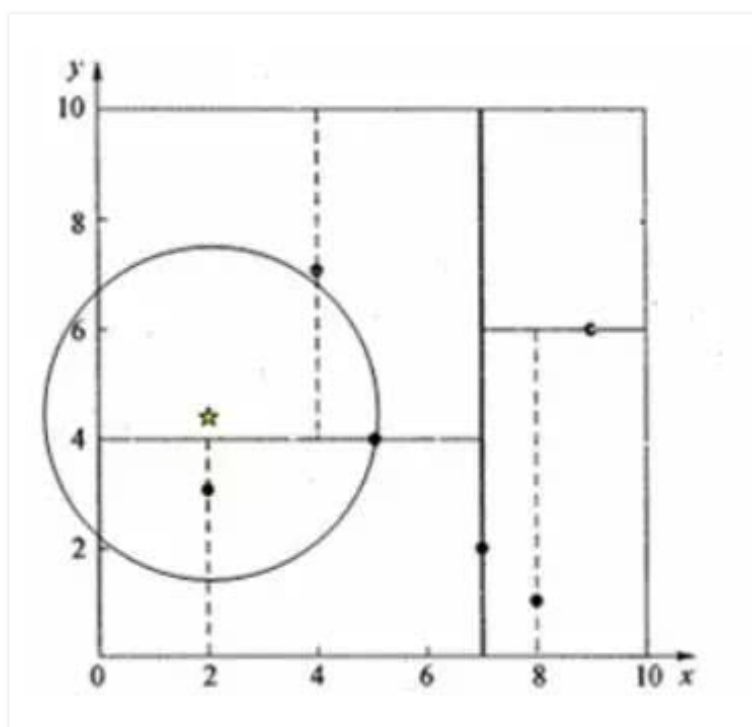


图 8.8: 这里标星的是当前的所在的位置, 然后递归的时候, 检查发现所做的圆与上一层的划分有交, 那么就需要 *trace back* 到他父亲的另一个孩子, 否则返回当前点为最邻近点, 图片来源: 网络

对于这个问题, 当我们使用 KNN 的时候, 得到的准确率为 82%, 这个结果也不太令人惊讶, 因为所给的数据集中同一类别的图片几乎是一样的, 因此使用容量不大的模型, 也完成了任务. 但是当我们加上噪声的时候, 精确度下降到了 78%, 下降的幅度是远远低于 SVM 的, 为什么会出现这个结果, 直观上, KNN 可以看成是一个比较迟钝的人, 那么他对外界的变换就不如一个比较敏感的人那个大. 因此加上了噪声, 结果仍然不错.

接下来我们考虑, 当前是在 256-dim 进行 KNN, 如果我们通过降维, 将其在其他的空间中度量, 就有可能提高精确度, 因此我们就使用了 PCA, 我们知道 PCA 对噪声是比较鲁棒的, 于此同时, 通过 PCA 降维可以去掉一部分与特征无关的信息, 这样可以使得 KNN 的分类精度有所提高, 实验结果发现精度稍微提高到了 85%, 加上噪声的时候, 精度在 82%, 也就是说, 使用 PCA+KNN 使得在噪声情况下的分类精度比同样在噪声情况下的 LBP+SVM 提高了大概 8% 左右, 在没有噪声的情况下, LBP+SVM 仍然远远比先前的效果好.

与 SVM 进行比较:

因此在理想情况下, 如 SVM 的 $\mathcal{O}(dN^3)$ 训练复杂度相比少了不少, 另一方面, 如果单次询问的情况下却不是这个, SVM 的预测复杂度是十分小的 $\mathcal{O}(d^2)$, 对于这个问题, 256-dim, 表示说计算量级是 $1e4$ 左右的, 这意味着, 在 ms 时间内就可以完成询问, 但是 KNN

需要 $\log N$ 的时间, 两者实际是相当的.

总而言之,KNN 效率比 SVM 高, 尤其是在有噪声对效率要求比较高的场合使用 PCA+KNN 是一个比较不错的选择.

8.5 可以改进的地方

- 首先添加噪声的方法, 可以换成信噪比的方式添加
- 添加噪声可以使用 Goodfellow 等人在 2016 年提供的一些 Adversial example 构造方法添加来看看效果是怎么样的
- 还可以绘制出决策边界, 看看为什么添加噪声之后的实验结果会下降这么明显, 一种直观的认识是认为过拟合了, 因为使用比 SVM 弱的学习器效果不错.
- 可以实现特征集成的方法来抵抗噪声, 具体的是将多特征送入 Bayes 学习器, 决定哪个特征在这个情况下占比应该更多一些. 也就是实现 [4] 中的方法.

8.6 Bayesian computational framework

概述计算框架: 也就是 [4] 中的方法, 我们对其思想做一个简述:

如果我们有 M 种不同的特征, 每个特征我们都形成一个字典, 字典中的词通过 K-Means 获得, 例如这里的 LBP 特征, 可以通过 Kmeans 获得 V_i 个聚类, 每个 cluster 对应一个词. 然后通过如下公式, 将不同特征的字典合成一个字典:

$$D = \left\{ w_1^{(1)}, \dots, w_{N_1}^{(1)}, w_1^{(2)}, \dots, w_{N_m}^{(m)} + V_1, \dots, w_{N_m}^{(m)} + \sum_{i=1}^{m-1} V_i \right\}$$

以上就是 Bags of words 中常用的方法.

然后使用如下的 LDA 模型:

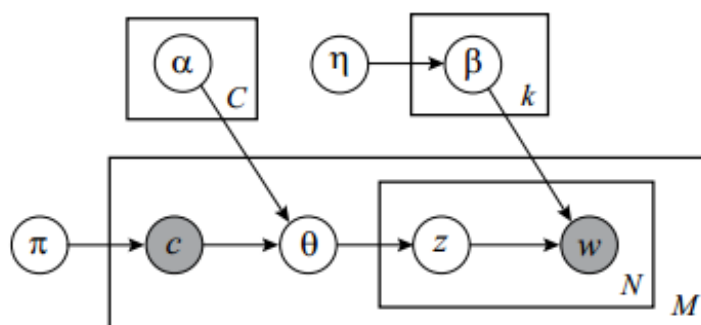


图 8.9:

在原始的 LDA 模型中, θ , 即 topic proportion 是由一个单一的 α 生成的, 这里我们对每一个类别都有一个对应的 α_c , 然后我们首先通过 multinomial distribution 生成类别 c ,

然后通过这个类别选择相应的 α_c , 生成 topic proportion. 由 θ 生成 topic assignment(使用 Multinomial distribution), 然后再由其和 topics 共同生成 word(也就是 BOF 意义中的 word).

然后进行数值优化, 自然的思想是用最大似然估计方法估计当前类别 c , 这里由于数值优化问题使用 $\mathcal{L}(\alpha_c, \eta)$. 然后为了防止过拟合, 需要加上 prior bias. 即 λ_c . 即

$$c^* = \operatorname{argmax}_c (L(\alpha_c, \eta) + \lambda_c)$$

也就是统计学习的经典方法. 这样就可以估计类别了.

至于为什么使用 LDA 实现可行, 暂时还没有弄透彻.

第 9 章 总结

9.1 课设结果总结

这次课程设计, 我们首将 500 多张 200*200 的图片进行 LBP 特征提取然后通过 SVM 进行 OvO 的多分类, 达到了 98.7% 的分类精度, 进一步, 破坏类别平衡条件, 以及扩充数据集对实验结果基本没影响, 但是在添加噪声的情况下, 发现分类精度有较大的下降, 当 $\sigma = 80$ 的时候, 接近下降了 20% 的百分点, 于是们尝试采用其他的特征提取方法以及数据预处理方法, 例如: 平滑滤波, 学习算法, 例如 KNN, 降维 +KNN,LDA,Deep filter bank 等方法进行改进, 但是由于时间关系, 只是从 KNN 中看到可以改进的地方

9.2 个人心得体会

李义: 这次课程设计, 我们首将 500 多张 200*200 的图片进行 LBP 特征提取然后通过 SVM 进行 OvO 的多分类, 达到了 98.7% 的分类精度, 进一步, 破坏类别平衡条件, 以及扩充数据集对实验结果基本没影响, 但是在添加噪声的情况下, 发现分类精度有较大的下降, 当 $\sigma = 80$ 的时候, 接近下降了 20% 的百分点, 于是们尝试采用其他的学习算法, 例如 KNN, 降维 +KNN,LDA,Deep filter bank 等方法进行改进, 但是由于时间关系, 只是从 KNN 中看到可以改进的地方 **单文俊:** 考研结束后, 我便开始参与课设

的完善中。先是稍微看了组长发的几篇论文, 大概了解了材质识别中的重点难点, 以及所需要运用的主要算法。从任务介绍中提到的参考文献, 我了解到材质识别可以通过支持向量机 SVM 来完成, 同时与室友讨论得知还可以提取 LBP 特征, 对于这种材质识别很有用。

自己只是在考研完成后, 大概看了下队友推荐的几篇文章, 主要还是通过队友的已经写好的一部分代码, 自己先百度了解了一些之前在课堂上没学到的知识, 看了一些博客, 更好的理解了 LBP 特征。同时对 python 有了初步了解, 如何表示数组以及相关处理函数, 感觉和 C++ 语言还是有比较大的差别的, 了解了一些基础模块 numpy, pandas, sys, 以及如何使用 sklearn 模块。这些也都为我之后进行毕业设计打下了一点基础。杜

宇之: 由于本人也是漫漫考研大军中的一员, 故任务布置下来之后, 我绝大多数时间都用在了考研准备上。直到考研结束后, 才有空闲用来考虑课程设计的事。所以很惭愧, 本人并未参与到课设的准备、设计阶段的工作中, 只是在任务完成后学习了组长完成项目所用的算法与代码。

模式识别课程设计报告

这次模式识别课设我参与不多，阅读论文与学习代码的过程中感到了自身还有许多不足。在阅读” Exploring Features in Bayesian Framework for Material Recognition” 这篇论文的过程中，我发现了自己对一些基础的数学知识的掌握有所不足：如狄利克雷分布；对那些研究中常用到的基本算法，还不够熟悉，如 SIFT 特征、HOG 算子。如果要走研究的道路，对这些基本概念都得加深了解。

武家声：因为整个学期都忙于出国的项目，并没有能好好的参与到代码的完成中。只是看了课设要求的论文和组长找到的几篇提高性质的论文。由于近两个学期没有用到模式识别的知识，许多曾经学过的东西都忘了，比如用 SVM 进行训练与分类。虽然知道了通过 LBP 提取特征向量的算法，但是看了组长的代码后还是深感知道算法与能解决问题之间还是有一段距离的。而后面提高性质的论文例如 Exploring Features in a Bayesian Framework for Material Recognition 的知识例如 SIFT 特征、Hog 算子以及 Deep Filter Banks for Texture Recognition and Segmentation 和 Deep TEN: Texture Encoding Network 中运用到了 CNN 等我们在课上也学过。但是时间有限并没能将之实施。通过这次课设，我学到了 LBP 提取特征向量对于纹理识别、材质识别的作用，以及运用 python 进行 SVM 训练与分类的知识。同时，查阅英文论文对于即将出国的我来说也是一项能力的锻炼。

参考文献

- [1] Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi. Deep filter banks for texture recognition and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3828–3836, 2015.
- [2] Eric Hayman, Barbara Caputo, Mario Fritz, and Jan-Olof Eklundh. On the significance of real-world conditions for material classification. In *European conference on computer vision*, pages 253–266. Springer, 2004.
- [3] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.
- [4] Ce Liu, Lavanya Sharan, Edward H Adelson, and Ruth Rosenholtz. Exploring features in a bayesian framework for material recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 239–246. IEEE, 2010.
- [5] Guoying Zhao and Matti Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):915–928, 2007.
- [6] Guoying Zhao and Matti Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):915–928, 2007.

