南開大學
Nankai University

数据挖掘实验报告

学　　号:

姓　　名:

年　　级:

学　　院:

专　　业:

完成日期: 2021 年 3 月 27 日

目录

1 第一次上机实验（LBP 提取图像的纹理特征） …… 3
1.1 实验要求 …… 3
1.2 实验步骤与原理 …… 3
1.2.1 LBP 特征的基本定义 …… 3
1.2.2 直方图特征 …… 4
1.2.3 实现细节（本实验手写 Python 要点） …… 4
1.2.4 复杂度与并行优化 …… 4
1.3 实验结果与分析 …… 5
1.4 实验代码 …… 5
2 第二次上机实验 …… 9
2.1 实验要求 …… 9
2.2 数据分析与处理 …… 9
2.3 实验步骤与原理 …… 9
2.4 实验结论与分析 …… 9
2.5 实验代码 …… 9
3 第三次上机实验 …… 10
3.1 实验要求 …… 10
3.2 数据分析与处理 …… 10
3.3 实验步骤与原理 …… 10
3.4 实验结论与分析 …… 10
3.5 实验代码 …… 10
4 第四次上机实验 …… 11
4.1 实验要求 …… 11
4.2 数据分析与处理 …… 11
4.3 实验步骤与原理 …… 11
4.4 实验结论与分析 …… 11
4.5 实验代码 …… 11

5 第五次上机实验 …… 12
5.1 实验要求 …… 12
5.2 数据分析与处理 …… 12
5.3 实验步骤与原理 …… 12
5.4 实验结论与分析 …… 12
5.5 实验代码 …… 12

第一章　第一次上机实验（LBP 提取图像的纹理特征）

1.1　实验要求

- 1. 给定若干张图像，利用局部二值模式特征 (LBP) 对这些图像进行特征提取
- 2. 图象是 W * H * 3 的矩阵
- 3. 将最终提取到的特征通过 plot 的形式展示，绘制特征曲线图直观对比不同类图片纹理提取到的特征的不同
- 4. 使用 Python 编程实现

1.2　实验步骤与原理

1.2.1　LBP 特征的基本定义

局部二值模式（Local Binary Pattern, LBP）通过比较像素与其邻域像素的灰度关系，编码局部纹理的微结构。给定中心像素 g_c 及以其为中心、半径为 R 的圆形邻域上 P 个等角度采样点的灰度 $\{g_p\}_{p=0}^{P-1}$，标准 LBP 的定义为

$$\mathrm{LBP}_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c)\, 2^p, \qquad s(t) = \begin{cases} 1, & t \geq 0, \\ 0, & t < 0, \end{cases}$$

其中邻域采样点坐标为

$$(x_p, y_p) = \big(x_c + R\cos(2\pi p/P),\ y_c - R\sin(2\pi p/P)\big),$$

本次实验只考虑以 g_c 为中心的九宫格的局部的 LBP 特征。

1.2.2 直方图特征

将整幅图像（或图像块）内的 LBP 代码统计为直方图作为纹理特征：

$$H[k] = \sum_{(x,y)} \mathbf{1}\{\mathrm{LBP}_{P,R}(x,y) = k\}, \qquad k \in \{0, \ldots, 2^P - 1\}.$$

常见做法是对直方图进行 ℓ_1 归一化以消除尺寸影响：

$$\hat{H}[k] = \frac{H[k]}{\sum_j H[j]}.$$

为表征空间布局，可将图像划分为 $M \times N$ 个网格单元，分别计算直方图并按行优先串接，得到最终特征向量。

1.2.3 实现细节（本实验手写 Python 要点）

1. **预处理：**彩色图像先转灰度；可选高斯平滑抑制噪声。
2. 按上述规则计算出图片的 LBP 特征直方图
3. **可视化：**使用 Matplotlib 绘制折线；多类对比时可叠加均值曲线与标准差带。

1.2.4 复杂度与并行优化

- 时间复杂度约为 $O(PWH)$，W, H 为图像宽高；P 通常较小，易于并行/向量化。
- 下面所呈现的代码采用串行方式计算 LBP 特征，但本人也给出了基于 cython 的并行加速版本。

加速计算技巧：

- 使用 cython 的 memoryview 接口直接操作 numpy.ndarray
- 在 cython 层开启 python 的 nogil 模式，绕开全局解释器锁，使用 OpenMP 实现并行计算

并行计算代码及各类计算方法的 benchmark 详见 https://github.com/flyingbucket/machinelearning/tree/main/LBP。

1.3 实验结果与分析

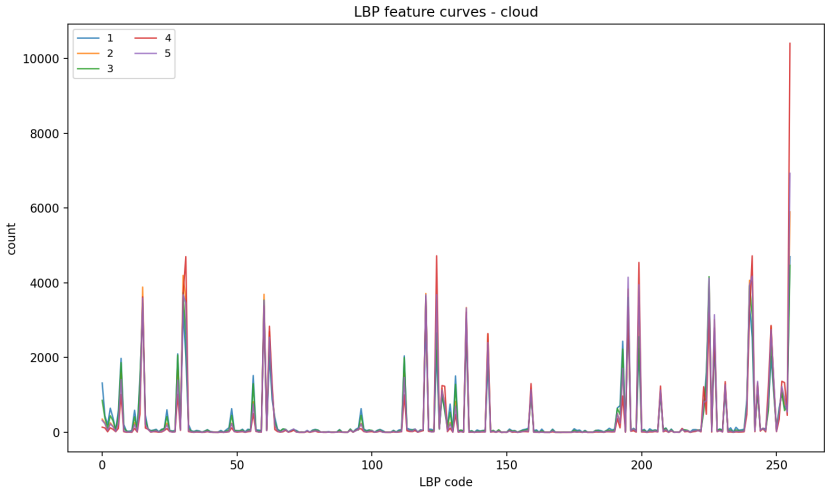

图 1.1: cloud LBP 特征曲线对比图

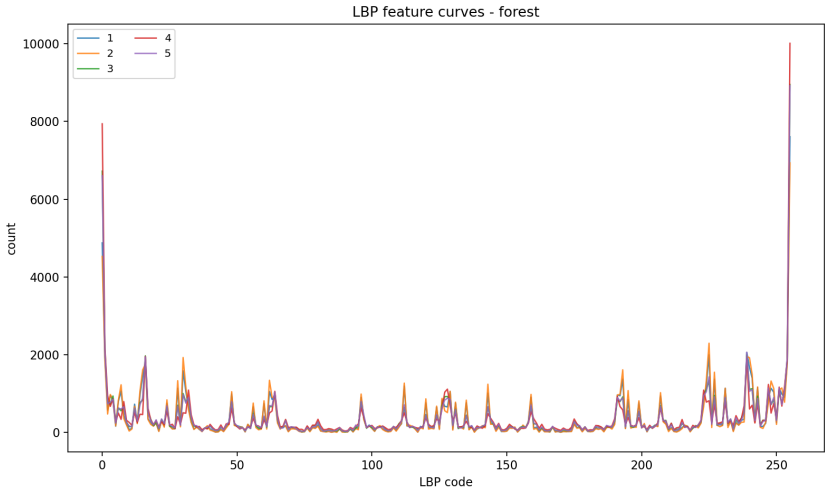

图 1.2: forestLBP 特征曲线对比图

1.4 实验代码

```
1 import numpy as np
2 from matplotlib import pyplot as plt
3 from collections import Counter
4 from PIL import Image
5
```

```
6  class LBP:
7      @staticmethod
8      def _read_img(imPath: str, pad: int = 1, mode: str = "
          reflect") -> np.ndarray:
9          im = Image.open(imPath).convert("L")
10         arr = np.array(im)
11         padded = np.pad(arr, pad_width=((pad, pad), (pad,
          pad)), mode=mode)
12         return padded
13
14     @staticmethod
15     def LBPkernel(im: np.ndarray, x, y) -> int:
16         h, w = im.shape
17         assert x + 2 < h and y + 2 < w, (
18             f"Index out of bound,please check padding. x:{x
          },y:{y},h:{h},w:{w}"
19         )
20         patch = im[x : x + 3, y : y + 3].copy()
21         patch = (patch >= patch[1, 1]).astype(np.uint8)
22         idxs = [0, 1, 2, 5, 8, 7, 6, 3]
23         bits = patch.reshape(-1)[idxs]
24         val = int("".join(map(str, bits)), 2)
25         return val
26
27 def walk_dir(root_dir: str, out_dir: str = "EX1/outputs"):
28     root = Path(root_dir)
29     out = Path(out_dir)
30     out.mkdir(parents=True, exist_ok=True)
31     LBPcyExecutor = LBP()
32
33     for class_dir in sorted([p for p in root.iterdir() if p
          .is_dir()]):
34         hist_list = []
35         img_names = []
36         all_codes = set()
37         for img_path in sorted(class_dir.iterdir()):
38             try:
```

```
39                  res_dict = LBPcyExecutor(str(img_path))  # {code: count}
40                  if not isinstance(res_dict, dict) or len(res_dict) == 0:
41                      print(f"[WARN] 空直方图：{img_path}")
42                      continue
43                  hist_list.append(res_dict)
44                  img_names.append(img_path.stem)
45                  all_codes.update(res_dict.keys())
46              except Exception as e:
47                  print(f"[WARN] 处理失败：{img_path} -> {e}")
48
49          codes = sorted(all_codes)  # 所有出现过的 LBP code
50          X = []  # 每张图对齐后的频率向量
51
52          for h in hist_list:
53              vec = np.array([h.get(c, 0) for c in codes], dtype=np.float64)
54              X.append(vec)
55
56          plt.figure(figsize=(10, 6))
57          for vec, name in zip(X, img_names):
58              plt.plot(codes, vec, linewidth=1.2, alpha=0.85, label=name)
59          plt.xlabel("LBP code")
60          plt.ylabel("count")
61          plt.title(f"LBP feature curves - {class_dir.name}")
62          plt.legend(ncol=2, fontsize=9, loc="best")
63          plt.tight_layout()
64
65          save_path = out / f"{class_dir.name}_lbp_curves.png"
66          plt.savefig(save_path, dpi=160)
67          plt.close()
68          print(f"[OK] Saved: {save_path}")
69
70  if __name__ == "__main__":
```

```
71      dir = "./EX1/data"
72      walk_dir(dir)
```

第二章 第二次上机实验

2.1 实验要求

2.2 数据分析与处理

2.3 实验步骤与原理

2.4 实验结论与分析

2.5 实验代码

第三章 第三次上机实验

3.1 实验要求

3.2 数据分析与处理

3.3 实验步骤与原理

3.4 实验结论与分析

3.5 实验代码

第四章 第四次上机实验

4.1 实验要求

4.2 数据分析与处理

4.3 实验步骤与原理

4.4 实验结论与分析

4.5 实验代码

第五章 第五次上机实验

5.1 实验要求

5.2 数据分析与处理

5.3 实验步骤与原理

5.4 实验结论与分析

5.5 实验代码