

# 酷炫的张昊文的技术文档

360° 旋转酷炫的张昊文

2019 年 7 月 20 日

## 1 友情提示

由于这是第一篇用 LaTeX 写的文档，所以如果太丑了请不要介意，将就着看吧。

## 2 创作历程

首先，感谢赵同学的推荐让我使用 VS code+LaTeX 去踩坑，并且把我推到坑里就不管我了，也感谢一些友好的同学在我努力的想要搭建环境创作时对我的嘲讽与打击，这些人都是为这篇文档做出了不可或缺贡献，感谢他们。

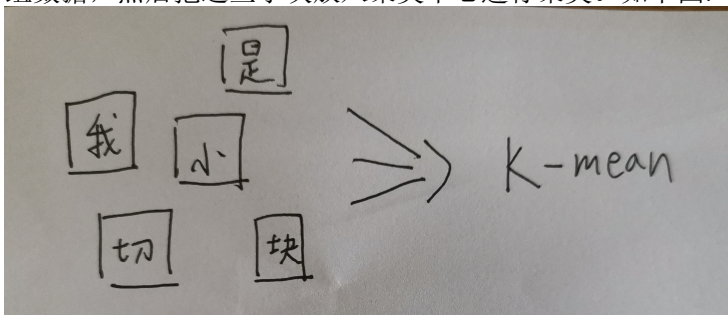
## 3 正文

### 3.1 论文的目的

这篇论文一直在讨论的是一个简单的问题，怎么样去用简单的分类器结构和较少的参数去达到和现在主流的复杂深度模型一样好的性能。后来论文的作者在发现如果对特征提取这方面做很多的工作，那么即使用简单的结构和普通的分类器搭建的模型也能达到很好的精确度。所以本篇论文很大一部分就是在讲如何提取合适的特征。

### 3.2 对特征进行提取

首先我们先随机的对图片进行取小块，然后把这些块通过展平像素，得到一组数据，然后把这些小块放入聚类中心进行聚类。如下图:

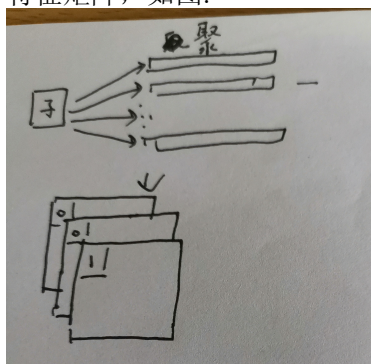


(这个图为

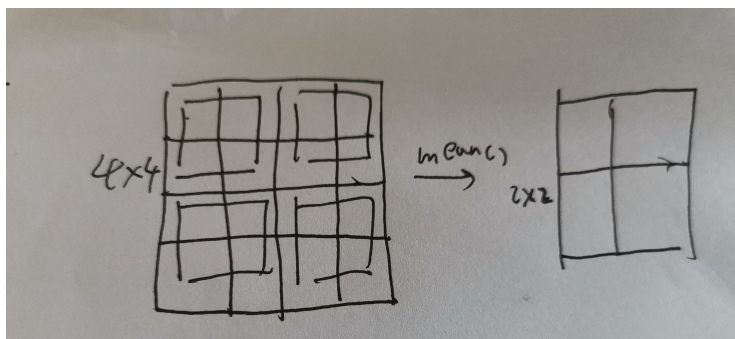
什么会有这么丑的上下标?) 下面为代码:

```
33 def Random_patches(X, pitch_num, pitch_size):
34     """
35     :param X: 输入的图片矩阵 y: 标签
36     :param pitch_num: 每张图片随机取块的数量
37     :param pitch_size: 每块的大小
38     :return:
39     """
40     result = np.zeros((X.shape[0] * pitch_num, pitch_size, pitch_size))
41     for i in range(X.shape[0]):
42         for j in range(pitch_num):
43             num = random.randint(0, X[i].shape[0] - pitch_size - 1)
44             result[i * pitch_num + j] = X[i][np.arange(num, num + pitch_size)
45             _ np.arange(num, num + pitch_size)]
46     return result #, y
```

得到若干个聚类中心，这时再次对原图进行切块，有规律地切，然后把每一块子图与每一个聚类中心求距离作对比，构造一个多层的与切块相对应的特征矩阵，如图:



矩阵中对应子图位置的子块与最近距离的聚类中心对应的层数的位置置为 1，其余为 0。接着我们对这个矩阵每一层进行池化，减少特征的数量



然后把每一层按顺序输出，组成一个行向量，对应的就是一个样本的特征向量。把特征向量和样本送入分类器进行训练，这就是所有特征提取的工作。然后把测试集再代入训练好的模型中就可以进行预测了，太难写了，结束了，不写了，嘻嘻。