



【PyTorch深度学习项目实战100例】—— 聚类算法用于降维，KMeans的矢量量化应用（图片压缩） 第20例



Lineage

前言

大家好，我是阿光。

本专栏整理了《PyTorch深度学习项目实战100例》，内包含了各种不同的深度学习项目，包含项目原理以及源码，每一个项目实例都附带有完整的代码+数据集。

正在更新中~ ✨

我的项目环境：

- 平台：Windows10
- 语言环境：python3.7
- 编译器：PyCharm
- PyTorch版本：1.8.1

项目专栏： [【PyTorch深度学习项目实战100例】](#)

一、聚类算法用于降维，KMeans的矢量量化应用

1、🐍 导入相关库

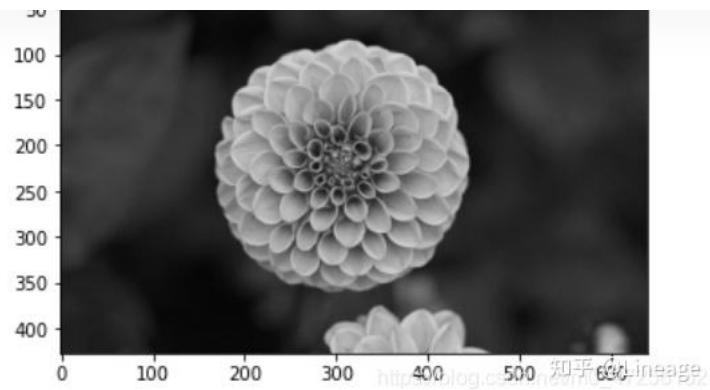
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.metrics import pairwise_distances_argmin
from sklearn.datasets import load_sample_image
from sklearn.utils import shuffle
```

2、🐍 读取数据

```
flower=load_sample_image('flower.jpg')

plt.imshow(flower)
```





在这里插入图片描述

3、👉 数据处理

```
w,h,d=flower.shape
flower_array=np.reshape(flower,(w*h,d))
flower_array

array([[ 2, 19, 13],
       [ 3, 18, 13],
       [ 7, 20, 13],
       ...,
       [ 7, 46, 28],
       [ 8, 45, 28],
       [ 9, 43, 27]], dtype=uint8)
```

图片数据为三维数据，第三维数据为像素信息，将其转化为二维数据，带入模型

4、👉 对数据进行KMeans的矢量量化

```
image_array_sample=shuffle(flower_array,random_state=0)[:1000]
kms=KMeans(n_clusters=k,random_state=0).fit(image_array_sample)

label=kms.predict(flower_array)

image_kmeans=flower_array.copy()
for i in range(w*h):
    image_kmeans[i]=kms.cluster_centers_[label[i]]

pd.DataFrame(image_kmeans).drop_duplicates().shape

image_kmeans=image_kmeans.reshape(w,h,d)
```

shuffle: 该函数是将图片二维数据随机排序，类似于洗牌，取出前1000个色素信息，训练模型，确定质心，然后利用已知质心再去分类所有的二维图片色素信息，分为k类，然后将这k类中的质心色素代替新的图片色素信息，再将二维数据改为三维，成为图片格式。

5、👉 对数据进行随机矢量量化

```
random_sample=shuffle(flower_array,random_state=0)[:k]
labels_random=pairwise_distances_argmin(random_sample,flower_array,axis=0)

image_random=flower_array.copy()
for i in range(w*h):
    image_random[i]=random_sample[labels_random[i]]
```

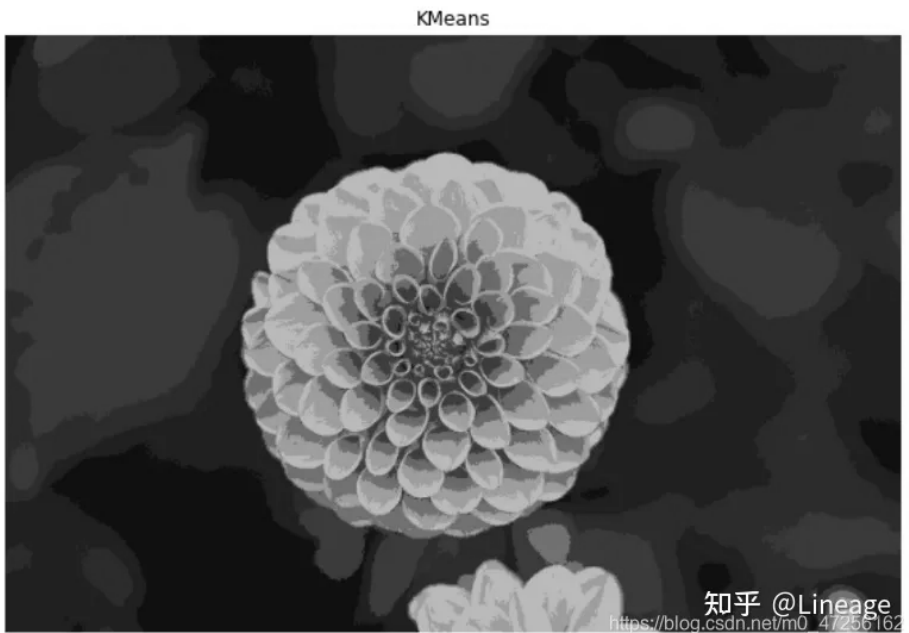
6、🐼 画图，查看结果

```
plt.figure(figsize=(10,10))
plt.axis('off')
plt.title('original')
plt.imshow(flower)
```



在这里插入图片描述

```
plt.figure(figsize=(10,10))
plt.axis('off')
plt.title('KMeans')
plt.imshow(image_kmeans)
```



在这里插入图片描述

```
plt.figure(figsize=(10,10))
```

由上图可以看出用KMeans压缩的图片和原图片差不多，因为KMeans是将所有色素信息聚类，不同类的色素信息用该类的中心点代替，所以颜色并没有什么缺失 然而随机取出的色素来代替，颜色会有缺失，因为随机选取色素的过程中丧失了主要的色素信息。

完整源码

发布于 2022-12-06 16:53 · IP 属地湖南

人工智能 神经网络 深度学习 (Deep Learning)

写下你的评论...



还没有评论，发表第一个评论吧



基于PyTorch实现文本聚类算法
DBSCAN附带案例

那个百分十... 发表于PyTor...

		P	N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

利用pytorch构建分类模型时
accuracy、precision、reca...

CC881C40

pytorch卷积层与池化层
尺寸的计算公式详解

pytorch卷积层与池化层输
尺寸的计算公式详解 要设计神
经网络的结构，必须匹配层与
间的输入与输出的尺寸，这就
较好的计算输出尺寸 先列出公
卷积后，池化后尺寸计算...

我是一只棒... 发表