大作业期末报告

1. 概述

随着信息时代的发展，人类产生的信息量成几何倍数增加，其中对于图像的检索要求也随之增长。我们的大作业旨在找到一种基于sift以及surf的图像检索方式，通过建立自己的图像库来是的图像检索更加便利。在该项目中主要实现相似图像的选取功能，即通过用户选择的图像来检索图像库中相似的图像，并尽可能取得更高的检索成功率。

1. 关键技术

i9-12900KF 3080 Win11

Pycharm+anaconda+python3.6

1. 算法描述

Train:

For file in trainpath:

Des = sift(file)

Train\_des.append(des)

Kmeans.train(train\_des)

For file in dirpath:

Vector = des2vector(des)

Train\_vec.append(Vector)

Train\_label.append(file.label)

Idf = calidf(train\_vec)

Train\_vec=caltfidf(train\_vec, idf)

Knn.fit(train\_vec)

Nn.fit(train\_vec)

Save()

Compare:

Des = sift(img)

Vector = des2vector(des)

Vector = caltfidf(vector, idf)

Img\_target = nn.search(vector)

Vector = get\_average(Img\_target.vector)

Img\_target = nn.search(vector)

For img in img\_target:

Dic[img.label] += 0.01 / (img.dis + 0.01)

Def compare(a, b):

If dic[a] > dic[b]:

Return -1

Elif dic[a] == dic[b]:

Return 0

Else:

Return 1

Label = sorted(list(set(img.label)), cmp = compare)

Show(img\_target) by label

Eval:

For file in testpath:

Des=sift(file)

Vector = des2vector(des)

Test\_vec.append(vector)

Test\_label.append(file.label)

Test\_vec=caltfidf(test\_vec, idf)

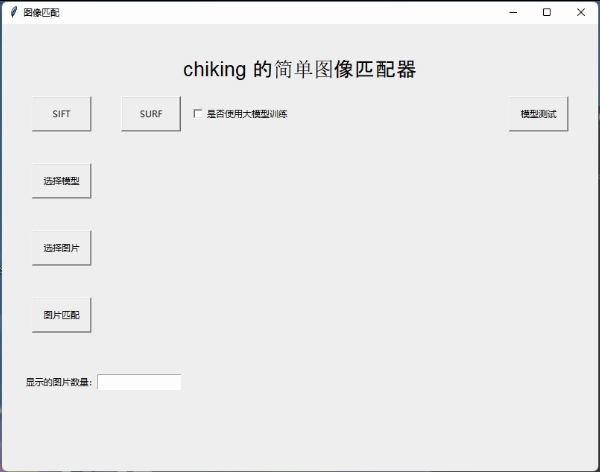
Test\_pred=knn.predict(test\_vec)

Print(recall(test\_pred, test\_label))

Print(precision(test\_pred, test\_label))

Print(map(test\_pred, test\_label))

1. 系统实现



界面的各个按钮的功能以及大致逻辑如下：

1. Sift surf 是否使用大模型训练

主要功能为模型训练，两个按钮分别代表使用sift和surf算法计算图片的特征点。是否使用大模型训练的主要差别在于kmeans聚类时聚类的数量，大模型聚类时将特征点分为100类，在后面的测试中准确度更高但训练时间更长在完整的数据集中训练大约需要15分钟左右；小模型聚类将特征点分为20类，模型准确度稍低但训练时间短，在完整的数据集中训练大约需要2分钟左右。模型训练完后将被自动保存到当前文件目录下方便后期调用。

1. 模型选择

模型选择可以通过文件选择系统调用已经训练好的模型，被选择的模型将被显示在界面上。没有被选择的模型将不能进行图片匹配也不能进行模型测试。

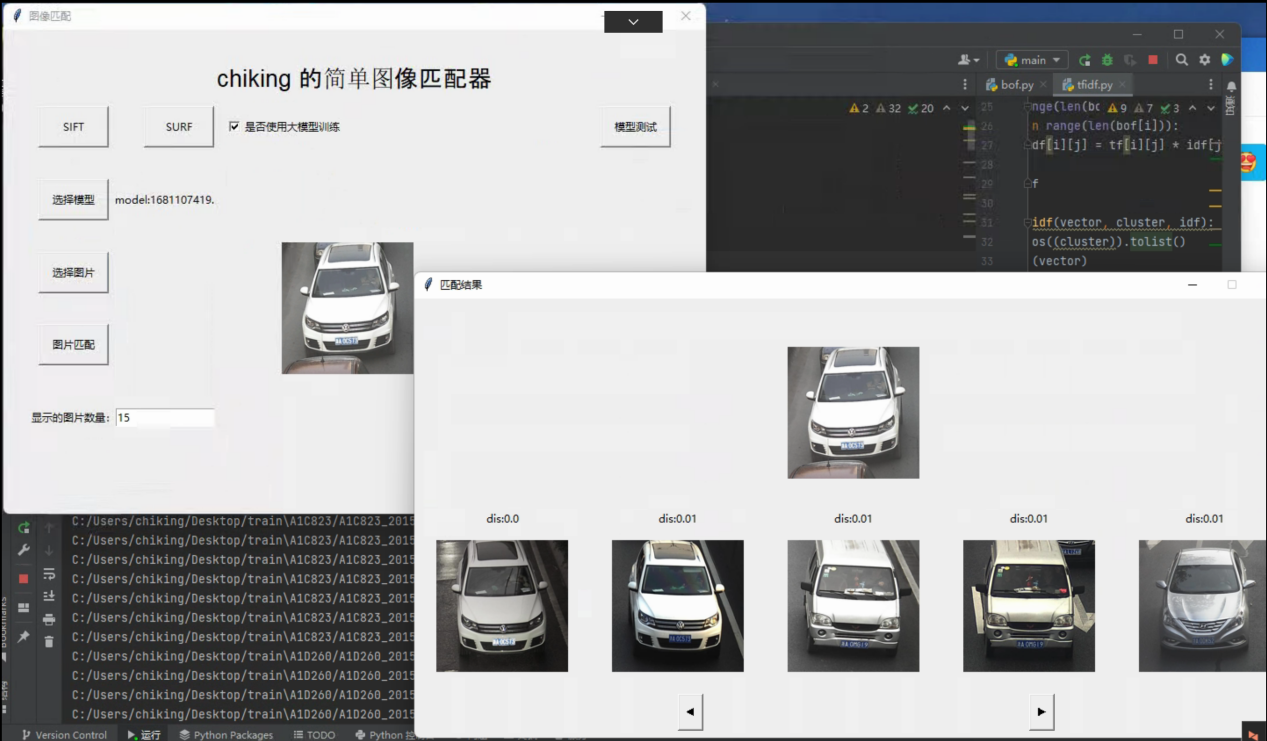
1. 选择图片

从系统文件中选择图片用于接下来的检索，被选择的图片将会显示在界面上。没有被选择图片将不能进行图片匹配。



1. 图片匹配 显示图片数量

在选择完模型和图片之后将能够使用图片匹配功能。通过设置显示图片数量设置跳出的新窗口中的图像数量。新窗口中可以通过下方的左右键翻页。



1. 模型测试

使用测试集对knn模型进行测试，并将结果返回在界面上，同时绘画ROC图，并通过matplotlib显示出来。



以下介绍核心代码逻辑

1. Train

该函数用于训练模型

首先遍历整个文件夹，使用sift或者surf算法获得每个图像的特征点，将所有的特征点放入同一个list中作为kmeans的训练数据，从而对每个特征点进行聚类。遍历所有图片，预测每个图片的特征点分别属于哪个类，并使用bof算法进行统计得到特征向量。再通过bof算法统计的特征向量进行tfidf统计分别获得更具有代表性的特征向量。而后调用sklearn中KNeighborsClassifier训练模型来对整个算法流程进行评估，调用sklearn中的NearestNeighbors训练模型进行检索。最后将使用的特征点提取算法，所有的图片，两个模型，kmeans，训练集的标签以及在tfidf中计算法的idf一起保存。

1. Eval

该函数用于评估训练出来的模型。

遍历整个测试集文件夹，使用与模型相同的算法对图像进行特征提取并保存测试集的标签，并参考train中的方法提取图像的特征向量。通过计算得到的特征向量以及先前保存的测试集的标签来对knn模型进行测评。

1. Compare

该函数用于展示与被检索图像前k相似的图像。

使用与模型相同的算法对被测试图像进行特征点提取，并用kmeans对每个特征点的所属的类编号进行预测，使用bof算法进行统计，而后使用先前保存的idf并计算当前图像的tf值来得出这个图像的tfidf向量，通过tfidf向量在NearestNeighbors模型中找到最接近的前k个图像。对前k个图像的特征向量取平均得到一个新的向量，再将这个新的向量放入NearestNeighbors进行查询拓展，得到新的一批前k个图像以及其与被查询向量的相似值。0.01除以相似值加上0.01并对相同的label进行求和，根据求和得到的大小对每个label进行重排序，求和较大的label在被显示的时候排在前，同一label根据相似值从小到大排序。

1. 结语

通过这次学习是的我对于图像识别方面的知识有了更加深刻的认识，对于surf，kmeans以及tfidf算法的了解和tkinter，opencv库的使用更加深刻，对于重排序算法、查询拓展算法以及如何在特征提取算法不变的情况下使用算法提升查询准确率的了解更加深刻。