1. 什么是数据驱动的视觉识别？简述其基本流程。（10 分）

首先我想从视觉识别任务的角度入手去阐述数据驱动的视觉识别的内容及其合理性。众所周知，视觉识别任务就是从视觉层面的信息中提取出一组模态。这个模态可能是物体的种类（ImageNet）、可能是目标的运动情况（tracking, pose estimation）、更高层次的场景上下文语义信息（segmentation, image caption）。不使用 learning 的传统方法往往是人为引入一些先验假设后对观测数据直接建模。比如课上提到的非常著名的光流三假设对物体的运动进行建模，又比如 tracking 中的卡尔曼滤波假设了所有状态量符合符合高斯分布等。但是为了达到更好的建模效果，不得不引入更多的假设，这会导致在泛化性有明显的不足。同时，引入的假设本身就是人为从历史以往观测数据中进行的模型特性的再抽象，属于经验结论，并不像数学那么严谨，这往往会引入人为的偏差，在一些极高精度要求的环境下可能会非常显著。在这种情况下，我们开始思考如何对以往的经验数据进行更好的利用，这也是在数据科学中许多人正在研究的事情。只有充分利用了数据，才能对各种情况下的识别任务有着更好的鲁棒性。基于数据驱动的视觉识别应运而生。从最简单的 kNN 到 CNN 到目前在这波人工智能浪潮里的各种各样千奇百怪的结构，都是在大量以往数据的训练下，在模型准确度、效率、鲁棒性、泛化性中找到一个更优解的过程。传统方法下的我们是指定一个模态，这个模态越详细越好，越把一个任务分解成越多的“基元”越好，而现在我们只需要设立几个超参数，剩下的工作交给那些显卡就行。综上所述，我认为数据驱动的视觉识别有以下流程：

1. 确定模型结构
2. 训练
3. 测试

2. 在数据驱动的识别框架中，如果想得到好的识别结果，哪些因素是重要的？把这些因素按照重要性排序，并说明理由。（10 分）

1. 数据，正如它的名字那样，在识别的路上，数据的重要性一定是第一位的。首先，并不是所有实验室都能像李飞飞组那样为了解决一个问题能够创造一个几千万的数据集，数据的量本身就是一个限制精度的大问题。其次就是数据的质量，通常自己在实验室中拍摄的数据是最好的，因为数据和研究目的是严格一一对应的，甚至会为了最好的效果在拍摄时就进行几轮筛选。但是倘若我们需要用到互联网中已有的极为丰富的数据，这可能是一些学界已经存在的数据集，也有可能是搜索引擎上爬取的 wild data，我们需要明确数据是否和研究目的契合，比如研究家庭的场景就不应该存在一些户外运动的图片。同样，标签也是重要的信息。无论是自己打还是众包平台，总会出现一些不可避免的人工误差。在诸如 VOT 的识别竞赛上，因为给定了训练集，大家都在同一起跑线上，但是在实际任务中，数据的重要性不言而喻。

2. 模型，在任何一个视觉的子领域，SOTA 可能每时每刻都在刷新。这些论文往往都使用相同的数据集刷分（eg:coco），那么数据本身的影响可以忽略了。接下来就是模型的影响，毕竟模型每年都在迭代。如果在 2020 年在 kaggle 上写一个 kNN，可能前 50%都进不了。接下来我想就我的理解去讲一下模型选取上的一些方法。但是我作为大二，还没有实操的经验。可能还是比较泛，比较“空中楼阁”。我想拿 point net 举个例子，这可以算是 3D 点云领域的祖师爷。在 point net 之前，处理 3D 点云往往是使用 3D 卷积核、多视图投影等等，但是

point net 首次用 raw point cloud 数据进行处理，使用对称函数去处理点云的无序性，使用局部特征和全局特征融合的方法去做场景 3D 点云分割的任务。这看起来非常简单，但是的确是非常基础的奠基性的工作。因为直接处理 raw point cloud 数据符合最直观的理解，也符合奥卡姆剃刀原理。所以在选取模型的时候，我们一定要对数据本身进行思考和理解，找到最能提取出数据本身特征的模型。当然，这种机遇往往可遇不可求，我们更多地还是在给模型调参期望有几个点的提升的路途中。这并不是玄学，正确的方法应该是像 mean-shift 一样逐渐收敛的结果。这需要我们对领域内的各个算法，以及数据驱动的算法本身有大量的调研和理解。

3. 训练，训练的一些 tricks 就不再赘述了，这在数据科学里有大量的分析和论证。我认为要想得到好的识别结果，首先是严格控制训练的进度，不能过拟合。在此基础上，卡多就行。

3. 简述在 Bag of Words （BoW）模型中，矢量量化是如何完成的？（10 分）

用 K-means 对提取的 N 个 SIFT 特征进行聚类，K-Means 算法是一种基于样本间相似性度量的间接聚类方法，此算法以 K 为参数，把 N 个对象分为 K 个簇，以使簇内具有较高的相似度，而簇间相似度较低。聚类中心（视觉词）有 k 个，码本的长度也就为 k，计算每一幅图像的每一个 SIFT 特征到这 k 个视觉词的距离，并将其映射到距离最近的视觉词中（即将该视觉词的对应词频+1）。完成这一步后，每一幅图像就变成了一个与视觉词序列相对应的词频矢量。但由于每一幅图像的 SIFT 特征个数不定，所以需要归一化。

4. 收集一组图像，建立一个小的数据集，划分训练集和测试集，在此基础上，完成下列任务：

(1) 建立与该数据集对应的视觉词典（码本），并可视化 2～3 个代表性码字，及与这些码字对应的实例图像局部图像块；（建议用 SIFT 特征，也可以用别的特征）（20 分）

(2) 进行基于 BoW 的图像识别或图像检索，并给出完整实验报告。（50 分）

实验目的

1. 加深对 Bag of words 词袋模型在视觉领域中的理解。
2. 积累实际的 Bag of words 词袋模型应用经验。

实验步骤

1、 假设训练集有 M 幅图像[1]，对训练图象集进行预处理。包括图像增强，分割，图像统一格式，统一规格等等。

2、提取 SIFT 特征。对每一幅图像提取 SIFT 特征（每一幅图像提取多少个 SIFT 特征不定）。每一个 SIFT 特征用一个 128 维的描述子矢量表示，假设 M 幅图像共提取出 N 个 SIFT 特征。

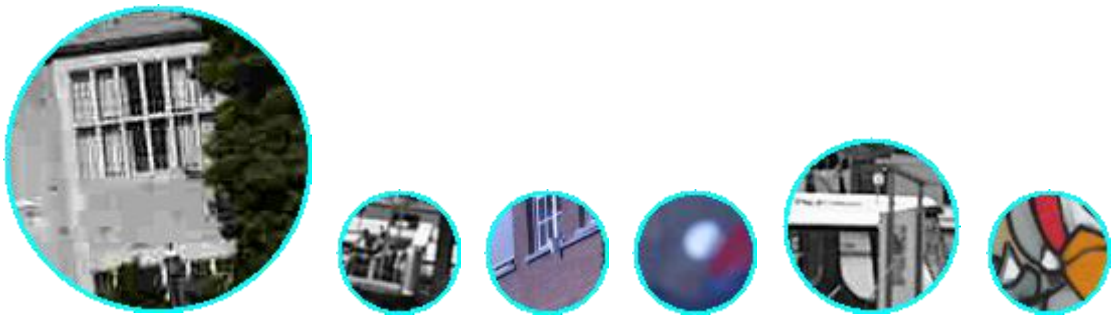
3、用 K-means 对 2 中提取的 N 个 SIFT 特征进行聚类，K-Means 算法是一种基于样本间相似性度量的间接聚类方法，此算法以 K 为参数，把 N 个对象分为 K 个簇，以使簇内具有较高的相似度，而簇间相似度较低。聚类中心有 k 个（在 BOW 模型中聚类中心我们称它们为视觉词），码本的长度也就为 k，计算每一幅图像的每一个 SIFT 特征到这 k 个视觉词的距离，并将其映射到距离最近的视觉词中（即将该视觉词的对应词频+1）。完成这一步后，每一幅图像就变成了一个与视觉词序列相对应的词频矢量。

4、构造码本。码本矢量归一化因为每一幅图像的 SIFT 特征个数不定，所以需要归一化。测试图像也需经过预处理，提取 SIFT 特征，将这些特征映射到为码本矢量，码本矢量归一化，最后计算其与训练码本的距离，对应最近距离的训练图像认为与测试图像匹配。
实验数据记录、实验结果计算

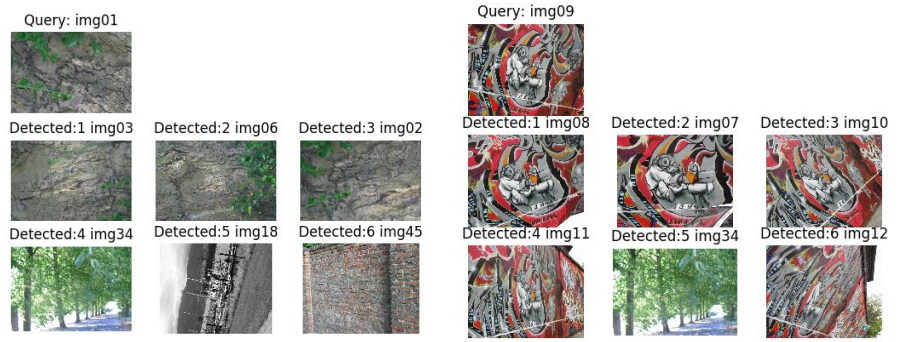
本实验所有数据集和代码均已上传 Github

实验结果

对视觉词进行可视化，并按照优先级排序，如下图所示。



通过视觉词计算测试集图片在每张训练集图片上的相似度（得分），并按照顺序输出。



通过搜索结果我们发现，每次搜索的结果均正确，故精确率为 100%，在所有情况下，同一场景拍摄的不同图片排在最前，成功实现了图像检索的效果。

性能分析

在执行 k-means 聚类的过程中首先计算聚的 k 个类的聚类中心，然后用找到的 k 个聚类中心进行再次聚类。当聚类中心不再变动时，聚类停止。设此时迭代次数为 t 次。设单次计算两个 SIFT 特征的距离的消耗为 d，所以 k-means 聚类复杂度为 $O(kNdt)$。由于 SIFT 的描述子有 128 维度，可得 k-means 约需要 $10^3 \times 10^5 \times 128 \times 100 = 10^{12}$ 次计算。在尝试进行可视化的过程中，遇到了一些性能的问题。假设有 k 个聚类中心，M 幅图像共提取出 N 个 SIFT 特征。在实验中得知 $k \approx 10^3$, $N \approx 10^5$，如果使用二重匹配匹配聚类

中心最近的特征点作为数据可视化的依据，需要在$O(kN)$时间复杂度下进行$\approx 10^8$次图像的读写和掩膜操作。通过进一步的分析，我选用了 kd 树[2]这种数据结构进行优化。它是一种对 k 维空间中的实例点进行存储以便对其进行快速检索的树形数据结构。主要应用于多维空间关键数据的搜索（如：范围搜索和最近邻搜索），在通常情况下，kd 树将查询最近邻的平均时间复杂度降低到了$O(\log N)$，故总体上为$O(k\log N)$，执行次数为 17000 次。

参考文献

[1]Do Better.BoW 词袋模型原理学习及 Python 实现[EB/OL].https://blog.csdn.net/weixin_41234001/article/details/102913445,2019-11-05.
[2]百度百科.kd-tree[EB/OL].https://baike.baidu.com/item/kd-tree/2302515,2020-09-29.