报告人：张志逸

实习项目：计算机视觉——图像检索

日期：2017年8月

一、实验背景

人类的眼睛捕捉到图像后，大脑会完成一系列的复杂分析从而为人眼可见的图像进行分类，进而人类可以通过学习完成对不同类别图像进行判断和定义。计算机视觉是一门研究如何使机器“看”的科学，更进一步的说，就是是指用摄影机和电脑代替人眼对目标进行识别、跟踪和测量等机器视觉，并进一步做图形处理，使电脑处理成为更适合人眼观察或传送给仪器检测的图像。作为一个科学学科，计算机视觉研究相关的理论和技术，试图建立能够从图像或者多维数据中获取‘信息’的人工智能系统。 可以用来帮助做一个“决定”的信息。因为感知可以看作是从感官信号中提取信息，所以计算机视觉也可以看作是研究如何使人工系统从图像或多维数据中“感知”的科学。

原理上来说，计算机视觉就是用各种成象系统代替[视觉器官](https://baike.baidu.com/item/%E8%A7%86%E8%A7%89%E5%99%A8%E5%AE%98" \t "_blank)作为输入敏感手段，由计算机来代替大脑完成处理和解释。计算机视觉的最终研究目标就是使计算机能象人那样通过视觉观察和理解世界，具有自主适应环境的能力。要经过长期的努力才能达到的目标。因此，在实现最终目标以前，人们努力的中期目标是建立一种视觉系统，这个系统能依据视觉敏感和反馈的某种程度的智能完成一定的任务。例如，计算机视觉的一个重要应用领域就是自主车辆的视觉导航，还没有条件实现象人那样能识别和理解任何环境，完成自主导航的系统。因此，人们努力的研究目标是实现在高速公路上具有道路跟踪能力，可避免与前方车辆碰撞的视觉辅助驾驶系统。这里要指出的一点是在计算机视觉系统中计算机起代替[人脑](https://baike.baidu.com/item/%E4%BA%BA%E8%84%91" \t "_blank)的作用，但并不意味着计算机必须按人类视觉的方法完成视觉信息的处理。计算机视觉可以而且应该根据[计算机系统](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%B3%BB%E7%BB%9F" \t "_blank)的特点来进行视觉信息的处理。但是，[人类视觉系统](https://baike.baidu.com/item/%E4%BA%BA%E7%B1%BB%E8%A7%86%E8%A7%89%E7%B3%BB%E7%BB%9F" \t "_blank)是迄今为止，人们所知道的功能最强[大和](https://baike.baidu.com/item/%E5%A4%A7%E5%92%8C" \t "_blank)完善的视觉系统。如在以下的章节中会看到的那样，对人类视觉处理机制的研究将给计算机视觉的研究提供启发和指导。因此，用计算机信息处理的方法研究人类视觉的机理，建立人类视觉的[计算理论](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E7%90%86%E8%AE%BA" \t "_blank)，也是一个非常重要和信人感兴趣的研究领域。这方面的研究被称为计算视觉（Computational Vision）。计算视觉可被认为是计算机视觉中的一个研究领域。目前，除了上述的车辆导航领域，计算机视觉技术已被多家公司将其与人工智能（Artificial Intelligence）整合并且运用。例如，美国Amazon已将该技术应用于正在调试阶段的无人超市Amazon Go。 顾客进入超市之后，摄像装置将自动捕捉每一位顾客的面部特征进行识别并且匹配该顾客的账户。完成一系列匹配之后，顾客即可进入超市挑选货物。每确认购买一件货物后系统将自动从客户的Amazon账户中扣款，无需排队结账。

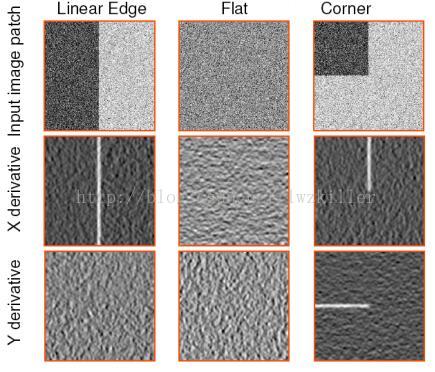
本次实验是便是基于计算机视觉理论基础构建图像检索系统。在五天的时间中学习多种局部特征和全剧特征分析算法。将所学习到的算法用MATLAB代码实现。最后使用一个GUI界面，可以允许使用者打开一张图片并在给定的图片库中搜索相似图片，按照相似度从高到底将匹配图像展示出来。

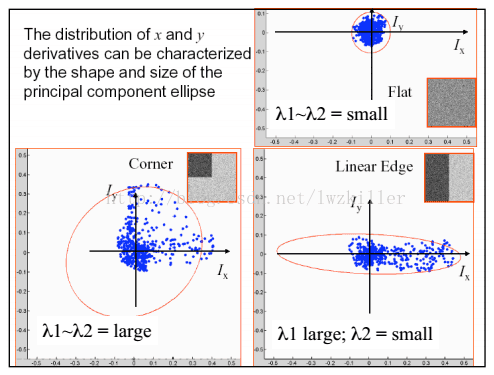
二、关键技术

三、如何实现

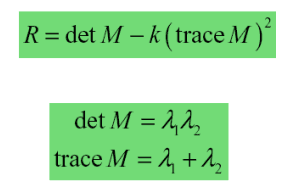
1. Harris检测子

Harris角点检测算子实质上就是对Moravec算子的改良和优化。算法基本思想是使用一个固定窗口在图像上进行任意方向上的滑动，比较滑动前与滑动后两种情况，窗口中的像素灰度变化程度，如果存在任意方向上的滑动，都有着较大灰度变化，那么我们可以认为该窗口中存在角点。Harris算法通过对x和y方向上的梯度进行分析来判断是否为角点。

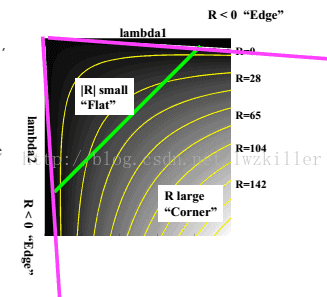




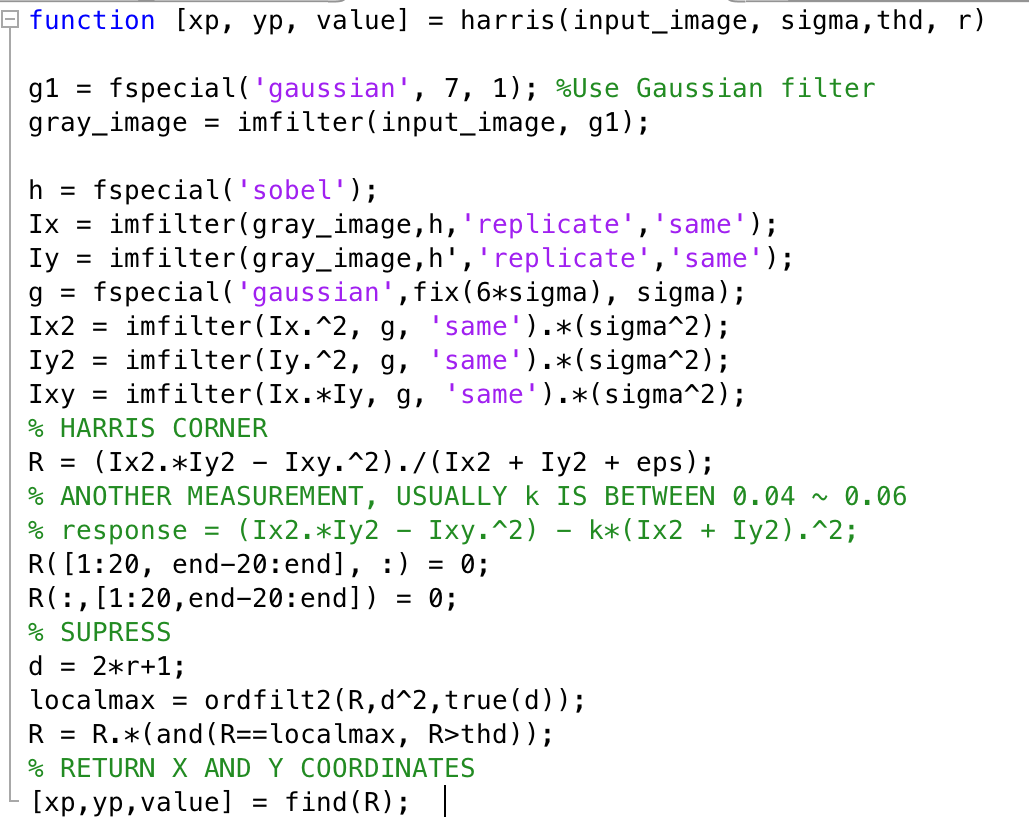
我们用以下方式度量角点响应。



其中k是常量，一般取值为0.04~0.06，这个参数仅仅是这个函数的一个系数，它的存在只是调节函数的形状而已。我们可以以下图来通过R值判断是否为角点。在我们的实验中我们通过改变这一点的灰度来标记。



代码实现：



1. 利用欧氏距离进行特征点匹配

欧式距离算法的核心是：设图像矩阵有n个元素（n个像素点），用n个元素值(x1，x2，...，xn)组成该图像的特征组（像素点矩阵中所有的像素点），特征组形成了n维空间（欧式距离就是针对多维空间的），特征组中的特征码（每一个像素点）构成了每一维的数值，就是x1（第一个像素点）对应一维，x2（第二个像素点）对应二维，. . .，xn（第n个像素点）对应n维。在n维空间下，两个图像矩阵各形成了一个点，然后利用数学上的欧式距离公式计算这两个点之间的距离，距离最小者就是最匹配的图像。

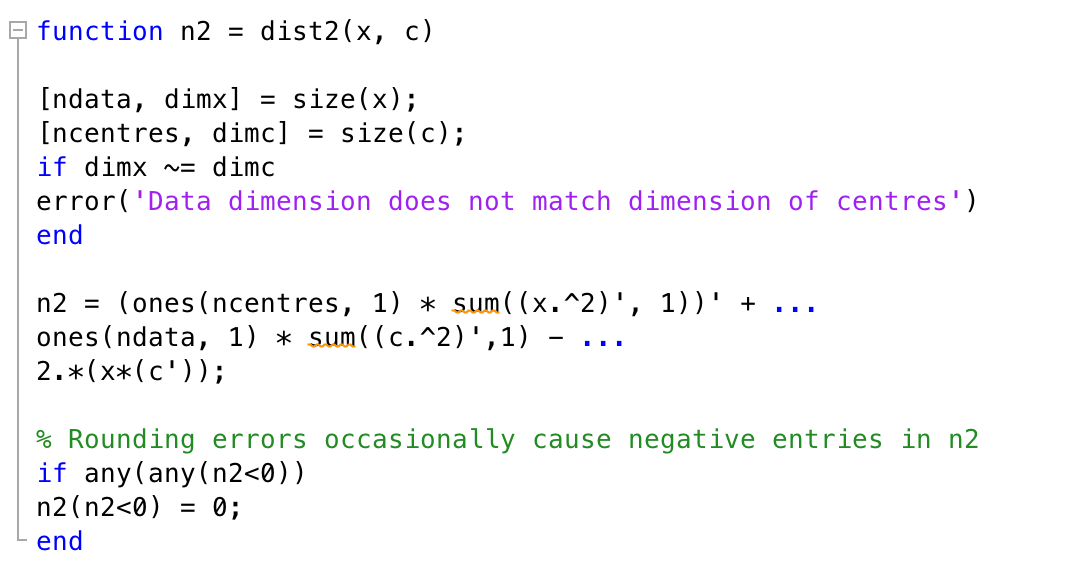
 欧式距离公式：

 点A = （x1, x2, ... , xn）

 点B = （y1, y2, ... , yn）

 AB^2  = (x1-y1)^2+(x2-y2)^2+...+(xn-yn)^2

代码实现：



  AB就是所求的A，B两个多维空间中的点之间的距离。

1. 哈希算法&局部敏感哈希

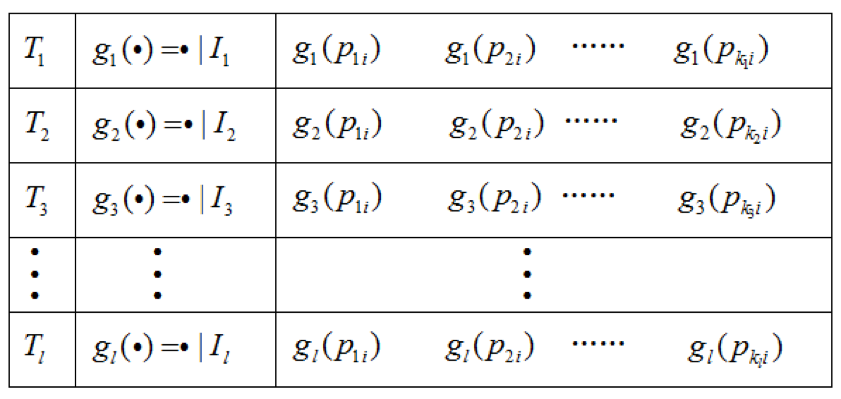
哈希(hash 散列，音译为哈希)算法将任意长度的二进制值映射为固定长度的较小二进制值，这个小的二进制值称为哈希值。哈希值是一段数据唯一且极其紧凑的数值表示形式。如果散列一段明文而且哪怕只更改该段落的一个字母，随后的哈希算法都将产生不同的值。要找到散列为同一个值的两个不同的输入，在计算上是不可能的，所以数据的哈希值可以检验数据的完整性。哈希表是根据设定的[哈希函数](http://baike.baidu.com/view/549615.htm" \t "_blank)H(key)和处理冲突方法将一组关键字映象到一个有限的地址区间上，并以关键字在地址区间中的项作为记录在表中的存储位置，这种表称为[哈希表](http://baike.baidu.com/view/329976.htm" \t "_blank)，所得存储位置称为哈希地址。作为线性数据结构与表格和队列等相比，哈希表无疑是查找速度比较快的一种。

局部敏感哈希与最近邻搜索、汉明距离结合，形成一种可以进行高维度图像检索的算法。最近邻检索就是根据数据的相似性，从数据库中寻找与目标数据最相似的项目，而这种相似性通常会被量化到空间上数据之间的距离，可以认为数据在空间中的距离越近，则数据之间的相似性越高。当需要查找离目标数据最近的前k个数据项时，就是k最近邻检索（K-NN）。汉明距离是使用在数据传输差错控制编码里面的，汉明距离是一个概念，它表示两个（相同长度）字对应位不同的数量，我们以d（x,y）表示两个字x,y之间的汉明距离。对两个字符串进行异或运算，并统计结果为1的个数，那么这个数就是汉明距离。汉明距离是以理查德·卫斯里·汉明的名字命名的。在信息论中，两个等长字符串之间的汉明距离是两个字符串对应位置的不同字符的个数。换句话说，它就是将一个字符串变换成另外一个字符串所需要替换的字符个数。例如：1011101 与 1001001 之间的汉明距离是 2；2143896 与 2233796 之间的汉明距离是 3；"toned" 与 "roses" 之间的汉明距离是 3。局部哈希算法（LSH）将高维空间中的元素视为点并赋以坐标值，坐标值为正整数。通过一组哈希函数将空间所有点映射到n个哈希表中，n=|F|，即每个哈希函数f∈F对应一个哈希表，每个哈希表都存放着空间所有的点。对于给定的查询子q，分别计算f1(q)、f2(q)、…、fn(q)，fi ∈F，i=1,2,…,n 。以所有fi(q)落入的哈希表Ti中的桶中所有点作为候选集，比较其与q之间的距离，选出距离最近的K个点（K-NNS）。

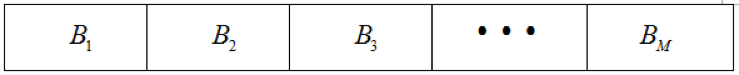
运用局部哈希算法，首先进行预处理。对于任意一点p∈P，P为d维空间。设={x1,x2,…,xd}，将空间P映射到d'维Hamming空间Hd'，映射方法如下：

p→p'∈P，p'=Unaryc(x1)Unaryc(x2)···Unaryc(xd)，Unaryc(xi)表示xi个1紧跟C-xi个0，C表示空间P中任意点p坐标的最大值。这种映射是保距的，即 p,q∈P，d1(p,q)=dH(p,q)。其中d1表示定义在空间P上l1准则下的欧几里得距离，dH表示定义在Hd'空间下的Hamming距离。因此原问题——找出空间P中与查询子q距离最近的K个点——转化为Hd'在空间中的 -NNS问题。在算法实现中，并没有显式将p转化为p'，而是用别的计算方法利用了这种转化，使算法易于描述并且实现的时空效率高。下面定义哈希函数的过程中，p代表了原始点的向量形式（即p'），即不会区分p与p'。

接着，定义哈希函数族：定义I={1,2,…,d'},定义正整数l，取l个i的子集，分别记为，，…，。定义p|I为向量p在坐标集I上的投影，即以坐标集I中每个坐标为位置索引，取向量p对应位置的比特值并将结果串联起来。比如I={1,5,7,8}，p =110010001110（对应原始点p={2,1,3}，d=3,C=4），则p|I=1100。记=p|I，我们将空间P中的所有点利用哈希函数族都存入哈希表的哈希桶中。哈希族的形式如下图所示：



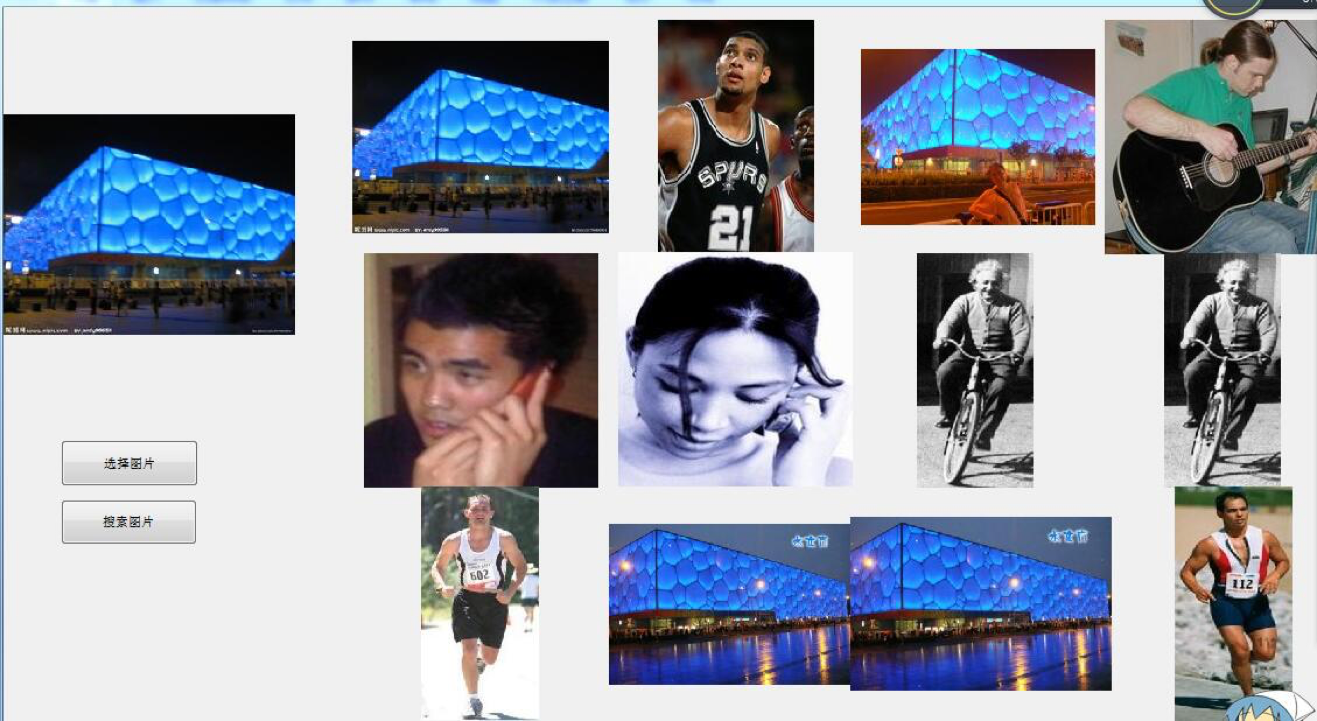
因为哈希桶的数量可能会很大，因此采用第二层哈希，利用标准哈希函数将所有的哈希桶映射到一个大小为M的哈希表中。记该哈希表中的哈希桶最大容量为B，在算法中采取的冲突解决方法是：当一个哈希桶内点的个数超过B时，则新分配一个大小为B的桶并将该新桶连接到原来的桶中，而在实现的过程中，采取了更简单的方法：当一个哈希桶中点的个数超过B时，则不能再有点插入，当有新点分配到该桶中时，该点会被分配到其他未满的哈希桶中。新的哈希表如下表所示：

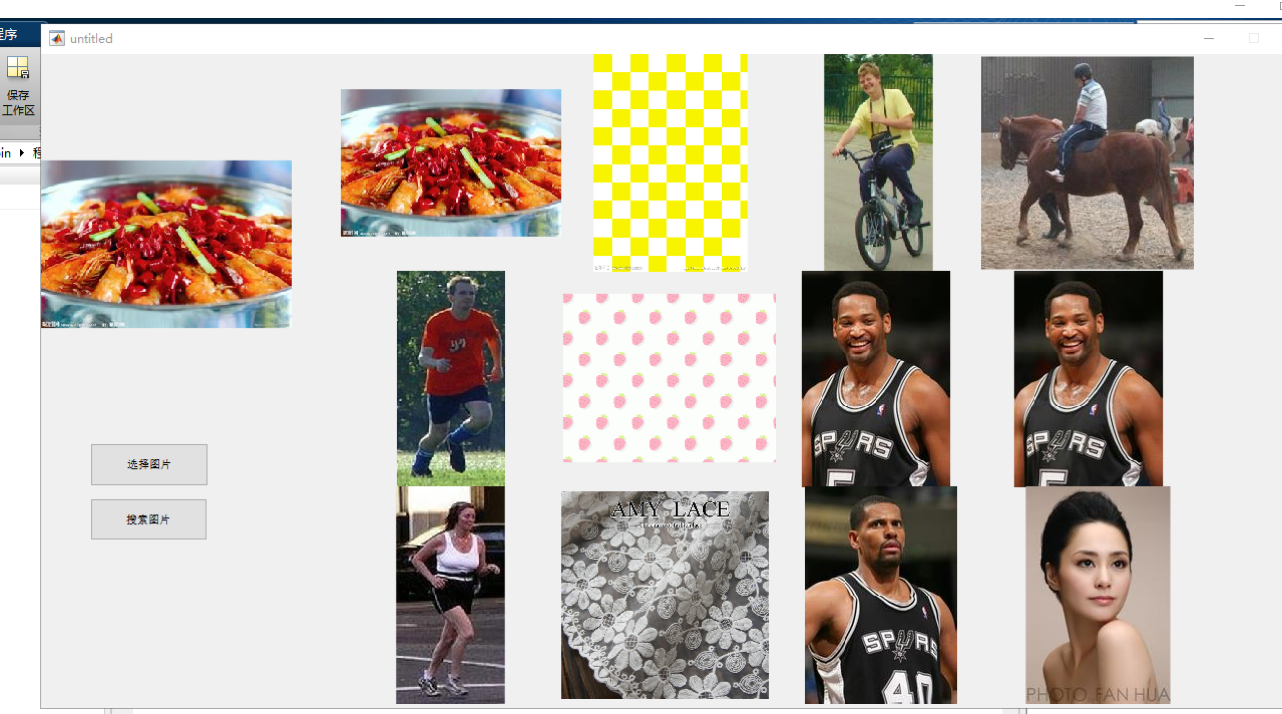


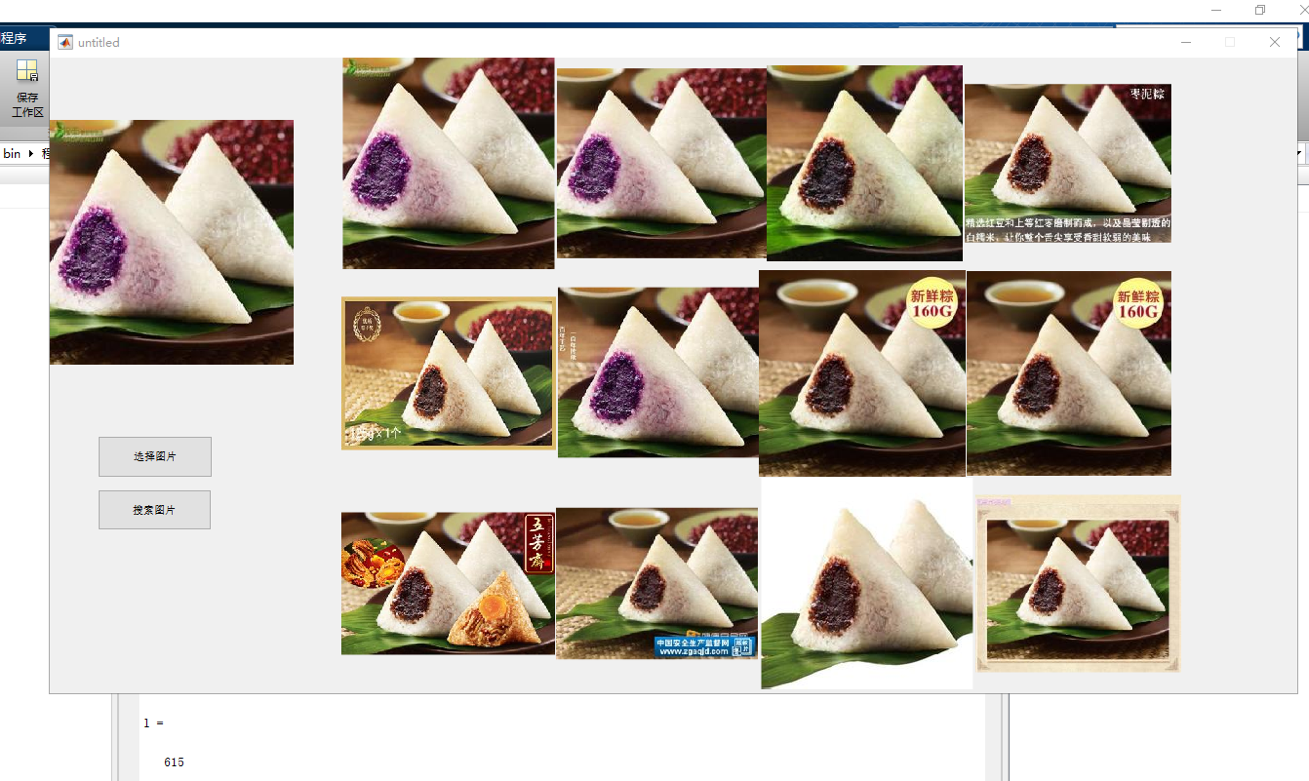
最后进行查询操作：对于给定的查询子q，计算，g1(q)，g2(q)…，gl(q)，取出gi(q)对应哈希桶中所有的点（即满足gi(q)=gi(p)的点p的集合）作为候选集。最后在候选集中选出K个距离查询子q最近的K个点。

由于导师提供了哈希函数的具体代码，此处不贴代码。

四、效果展示







五、遇到问题

最早遇到的问题是从Harris角点检测部分开始的，由于自身的数学能力还不够从而看不懂数学公式表达的意义。然后从一些公开课网站上找了一些与线性代数相关的视频和资料补了一下后，大致可以理解。然后在最后整合程序的时候也遇到了一些问题，在和组员不断讨论调试以及问过导师后最终程序得以正常运行。MATLAB的运用也不算很好，因为还没有在学校里系统的学过。借鉴了一些博客上的代码，看懂了这些代码加以改动并且写上批注后加入我们的程序中。

对于哈希算法的内容，问题仍然存在。局部敏感哈希的算法思想和逻辑已经可以理解，但是由于自身对于哈希算法的数学理论仍没有很好的理解，所以还是无法靠自己写出哈希的代码。将在剩余的暑假时间里好好把哈希算法理解一下。

最后两天小组内部也遇到了问题，两位组员因为家里事情和自身疾病导致缺席白天的实习工作。但是我们在私下通过不断地沟通最终也把程序按照当初设想的思路做了出来。总的来说有一些问题仍然存在并且需要解决，但整体上我们还是克服了期间出现的一些小问题并且完成了程序设计。