

大作业的一些思路

首先介绍一下识别 ID 的一些思路：

图片的 ID 需要根据宠物的图像特征来进行识别，主要思路如下：

1. 首先需要确定宠物在图片中的大概范围。每张图片的像素大小是不一样的，但宠物出现的位置大概是固定的，大家可以先计算出图像的尺寸，然后按比例截取出包含宠物的那一块区域（例如取上 20%，下 50%，左 20%，右 20%），如下图所示：



2. 对截取的图像提取特征，这里主要介绍一下如何用特征描述算子（如 SIFT）和词袋算法（Bag-of-words）来提取特征。
 - (1) 对 train 文件里面的每张图片，首先需要检测出关键点，并对关键点进行描述。关键点可以认为是角点等位置比较特殊的点。Matlab 里面提供了一步到位的函数，参考以下链接：
http://cn.mathworks.com/help/vision/local-feature-extraction.html?searchHighlight=Local%20Feature%20Extraction&tid=doc_srchtile
SURF, Harris 等都可以，大家自己试试哪种特征效果比较好。代码很简单，根据上面提供的例子写就行。
 - (2) 在对 train 上面的每个图片检测并描述了关键点之后，需要对这些关键点做聚类并提取特征，这一块网上有很多资料，大家可以自己看看，搜“词袋”或者“Bag of words”都可以。我简单说下步骤：
<1> 对 train 图片的关键点做聚类：此时每个关键点都被描述为一个向量（例如 SIFT 是 128 维向量），假设有 10000 个关键点，那就是对这 10000 个向量做聚类。当 train 图片的关键点数量很多的时候，大家可以随机抽取一些关键点做聚类，不然运算速度会很慢。聚类时需要设定聚类中心的数量，一般几百个聚类中心就可以了。Matlab 有现成的 K-means 函数可以用。

<2> 获取聚类中心后，对 train 图像中的每张图片，计算这个图片里面所有关键点离的最近的聚类中心点，然后统计频数。假设有 100 个聚类中心点，其中有 20 个关键点属于第一个聚类中心点，15 个属于第二个中心点，这样从第 1 到第 100，每个中心点都会有个关键点频数，组合起来就是这个图像的特征。可以看出来，聚类中心点的个数就是图像特征的维度。

<3> 计算出来之后，需要进行归一化，即 TF-IDF，网上有教程，大家自己看。这一步可以有效的提高特征提取和 ID 识别的精度。

<4> 保存聚类中心点和 TF-IDF 参数，这些在 val 或 test 图片中要用到。

3. 对 val 中的每张图片，也要检测关键点，然后按上面的步骤来提取特征。注意这里所用到的聚类中心点是 train 数据的中心点，TF-IDF 参数也是 train 数据的参数。
4. 当 train 和 val 中的图片都提取了特征之后，现在就可以进行 ID 的识别了。用的方法叫 KNN，对于 val 中每张图片的特征，对 train 所有图片的特征向量计算他们的欧氏距离（也可以是其他类型的距离）。找出最接近的一张或多张图片，从而根据这些最接近图片的 ID，来判断 val 图片的 ID。参考这个函数 knnsearch:

<http://cn.mathworks.com/help/stats/knnsearch.html>

此外，除了基于 SIFT 和 BoW 的特征，大家也可以用颜色特征来做，就是颜色直方图之类的，因为这个数据集里面，颜色可以有效的区分不同的宠物。参考这个文档：<http://cn.mathworks.com/help/images/ref/imhist.html>
大家可以把获取的颜色特征和 BoW 特征拼接起来，这样能提高检测的精度。

对于 CP, HP 和 POWER UP 的一些思路:

这三个参数在图片出现的位置也比较固定，大家也可以像识别 ID 那样，大概的确定出来范围。然后可以直接把彩色图片变成灰度图或者二值图片，这里面彩色信息用处不大。然后需要精确确定数字的位置和范围，例如可以先定位 CP、HP 这些字母，从而更精确的确定数字所在的位置。然后把数字包含的小图片截取出来，如下所示：



然后对这些图片进行识别，可以用模板匹配的方法，就是自己做 0-9 的图片模板，然后进行匹配。或者找一些训练集，自己训练一个分类器（神经网络，SVM 等），用分类器来进行数字的分类和识别。

对于 level 和 cir_center 的一些思路:

这一问题需要检测出圆弧，可以采用 hough 变换来检测圆弧，大家可以自己在网上搜下解决方案，有很多例子。可以参考这个函数：

<http://cn.mathworks.com/help/images/ref/imfindcircles.html>