基于图像处理的狗识别分类系统

**摘要**

不同品种的狗有不同的特点，与之相处或饲养时需要区别对待，不能一概而论，基于图像处理的狗识别分类系统，通过对带有狗的图片进行处理，分析，然后得出图片上的狗的品种，从而可以帮助人们识别、了解身边的狗相处；此系统通过opencv实现，首先利用收集到的狗图片建立了包括藏獒、杜宾犬和阿拉斯加等品种在内的狗图片数据库，用于和输入的狗图片进行匹配；系统判断图片上的狗是何种狗的的主要过程是，首先使用交互式的基于Grab Cuts的图割算法先对图片进行预处理，将整张图片中的狗的部分抠出来，接着使用分块颜色矩索引算法提取图片中狗的颜色特征，使用高斯滤波、canny算子、基于形状比例和Hu矩算法提取图片狗的轮廓特征，使用融合LBP和GLCM的纹理特征提取方法提取图片狗的纹理特征，使用SIFT+BBF算法提取图片狗的局部特征，然后将提取到的图像特征与已建立的狗的品种数据库中的狗类的特征进行匹配，根据资料和先验知识，通过调整算法中不同参数的权值和设置合理的阈值来提高匹配的准确度，对图片中的狗进行类别判断，给出判断结果。

**关键词：**算法、特征、匹配、判断。

**目录**

1.背景分析............................................................................................1

2.狗脸识别分类系统的目标................................................................1

3.系统结构及工作流程........................................................................1

4.实现过程............................................................................................1

4.1图像预处理.....................................................................................1

4.2提取颜色特征.................................................................................7

4.3提取形状特征................................................................................10

4.4提取纹理特征................................................................................12

4.5提取局部特征................................................................................15

4.6分析判断........................................................................................23

1. 结论...................................................................................................24
2. 误差分析...........................................................................................24
3. 系统使用说明书...............................................................................25
4. **背景分析**

目前，包括流浪狗、宠物狗等在内的不同用途的与人接触较多的狗，常见的有100多种，但是每种狗都有不同的身理和性格特点，例如，阿拉斯加容易拉肚子，自律性较差，拉布拉多则是性格温和、没有攻击性、智商高等，人们需要区别对待，才能更好的与之相处，但是大多数人往往无法直接通过外形判断狗的种类，按照不同的狗的特点对其区别对待更是无从谈起，譬如流浪狗收容所一般收容的狗有几十上百条之多，狗的饲养员很有可能不能给器分类，因此在饮食上也就无法区别对待，狗由于长期饮食不善，可能造成患病等；因此我们系统基于图片处理的知识，实现一个狗识别分类系统，通过图片处理分析的方式，帮助人们判断狗的种类。

1. **狗识别分类系统的功能**

狗识别分类系统的功能是，通过对狗的图片进行处理，通过从图片上提取狗的诸多外在特征，如颜色、纹理、轮廓等，通过将这些特征和后台数据库中根据资料和先验知识建立的不同品种狗的特征相匹配，分析判断，识别图片上狗的种类，从而帮助人们快速识别图片上的狗的种类。

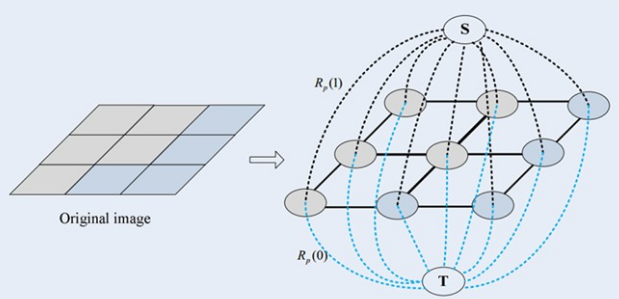
1. **系统结构及工作流程**

基于图像处理的狗识别分类系统由图片预处理机制，目标物特征提取机制，分析判断和误差分析机制三部分组成；其工作流程为：首先获得要分类的狗的图片，然后将其输入狗识别系统，系统通过对图片进行处理分析得出结论，并进行误差判断，为人们提供更加客观可靠的数据结果。

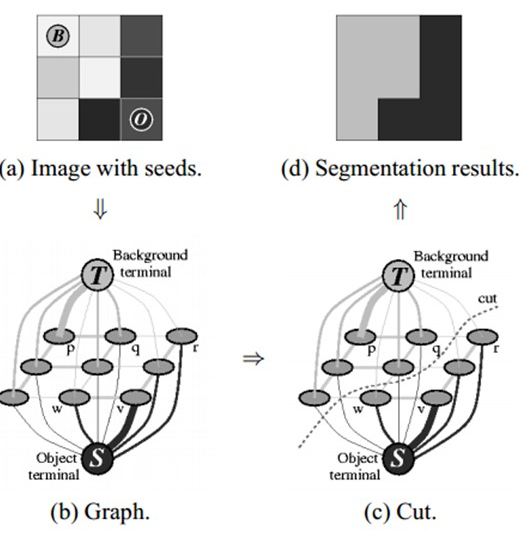
**4.实现过程**

**4.1图像预处理**

图像预处理使用grab cut算法，此算法主要功能是分割和抠图。它是graph cut算法的改进版本，与graph cut算法一样，具有少量的交互性，即用户需要简单标识图片上需要抠去的部分。首先介绍下graph cut算法，此算法将图像的图像分割问题与图的最小割问题相关联。将要分割的图像表示为一个带权无向图（每个像素对应图中一个顶点，权值非负），与普通的带权无向图不同，此图在普通图的基础上多了两个顶点“S”和“T”（称为终端节点），其他所有普通顶点都必须和这两个顶点相连形成边集合中的一部分，易知，此图有两种顶点，普通顶点和终端顶点，同时也有两种边，普通顶点之间相连的边（称为n-link），普通顶点与终端顶点之间相连的边（称为t-link）。如下图所示：



算法的主要目标是在图中寻找一个最小割集合cut，此集合是指元素为边的权值，且元素之和最小，当其包含的所有边断开时，图的所有顶点被划分为两个不相交的子集S和T，这两个子集就对应于图像的前景像素集和背景像素集，那就相当于完成了图像分割。如图所示：



Grab cut算法相比graph算法做了如下改进：

（1）Graph Cut的目标和背景的模型是灰度直方图，Grab Cut改进为RGB三通道的混合高斯模型GMM；

（2）Graph Cut的图像分割是一次达到的，而Grab Cut改进为一个不断进行分割估计和模型参数学习的交互迭代过程；

（3）Graph Cut需要用户指定目标和背景的一些种子点，但是Grab Cut只需要提供背景区域的像素集就可以了。也就是说你只需要框选目标，那么在方框外的像素全部当成背景，这时候就可以对GMM进行建模和完成良好的分割了。如图所示：



Grab cut算法实现步骤：

1. 建立颜色模型。采用RGB颜色空间，分别用一个具有K个高斯分量（一取般K=5）的全协方差GMM（混合高斯模型）来对目标和背景进行建模。于是就存在一个额外的向量**k** = {k1, . . ., kn, . . ., kN}，其中kn就是第n个像素对应于哪个高斯分量，kn∈ {1, . . . K}。对于每个像素，要么来自于目标GMM的某个高斯分量，要么就来自于背景GMM的某个高斯分量。

所以用于整个图像的Gibbs能量为：



其中，U称作区域项（所谓区域项就是与t-link边相关的项）；

其中，GMM的参数**θ**有三个：每个高斯分量的权重π、每个高斯分量的均值向量**u**（因为有RGB三个通道，故为三个元素向量）和协方差矩阵**∑**（因为有RGB三个通道，故为3x3矩阵）。也就是说描述目标的GMM和描述背景的GMM的这三个参数都需要学习确定。一旦确定了这三个参数，那么我们知道一个像素的RGB颜色值之后，就可以代入目标的GMM和背景的GMM，就可以得到该像素分别属于目标和背景的概率了，也就是Gibbs能量的区域能量项就可以确定了，即图的t-link的权值我们就可以求出。

其中，V称作边界项（所谓边界项就是与n-link边相关的项）：



如果两邻域像素差别很小，那么它属于同一个目标或者同一背景的可能性就很大，如果他们的差别很大，那说明这两个像素很有可能处于目标和背景的边缘部分，则被分割开的可能性比较大，所以当两邻域像素差别越大，能量越小。而在RGB空间中，衡量两像素的相似性，我们采用欧式距离（二范数）。这里面的参数β由图像的对比度决定，可以想象，如果图像的对比度较低，也就是说本身有差别的像素m和n，它们的差||zm-zn||还是比较低，那么我们需要乘以一个比较大的β来放大这种差别，而对于对比度高的图像，那么也许本身属于同一目标的像素m和n的差||zm-zn||还是比较高，那么我们就需要乘以一个比较小的β来缩小这种差别，使得V项能在对比度高或者低的情况下都可以正常工作。常数γ为50（经验参数）。那这时候，n-link的权值就可以通过上式来确定了，这时候我们想要的图就可以得到了，我们就可以对其进行分割了。

1. 初始化

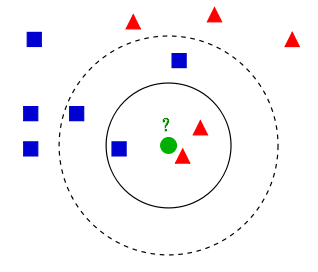
①通过直接框选目标来得到一个初始的trimap T，即方框外的像素全部作为背景像素TB，而方框内TU的像素全部作为“可能是目标”的像素。

②对TB内的每一像素n，初始化像素n的标签αn=0，即为背景像素；而对TU内的每个像素n，初始化像素n的标签αn=1，即作为“可能是目标”的像素。

③经过上面两个步骤，我们就可以分别得到属于目标（αn=1）的一些像素，剩下的为属于背景（αn=0）的像素，这时候，我们就可以通过这个像素来估计目标和背景的GMM了。我们可以通过**k-mean算法**分别把属于目标和背景的像素聚类为K类，即GMM中的K个高斯模型，这时候GMM中每个高斯模型就具有了一些像素样本集，这时候它的参数均值和协方差就可以通过他们的RGB值估计得到，而该高斯分量的权值可以通过属于该高斯分量的像素个数与总的像素个数的比值来确定。

k-mean算法简介:

K近邻算法，即是给定一个训练数据集，对新的输入实例，在训练数据集中找到与该实例最邻近的K个实例（也就是上面所说的K个邻居），这K个实例的多数属于某个类，就把该输入实例分类到这个类中。如图：



1. 迭代最小化

①对每个像素分配GMM中的高斯分量（例如像素n是目标像素，那么把像素n的RGB值代入目标GMM中的每一个高斯分量中，概率最大的那个就是最有可能生成n的，也即像素n的第kn个高斯分量）：

IMG_256

②对于给定的图像数据**Z**，学习优化GMM的参数（因为在步骤（1）中我们已经为每个像素归为哪个高斯分量做了归类，那么每个高斯模型就具有了一些像素样本集，这时候它的参数均值和协方差就可以通过这些像素样本的RGB值估计得到，而该高斯分量的权值可以通过属于该高斯分量的像素个数与总的像素个数的比值来确定。）：

IMG_257

③分割估计（通过1中分析的Gibbs能量项，建立一个图，并求出权值t-link和n-link，然后通过max flow/min cut算法来进行分割）：

IMG_258

④重复步骤（1）到（3），直到收敛。经过（3）的分割后，每个像素属于目标GMM还是背景GMM就变了，所以每个像素的kn就变了，故GMM也变了，所以每次的迭代会交互地优化GMM模型和分割结果。另外，因为步骤（1）到（3）的过程都是能量递减的过程，所以可以保证迭代过程会收敛。

⑤采用border matting对分割的边界进行平滑等等后期处理。

**4.2提取颜色特征**

提取图像颜色特征，使用的是分块颜色矩索引算法，主要功能是图取图片中狗的颜色特征。其实现过程如下：

1. 颜色空间的选择。数字图像处理中常用的模型是RGB（红，绿，蓝）模型和HSV（色调，饱和度，亮度）模型，由于HSV的彩色描述对人来说是自然且非常直观的，且更符合人描述和解释颜色的方式，所以我们选择使用HSV模型。

①HSV模型。HSV模型中颜色的参数分别是：色调（H：hue），饱和度（S：saturation），明度（V：value）。色调（H：hue）：用角度度量，取值范围为0°～360°，从红色开始按逆时针方向计算，红色为0°，绿色为120°,蓝色为240°。它们的补色是：黄色为60°，青色为180°,品红为300°；饱和度（S：saturation）：取值范围为0.0～1.0，值越大，颜色越饱和；明度（V：value）：取值范围为0(黑色)～255(白色)。

②RGB转HSV。设 (r, g, b) 分别是一个颜色的红、绿和蓝坐标，它们的值是在 0 到 1 之间的实数。设 max 等价于 r, g 和 b 中的最大者。设 min 等于这些值中的最小者。要找到在 HSV 空间中的 (h, s, v) 值，这里的 h ∈ [0, 360）是角度的色相角，而 s, v ∈ [0,1] 是饱和度和亮度，计算公式为：

max=max(R,G,B)

min=min(R,G,B)

if R = max, H = (G-B)/(max-min)

if G = max, H = 2 + (B-R)/(max-min)

if B = max, H = 4 + (R-G)/(max-min)

H = H \* 60

if H < 0, H = H + 360

V=max(R,G,B)

S=(max-min)/max

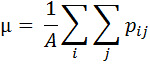
③实现。OpenCV下有个函数（cvtColor）可以直接将RGB模型转换为HSV模型，注意的是OpenCV中H∈ [0, 180）， S ∈ [0, 255]， V ∈ [0, 255]。我们知道H分量基本能表示一个物体的颜色，但是S和V的取值也要在一定范围内，因为S代表的是H所表示的那个颜色和白色的混合程度，也就说S越小，颜色越发白，也就是越浅；V代表的是H所表示的那个颜色和黑色的混合程度，也就说V越小，颜色越发黑。经过实验，识别蓝色的取值是 H在100到140，S和V都在90到255之间。

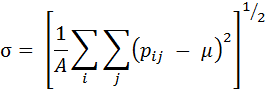
（2）分量量化。为了减少计算量，分别对 H，S，V，三个分量进行量化，即把色调分为16份，饱和度分为4份，明度分为4份。再将三维 H， S， V转化为一维分量，用公式表示为：

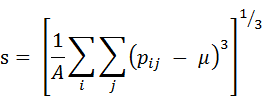
L = 16H +4S + V

色调、 饱和度、 明度的权值分别设为16，4和1，突出了人类视觉对色调敏感的特点，减小了图像明度和饱和度对检索结果的影响。

（3）提取颜色矩特征及测量。颜色矩方法的思想在于图像中任何的颜色分布均可以用它的矩来表示。此外，由于颜色分布信息主要集中在低阶矩中，因此仅采用颜色的一阶矩、二阶矩和三阶矩就可以表达图像的颜色分布。一阶中心矩、二阶中心矩和三阶中心矩分别表示图像或子区域图像的平均颜色、标准方差和三次根非对称性，其表示式分别为：







其中A代表比较区域的像素总数（图像面积Aera），Pij代表图像空间二维坐标（i，j）处合成后的HSV像素值。

这样，图像就能用颜色矩特征向量（μ，δ，s）表征，两幅图像的相似度用对应的颜色分布特征矩向量间的加权欧氏距离D（Q，I）来度量：

D（Q，I）= w1| μq - μi | + w2| δq - δi | + w3| sq - si |

其中w1,w2,w3分别表示均值特征、标准方差特征、三次根非对称性特征的权值。

这样，欲识别图像和图像库中的图像都有D，当D越小时，两图像的颜色差别越小，两图像越近似。

**4.3提取轮廓特征**

提取图像轮廓特征，主要使用的是基于形状比例和Hu矩算法，主要功能是提取图片中的狗的轮廓特征。

（1）提取轮廓。先将图片转化为二值图像并保存为灰度图，再对图像进行高斯滤波和canny边缘检测算子，提取出图像的轮廓。

①高斯滤波算法原理：

高斯滤波是对整幅图像进行加权平均的过程，每一个像素点的值，都由其本身和邻域内的其他像素值经过加权平均后得到。高斯滤波的具体操作是：用一个模板（或称卷积、掩模）扫描图像中的每一个像素，用模板确定的邻域内像素的加权平均灰度值去替代模板中心像素点的值用。高斯平滑滤波器对于抑制服从正态分布的噪声非常有效。

②Canny边缘检测算子实现步骤：

1.生成高斯滤波系数；

2.用生成的高斯滤波系数对原图像进行平滑；

3.求滤波后图像的梯度；

4.进行非最大抑制；

5.统计图像的直方图，对阈值进行判定；

6.利用函数寻找边界起点；

7.根据第6步执行的结果，从一个像素点开始搜索，搜索以该像素点为边界 起点的一条边界的一条边界的所有边界点。

（2）轮廓的量化。再进行完canny边缘检测后，使用findContours函数将轮廓信息保存在向量中，向量内每个元素保存了一组由连续的Point点构成的点的集合的向量。

void findContours(InputOutputArray image, OutputArrayOfArrays contours, OutputArray hierarchy, int mode, int method, Point offset=Point())

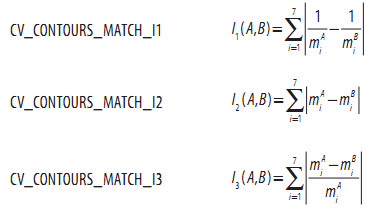
其中，该函数的各个参数：

输入图像image必须为一个2值单通道图像；contours参数为检测的轮廓数组；hiararchy参数和轮廓个数相同，每个轮廓contours[ i ]对应4个hierarchy元素hierarchy[ i ][ 0 ] ~hierarchy[ i ][ 3 ]，分别表示后一个轮廓、前一个轮廓、父轮廓、内嵌轮廓的索引编号，如果没有对应项，该值设置为负数；mode表示轮廓的检索模式或近似方法；offset表示代表轮廓点的偏移量，可以设置为任意值。对ROI图像中找出的轮廓，并要在整个图像中进行分析时，这个参数还是很有用的。

（3）轮廓的匹配。使用Hu矩进行匹配。cvMatchShapes是输入图像直接进行hu矩匹配的函数，返回的是两个图像或轮廓之间hu矩的相似度，当返回的值越接近0，表示两幅图像的形状越近似。

double cvMatchShapes(const void\* object1,const void\* object2,int method,double parameter = 0);

其中，method表示匹配的方法，共有三种：



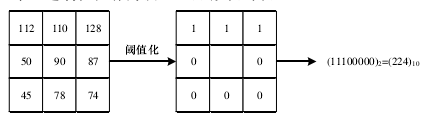
这样，可以得到图像轮廓进行Hu矩匹配后的值，值越小两幅图像越接近。并且，为了提高识别率，我们还加入了身形比例信息进行对比，即对输入的图像进行映射，提取出图像中狗身体的长度和高度的比例。再求出该比例和图像库中的身形比例的欧氏距离。这样再将该欧式距离和上一步中提取出的Hu矩匹配值做权值相加，得到最终的匹配值。通过该匹配值与图像进行相似匹配，最终可以得到与输入图像近似的图像。

**4.4提取纹理特征**

提取图片纹理特征，使用融合LBP和GLCM的纹理特征提取方法，主要功能是提取图片中狗的纹理特征。其具体操作如下：

（1）LBP 算子 。LBP 是一种基于灰度描述图像纹理特征的不相关算子，它通过对图像任意一点与其周围点的灰度值大小关系来表征图像的局部纹理特征。

基本的 LBP 算子是一个固定大小为 3×3 的矩形块，此矩形中的一个中心子块和 8 个邻近子块对应于 9 个灰度值。基本 LBP 算子的计算过程如图 1 所示，具体步骤如下：将四周的 8 个灰度值与中心灰度值相比较，大于中心灰度值的子块用 1 表示，反之，则用 0 表示。然后根据顺时针方向读出8 个二进制值，作为该 3×3 矩形块的纹理。 基本的LBP算子计算过程：



基本 LBP 算子只覆盖了一个固定范围内的小区域，而且当图像发生旋转时，二进制值的顺序会随之发生变化，这显然不能满足不同尺度的纹理特征和旋转不变性的要求。因此，文献[5]对 LBP 算子进行了发展和改进，将 3×3 邻域扩展到任意邻域，并用圆形邻域代替了矩形邻域，允许在半径为 R 的圆形邻域内有任意多个像素点，通常记作 LBPP,R，表示在半径为 R 的圆形邻域内有 P 个像素点。常见的 LBP算子有 LBP8,1、LBP16,2和 LBP24,3。此外，还可使用旋转不变的 LBP 算子，即不断旋转圆形邻域得到一系列初始定义的 LBP 值，取其最小值作为该邻域的 LBP 值，计算公式如下：



其中，表示旋转不变的LBP算子；函数为转函数，表示将 循环右移 i 位。

图像的每个像素经过LBP运算后得到的是一幅和原图像同大小的新图像，称为 LBP 图像。一般都不将 LBP 图像作为描述纹理的特征参数，而是采用 LBP 图像的统计直方图，但这样会导致描述纹理的特征参数过多，给特征参数计算带来了很大困难。

（2）灰度共生矩阵及其 Haralick 特征。灰度共生矩阵描述了图像中满足一定距离和一定方向的 2 点灰度出现的概率，定义为：灰度为 i 和 j 的一对像素点位置方向为θ，距离为 d 的概率记为 P (i , j , d , θ)。 一般θ有4个取值：0度, 45度, 90度和135度。像素对 I (k , l )=i和 I ( m, n )=j在这 4 个方向上出现的概率分别为：









其中，# 表示在该集合中的元素数目。

本文将对比度、相关性、能量、逆差矩作为特征，分别

记为 f1, f2, f3, f4，计算公式如下：







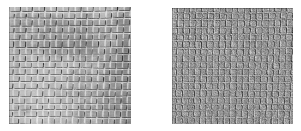


其中，；；；。

（3）纹理特征提取方法 。

①选取旋转不变的算子（其中，P=8，R=1），对图像的每个像素经过LBP 运算，得到 LBP 图像。原始纹理图像如下图左所示，LBP

图像如下图右所示：



②计算 LBP 特征谱的灰度共生矩阵，选取距离 d =1，θ = 0度, 45度,90度,135度，得到 4 个灰度共生矩阵。

③计算各个灰度共生矩阵的对比度、相关性、能量、逆差矩，并将求得的每个灰度共生矩阵的 4 个特征值级联起来作为图像的纹理特征，共有 16 个纹理特征。如此，表征纹理的特征参数为 16 个，与采用LBP 直方图相比，特征维数得到大幅降低。

**4.5提取局部特征**

提取图片的局部特征，使用的是SIFT算法，主要功能是提取图片中狗的的局部性特征。SIFT算法的实质是在不同的尺度空间上查找关键点(特征点)，并计算出关键点的方向。SIFT所查找到的关键点是一些十分突出，不会因光照，仿射变换和噪音等因素而变化的点，如角点、边缘点、暗区的亮点及亮区的暗点等。

SIFT算法分解为如下四步：

1. 尺度空间极值检测：搜索所有尺度上的图像位置。通过高斯微分函数来识别潜在的对于尺度和旋转不变的兴趣点。

①尺度空间理论

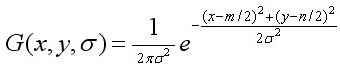
尺度空间理论的基本思想是：在图像信息处理模型中引入一个被视为尺度的参数，通过连续变化尺度参数获得多尺度下的尺度空间表示序列，对这些序列进行尺度空间主轮廓的提取，并以该主轮廓作为一种特征向量，实现边缘、角点检测和不同分辨率上的特征提取等。

②尺度空间的表示

一个图像的尺度空间，L(x,y,σ)定义为一个变化尺度的高斯函数G(x,y,σ)与原图像I(x,y) 的卷积：

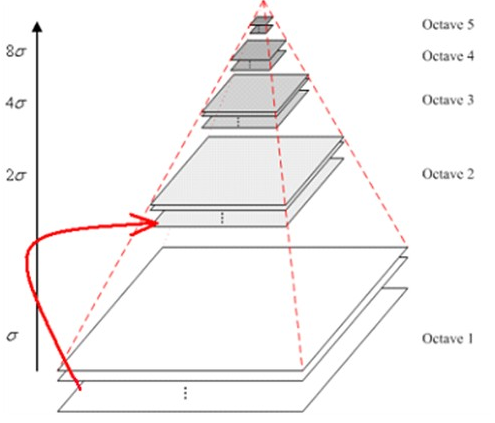
L(x,y,σ) = G(x,y,σ)\*I(x,y)  其中，\*表示卷积运算

高斯函数：



### ③高斯金字塔的构建

### 尺度空间在实现时使用高斯金字塔表示，高斯金字塔的构建分为两部分：对图像做不同尺度的高斯模糊和对图像做降采样(隔点采样)。



图像的金字塔模型是指，将原始图像不断降阶采样，得到一系列大小不一的图像，由大到小，从下到上构成的塔状模型。原图像为金子塔的第一层，每次降采样所得到的新图像为金字塔的一层(每层一张图像)，每个金字塔共n层。金字塔的层数根据图像的原始大小和塔顶图像的大小共同决定，其计算公式如下：

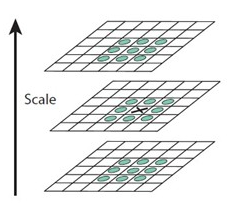


④高斯差分金字塔

由于尺度归一化的高斯拉普拉斯函数的极大值和极小值同其它的特征提取函数相比，能够产生最稳定的图像特征，而高斯差分函数（简称DOG算子）与尺度归一化的高斯拉普拉斯函数非常相似，所示使用更高效的高斯差分算子代替拉普拉斯算子进行极值检测，在实际计算时，使用高斯金字塔每组中相邻上下两层图像相减，得到高斯差分图像。

⑤空间极值点监测

关键点是由DOG空间的局部极值点组成的，关键点的初步探查是通过同一组内各DoG相邻两层图像之间比较完成的。为了寻找DoG函数的极值点，每一个像素点要和它所有的相邻点比较，看其是否比它的图像域和尺度域的相邻点大或者小。如图所示，中间的检测点和它同尺度的8个相邻点和上下相邻尺度对应的9×2个点共26个点比较，以确保在尺度空间和二维图像空间都检测到极值点。



由于要在相邻尺度进行比较，如图右侧每组含4层的高斯差分金子塔，只能在中间两层中进行两个尺度的极值点检测，其它尺度则只能在不同组中进行。为了在每组中检测S个尺度的极值点，则DOG金字塔每组需S+2层图像，而DOG金字塔由高斯金字塔相邻两层相减得到，则高斯金字塔每组需S+3层图像，实际计算时S在3到5之间。当然这样产生的极值点并不全都是稳定的特征点，因为某些极值点响应较弱，而且DOG算子会产生较强的边缘响应。

⑥确定构建尺度空间的参数

δ—尺度空间坐标

 O—组(octave)数

 S— 组内层数

在上述尺度空间中，O和S，δ的关系如下：

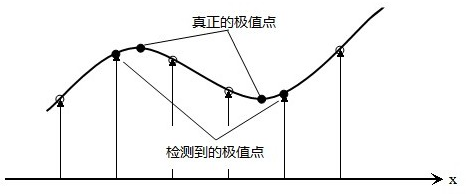


其中δ0是基准层尺度，o为组octave的索引，s为组内层的索引，在最开始建立高斯金字塔时，要预先模糊输入图像来作为第0个组的第0层的图像，这时相当于丢弃了最高的空域的采样率。因此通常的做法是先将图像的尺度扩大一倍来生成第-1组。

（2）关键点定位。在每个候选的位置上，通过一个拟合精细的模型来确定位置和尺度。关键点的选择依据于它们的稳定程度。

①关键点的精确定位

离散空间的极值点并不是真正的极值点，下图显示了二维函数离散空间得到的极值点与连续空间极值点的差别。利用已知的离散空间点插值得到的连续空间极值点的方法叫做子像素插值。



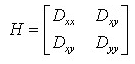
为了提高关键点的稳定性，需要对尺度空间DoG函数进行曲线拟合。利用DoG函数在尺度空间的Taylor展开式(拟合函数)为：



其中，，求导并让方程等于零，可以得到极值点的偏移量，当它在任一维度上的偏移量大于0.5时，意味着插值中心已经偏移到它的邻近点上，所以必须改变当前关键点的位置。同时在新的位置上反复插值直到收敛；也有可能超出所设定的迭代次数或者超出图像边界的范围，此时这样的点应该删除。

②消除边缘影响

DOG算子会产生较强的边缘响应，需要剔除不稳定的边缘响应点。获取特征点处的Hessian矩阵，主曲率通过一个2x2 的Hessian矩阵H求出：



H的特征值α和β代表x和y方向的梯度，



表示矩阵H对角线元素之和，表示矩阵H的行列式。假设是α较大的特征值，而是β较小的特征值，令α=rβ，则



导数由采样点相邻差估计得到,

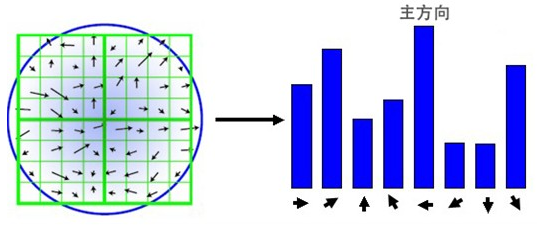
为了检测主曲率是否在某域值r下，只需检测

是否成立，成立时将关键点保留，反之剔除。

（3）方向确定：基于图像局部的梯度方向，分配给每个关键点位置一个或多个方向。所有后面的对图像数据的操作都相对于关键点的方向、尺度和位置进行变换，从而提供对于这些变换的不变性。为了使描述符具有旋转不变性，需要利用图像的局部特征为给每一个关键点分配一个基准方向。使用图像梯度的方法求取局部结构的稳定方向。对于在DOG金字塔中检测出的关键点点，采集其所在高斯金字塔图像3σ邻域窗口内像素的梯度和方向分布特征。梯度的模值和方向如下：



在完成关键点的梯度计算后，使用直方图统计邻域内像素的梯度和方向。梯度直方图将0~360度的方向范围分为36个柱(bins)，其中每柱10度。如图5.1所示，直方图的峰值方向代表了关键点的主方向，(为简化，图中只画了八个方向的直方图)。



方向直方图的峰值则代表了该特征点处邻域梯度的方向，以直方图中最大值作为该关键点的主方向。为了增强匹配的鲁棒性，只保留峰值大于主方向峰值80％的方向作为该关键点的辅方向。因此，对于同一梯度值的多个峰值的关键点位置，在相同位置和尺度将会有多个关键点被创建但方向不同。仅有15％的关键点被赋予多个方向，但可以明显的提高关键点匹配的稳定性。实际编程实现中，就是把该关键点复制成多份关键点，并将方向值分别赋给这些复制后的关键点，并且，离散的梯度方向直方图要进行插值拟合处理，来求得更精确的方向角度值。

至此，将检测出的含有位置、尺度和方向的关键点即是该图像的SIFT特征点。

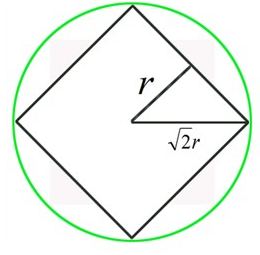
（4）关键点描述。在每个关键点周围的邻域内，在选定的尺度上测量图像局部的梯度。这些梯度被变换成一种表示，这种表示允许比较大的局部形状的变形和光照变化。通过以上步骤，对于每一个关键点，拥有三个信息：位置、尺度以及方向。接下来就是为每个关键点建立一个描述符，用一组向量将这个关键点描述出来，使其不随各种变化而改变，比如光照变化、视角变化等等。这个描述子不但包括关键点，也包含关键点周围对其有贡献的像素点，并且描述符应该有较高的独特性，以便于提高特征点正确匹配的概率。

SIFT描述子是关键点邻域高斯图像梯度统计结果的一种表示。通过对关键点周围图像区域分块，计算块内梯度直方图，生成具有独特性的向量，这个向量是该区域图像信息的一种抽象，具有唯一性。

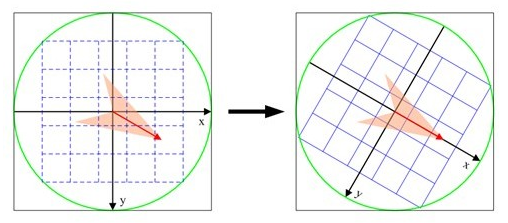
描述子使用在关键点尺度空间内4\*4的窗口中计算的8个方向的梯度信息，4\*4\*8=128维向量表征。表示步骤如下：

① 确定计算描述子所需的图像区域，实际计算所需的图像区域半径为：

其中，计算结果四舍五入取整



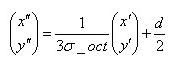
②将坐标轴旋转为关键点的方向，以确保旋转不变性。



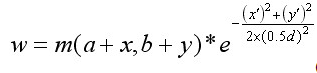
旋转后邻域内采样点的新坐标为：



③将邻域内的采样点分配到对应的子区域内，将子区域内的梯度值分配到8个方向上，计算其权值。旋转后的采样点坐标在半径为radius的圆内被分配到d×d的子区域，计算影响子区域的采样点的梯度和方向，分配到8个方向上。旋转后的采样点（x’，y’）落在子区域的下标为：

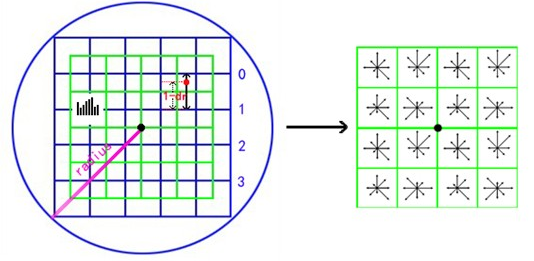


子区域的像素的梯度大小按δ=0.5d的高斯加权计算：



其中a，b为关键点在高斯金字塔图像中的位置坐标。

（4） 插值计算每个种子点八个方向的梯度。

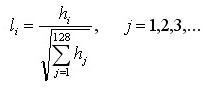


同理，对邻近两列的贡献因子为dc和1-dc，对邻近两个方向的贡献因子为do和1-do。则最终累加在每个方向上的梯度大小为：



其中k，m，n为0或为1。

⑤如上统计的4\*4\*8=128个梯度信息即为该关键点的特征向量。特征向量形成后，为了去除光照变化的影响，需要对它们进行归一化处理，对于图像灰度值整体漂移，图像各点的梯度是邻域像素相减得到，所以也能去除。得到的描述子向量为归一化后的特征向量为则



⑥描述子向量门限。非线性光照，相机饱和度变化对造成某些方向的梯度值过大，而对方向的影响微弱。因此设置门限值(向量归一化后，一般取0.2)截断较大的梯度值。然后，再进行一次归一化处理，提高特征的鉴别性。

⑦按特征点的尺度对特征描述向量进行排序。

至此，SIFT特征描述向量生成。

**4.6分析判断**

狗的种类识别中，先将狗从图像中分割出来，然后用颜色、形状、纹理特征进行图片的粗粒度筛选，最后使用sift特征进行最后种类的确定。分割中，1~3次交互可获得较好的前景图。筛选过程中，颜色特征1次迭代能过滤5~10张库图片，形状、纹理特征1次迭代能过滤1~3张图片，可见颜色特征具有更好的过滤能力。迭代次数选择上，颜色3~5次，形状纹理1~2次比较合适，这样既有较大的筛选力度，可以筛除60%以上的无用图片，又可以保证较低的误删率。Sift特征匹配上，特征点直接匹配会导致匹配点过多，ransac匹配法又会导致匹配点接近于0，故我们采用了劳式算法，即先对特征点进行knn分类（分成2类），然后设定距离比例的阈值进行匹配，这样具有较好的匹配效果。

1. **结论**

基于图像处理的狗种识别，分割——筛选——判断模式进行的识别具有一定的可行性。在分割阶段，基于GraphCuts的图割算法能在1~3次的交互中获得理想的前景图片；在筛选阶段，基于颜色矩索引的颜色特征，基于比例和Hu不变矩的形状特征，基于LBP和GLCM的纹理特征能够过滤60%的无用图片，同时保持较低的误删率；在判断阶段，使用sift特征和劳式算法进行的匹配能够在一定程度上判断出狗的品种，获得准确的种类信息。

1. **误差分析**

误差的主要来源来自sift特征点的匹配上，由于角度、光照、噪点、个体差异等因素的影响，同一种狗的sift特征会不尽相同，这加大了种类判断的难度；其次，不同种类的狗可能颜色相近，不同视角下的狗可能形状相近，同一种狗可能因为角度具有不同的纹理，这些因素都具有较大的混淆作用，致使筛选过程难以作用，这也是误差的来源；还有图像分割的狗具有不完整性，使得后期特征提取上产生误差，造成判断出错。

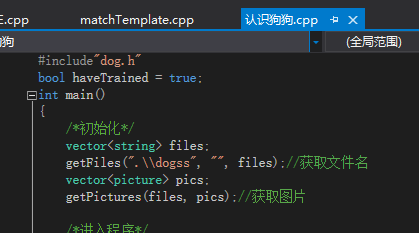
**7.系统使用说明书**

使用流程说明：训练库中图像-选取识别图片-图片处理-识别图片

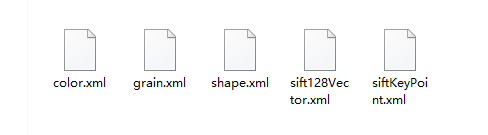
（1）训练库中图像

训练时，将 bool haveTrained = true 改为 = false ,然后执行程序即可。

下图中第7行是读取库中的图片，图片保存在dogss文件夹中，可以根据需要自行修改图片来源。



训练完毕后项目文件中会有以下文件：

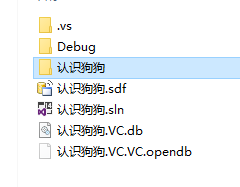


建议在训练前先删去已存在的.xml文件。

**注：**训练完毕后，务必将bool haveTrained = false 改为 = true。

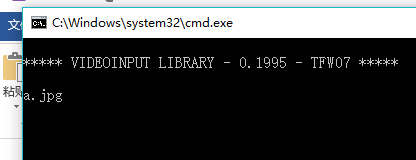
（2）选取识别图片

将要识别的图片放在该文件夹中：

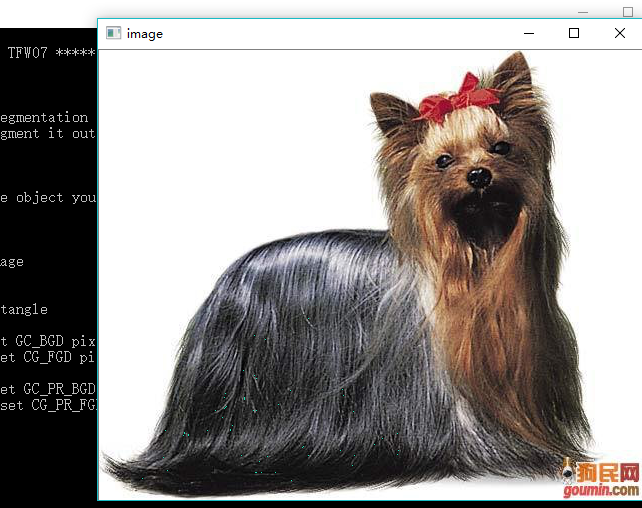


（3）对图片预处理

确定bool haveTrained = true 后，执行程序。输入要识别的图像名：



按回车会出现图像：

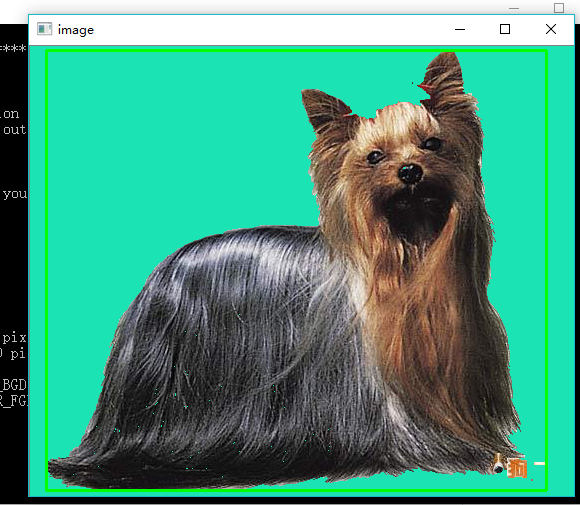


对图像处理：

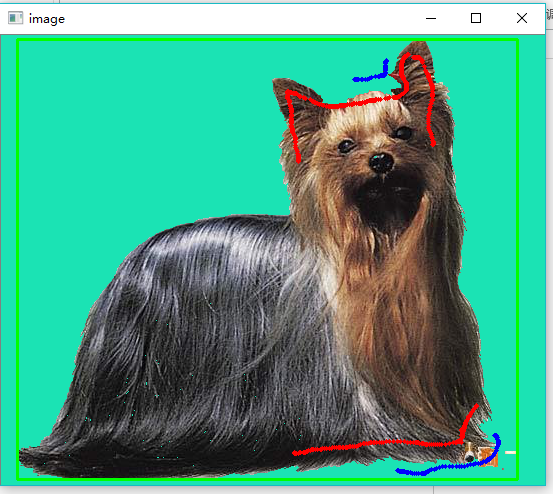
先用鼠标左键划出图像区域：



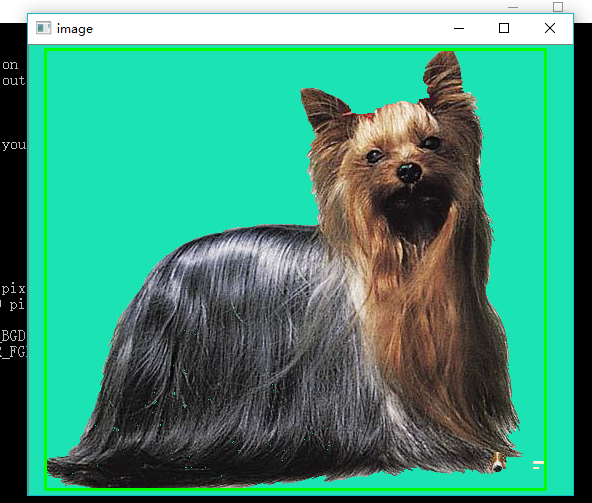
然后输入n确定，得到处理后图片：



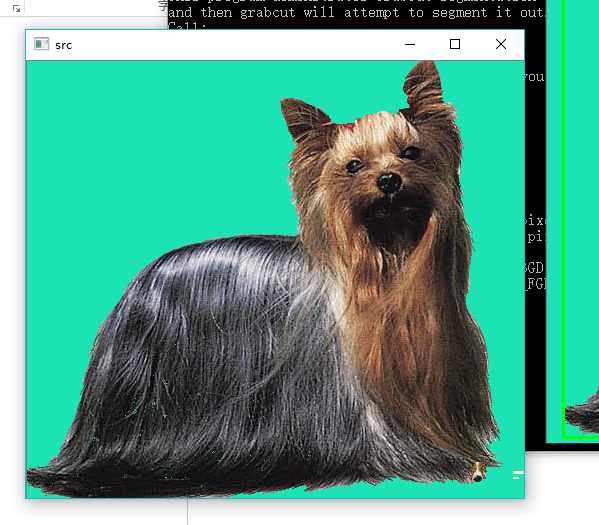
如果图片处理的不太好，可以进行进一步处理，按住Ctrl选取想要删去的内容（蓝色），按住Shift选取要留下的内容（红色）：



然后按n确定：



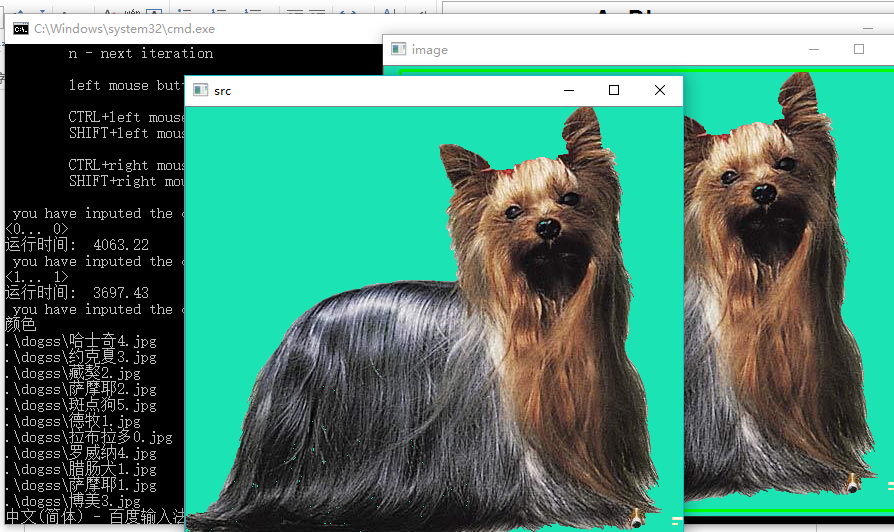
如果觉得处理得不好可以重复按上一步处理，如果处理完毕，按E键确认，会弹出名为src的图像框：



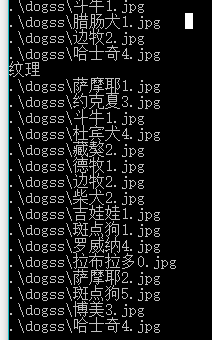
如果在以上步骤中任何一处出现由于用户失误导致图像处理得不好，可以按R回到刚开始处理时的界面。

（4）图像识别

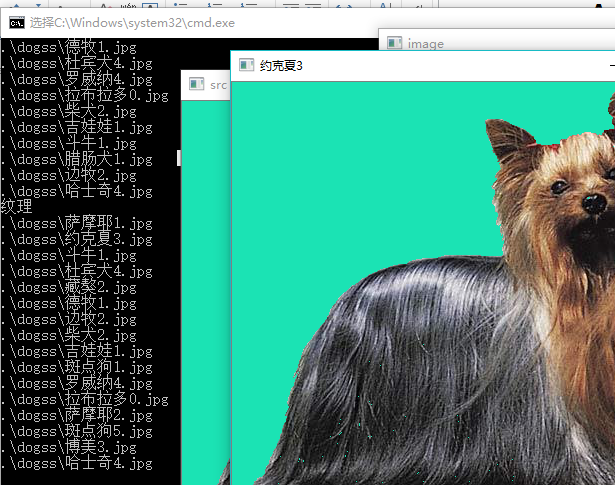
当第三步处理完毕后，在弹出src图像界面时按回车即可开始对处理好的图像进行处理：



在控制台应用程序中会显示识别的过程，其中“颜色”下方留下的是颜色识别后留下的近似图片，“形状”下方留下的是形状识别后留下的近似图片，“纹理”后留下的是纹理匹配后的图片。



注：形状匹配，是处理颜色处理后留下的图片，纹理处理得是形状处理后留下的图片。



处理完毕后，会弹出一个图像窗口，窗口名即为识别出的结果，上图中识别的结果是“约克夏”。