这种技术的原理是什么？计算机怎么知道两张图片相似呢？

根据[Neal Krawetz](http://www.hackerfactor.com/blog/index.php?/archives/432-Looks-Like-It.html)博士的解释，实现相似图片搜素的关键技术叫做"感知哈希算法"（Perceptualhash algorithm），它的作用是对每张图片生成一个"指纹"（fingerprint）字符串，然后比较不同图片的指纹。结果越接近，就说明图片越相似。

以下是一个最简单的Java实现：

预处理：读取图片

第一步，缩小尺寸。

将图片缩小到8x8的尺寸，总共64个像素。这一步的作用是去除图片的细节，只保留结构、明暗等基本信息，摒弃不同尺寸、比例带来的图片差异。

第二步，简化色彩。

将缩小后的图片，转为64级灰度。也就是说，所有像素点总共只有64种颜色。

第三步，计算平均值。

计算所有64个像素的灰度平均值。

第四步，比较像素的灰度。

将每个像素的灰度，与平均值进行比较。大于或等于平均值，记为1；小于平均值，记为0。

第五步，计算哈希值。

将上一步的比较结果，组合在一起，就构成了一个64位的整数，这就是这张图片的指纹。组合的次序并不重要，只要保证所有图片都采用同样次序就行了。

得到指纹以后，就可以对比不同的图片，看看64位中有多少位是不一样的。在理论上，这等同于计算"汉明距离"（Hammingdistance）。如果不相同的数据位不超过5，就说明两张图片很相似；如果大于10，就说明这是两张不同的图片。

你可以将几张图片放在一起，也计算出他们的汉明距离对比，就可以看看两张图片是否相似。

这种算法的优点是简单快速，不受图片大小缩放的影响，缺点是图片的内容不能变更。如果在图片上加几个文字，它就认不出来了。所以，它的最佳用途是根据缩略图，找出原图。

实际应用中，往往采用更强大的pHash算法和SIFT算法，它们能够识别图片的变形。只要变形程度不超过25%，它们就能匹配原图。这些算法虽然更复杂，但是原理与上面的简便算法是一样的，就是先将图片转化成Hash字符串，然后再进行比较。

以上内容大部分直接从阮一峰的网站上复制过来，想看原著的童鞋可以去在最上面的链接点击进去看。

提供源码下载，源码下载链接：http://download.csdn.net/detail/yjflinchong/4239243