接口设计

类图

预处理



查看模型性能



预处理

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | 接口返回类型 | 接口参数 | 功能说明 |
| concurrent | int | threadNum | 建立多个线程并发地处理收到的舌象图片，threadNum决定最大线程数,若线程建立成功返回线程数，否者返回-1. |
| getTonguePic | Int[][][] | rawPic | 对原始图片rawPic进行自动识别，分离舌图的有效部分 |
| tongueDiv | int[][][] | picture | 根据舌体划分的结果将舌体从图片背景中分割出来，将分割后的舌体图片输出。 |
| getPicInfor | int[] | picture | 根据输入的图片得出其像素信息 |
| divPixel | int[][][] | picture  x\_pix  y\_pix | 当图像中的像素数量少于规范的图片时，应对部分像素进行分割操作，最后输出标准格式的图片。 |
| mergePixel | int[][][] | picture  x\_pix  y\_pix | 当图像中的像素数量多于规范的图片时，应对部分像素进行合并操作，最后输出标准格式的图片。 |

查看模型性能参数

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | 接口返回类型 | 接口参数 | 功能说明 |
| setBiasrDem | null | bias\_dem | 改变检测模型训练的速度的误差要求 |
| getModelClassifyTimeSpent | int | null | 获取模型对测试集分类耗时 |
| getModelClassifyAccuracy | double | null | 获取模型对测试集分类准确率 |
| getModelTrainingSpeed | int | null | 获取模型训练时误差减小到误差要求迭代次数 |

算法原理设计

图片预处理

舌图提取

一般采用基于边缘检测方法完成从背景中提取 舌像。操作要点为: ( 1) 由于原始图像的舌体边缘与皮肤的亮 度值差距不大, 可以通过增强舌体边缘与周围皮肤的对比, 得 到强化对比的矩形灰阶图像; ( 2) 通过选择适当的临界值将强 化对比图像二分为黑白图像; ( 3) 通过对原始图像和黑白图像 的边缘检测找出舌体边界。

舌体分割

舌体图像分割可依赖于颜色数据和形态 数据。依赖前者方法简单, 但适应性差; 依赖后者所需技术含 量高, 但算法的适应性好。从获得的舌图可以发现, 在色彩上 脸色与舌色有明显的区别, 在空间上舌的像素是集中在一起 的, 且在舌体的边缘线上存在色差, 因此将颜色数据与形态数 据相结合, 就能更好的实现分割。具体方法如下:

Snake 算法

(1) Snake 是能量极小化的样条, 内力约 束它的形状, 外力引导它的行为, 图像力将其拖向显著的图像 特征; Snake 是“主动”的轮廓线模型, 它锁定在图像特征附近, 准确地将它们极小化[28] 。其非常适合于非刚体的分割, 但是对 初始位置敏感, 需要依赖其他机制将 Snake 放置在感兴趣的 图像特征附近。由于 Snake 模型的非凸性, 它有可能收敛到局 部极值点, 甚至发散。在传统 Snake 模型的基础上, 根据舌象 的特征和 HSV 空间的特点, 实现 active contour 自动初始化。

(2) 利用 Gabor 小波的特性, 改进 snake 的外部能量项, 解决 传统 Snake 模型中存在的一些问题, 改进后的模型可更好地 应用于舌体的分割。( 3) 再利用 Gabor Jets 求得前后两帧的偏 移量, 和前一帧的轮廓线相加, 就得到后一帧的初始轮廓线, 然后采用改进的 Snake 算法得到动态舌体较佳的分割位置。 本算法为今后研究动态信息( 如中风时舌体的震颤、吐弄等) 奠定了基础。

GVF ( Gradient Vector Flow) Snake 方法 利用力平 衡条件作为 Snake 的起始点, 定义一个新的静态外力场 GVF 场, 用该矢量场的函数代替传统 Snake 模型中的外部能量。利 用 GVF Snake 找到对象凹处的边界性质, 用其求舌象的齿痕 边( 又称齿痕轮廓线) , 再根据齿痕轮廓线上各点的曲率和梯 度的特点, 求得齿痕数, 并建立齿痕程度的评定方法。

基于舌图提取、分割与识别的有关算法 王艳清等提出了一种基于先验知识的自动舌体分割算法, 根据舌体的位 置、颜色等先验信息取得了较好的分割效果。

孙炀等应用 灰度差分统计方法, 从对比度、角度方向二阶矩、熵、平均值 4 个参数着手, 在舌象纹理的量化与舌质老嫩的判别等方面均 取得了一定效果, 并提出一种基于分裂-合并方法的区域分割 算法, 以实现对舌图的分割。

有学者提出色貌评价与三刺激值 匹配相结合的在线彩色校正方法与基于二分光反射模型的舌 苔润燥分析算法。

周越等指出一种基于颜色空间的纹理 特征来研究舌特征的途径, 利用 YCbCr 空间的色度饱和度信 息以及 2D Gabor 小波系数能量分布特征, 将舌体从原始图中 可靠分离出来。

王郁中等使用一种基于颜色纹理的无监督 图像分割方法对舌图进行初分割, 然后利用基于色度参数调 整后生成的模版进行区域匹配合并完成舌体的提取算法。

并行图片预处理

多线程（multithreading），是指从软件或者硬件上实现多个线程并发执行的技术。具有多线程能力的计算机因有硬件支持而能够在同一时间执行多于一个线程，进而提升整体处理性能。

具有这种能力的系统包括对称多处理机、多核心处理器以及芯片级多处理或同时多线程处理器。

在一个[程序](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F/71525" \t "_blank)中，这些独立运行的程序片段叫作“[线程](https://baike.baidu.com/item/%E7%BA%BF%E7%A8%8B/103101)”（Thread），利用它编程的概念就叫作“多线程处理”

实现多线程是采用一种并发执行机制[3]  。

并发执行机制原理：简单地说就是把一个处理器划分为若干个短的时间片，每个时间片依次轮流地执行处理各个应用程序，由于一个时间片很短，相对于一个应用程序来说，就好像是处理器在为自己单独服务一样，从而达到多个应用程序在同时进行的效果[3]  。

多线程就是把操作系统中的这种并发执行机制原理运用在一个程序中，把一个程序划分为若干个子任务，多个子任务并发执行，每一个任务就是一个线程。这就是多线程程序

获取图片信息

利用语言已有的库来获取图片的信息，或者可以直接通过循环计算得出图片的有效像素数量。

裁切像素

按照输入的水平，垂直方向的像素数量对一个像素进行裁切操作将其分为多个像素，最终使得图片格式与标准格式相符。裁切操作尽量在图片边缘进行而减小对分离出的舌体图像的影响。

合并像素

按照输入的水平，垂直方向的像素数量对多个像素进行合并操作将其合并为一个像素，若合并的多个像素的像素的值不一致时，合并出来的像素的值应该是被合并的多个像素的平均值，最终使得图片格式与标准格式相符。合并操作应该尽量在图片的边缘进行以减小对分割出的舌体图片的影响。

获取模型分类耗时

调用模型对选择的数据集进分类，分类开始到结束的耗时就是模型分类的时间性能

获取模型分类准确率

调用模型对选择的数据集进分类，分类结束时的分类结果准确率就是模型分类的准确性性能。

获取模型训练速度

调用模型对选择的数据集进分类，分类开始到误差小于或等于模型结束的迭代次数就是模型训练的时间性能，若最后输出的迭代次数与训练时的限制迭代代数相同，则输出消息提醒模型管理员当前设置的误差要求对于此模型的训练次数参考价值较低。

设置误差要求

将系统管理员设置的误差要求作为参数传入，赋值给模型使用的成员变量。

数据结构设计

预处理：

|  |  |  |
| --- | --- | --- |
| 变量名 | 变量数据类型 | 变量说明 |
| picture | Int[1024][1024][3] | 存储图片，第一维度为水平方向像素索引，第二维度为垂直方向像素索引，第三维度为像素颜色RGB值。 |
| rawPic | Int[1024][1024][3] | 存储原始舌象图片，第一维度为水平方向像素索引，第二维度为垂直方向像素索引，第三维度为像素颜色RGB值 |
| threadNum | Int | 限定并发的线程数量 |
| x\_pix | Int | 水平方向的有效像素数量 |
| y\_pix | Int | 垂直方向的有效像素数量 |

查看模型性能

|  |  |  |
| --- | --- | --- |
| 变量名 | 变量数据类型 | 变量类型 |
| bias\_dem | double | 误差要求，作为测量训练速度的标准 |

数据库设计

属性说明

|  |  |  |  |
| --- | --- | --- | --- |
| 实体名 | 主键 | 属性 | 外键 |
| MachineLearningModel | ModelID | ModelID，time | ModelID |
| DataSet | DataSetID | DataSetID，picture | DataSetID |
| ModelPerformParams | ModelID，TestSetID | ModelID，TestSetID，TrainingBiasDem，TrainingIteration，TestAccuracy，  TestTimeSpent |  |

属性数据类型与字段属性

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 属性数据类型 | 字段属性 | 属性说明 |
| ModelID | int | Primary key  Auto\_increment | 分配给机器学习模型的ID，唯一标识机器模型，同时，与TestSetID构成属性需唯一标识模型性能参数 |
| DataSetID | Int | Primary key  Auto\_increment | 分配给数据集的ID，唯一标识机器模型，同时，与TestSetID构成属性需唯一标识模型性能参数 |
| TrainingBiasDem | Double | Not null | 使用模型查看模型训练速度的误差要求 |
| TrainingIteration | Int |  | 迭代次数，反映不大于误差要求情况下训练集的最小迭代次数 |
| TestAccuracy | Double |  | 测试集分类结果准确率 |
| TestTineSpent | timestamp |  | 测试集分类耗时 |
| Picture | Varchar(100) | Not null | 图片数据存储路径 |
| TestSetID | Int | Primary key  Auto\_increment | 分配给测试数据集的ID，与ModelID构成属性需唯一标识模型性能参数 |

处理流程

查看模型性能参数

预处理