**德塔Java数据分析算法引擎系统说明书V\_1\_0\_2**

**作者: 罗瑶光**

**ID:430181198505250014**

**2019年8月8日**

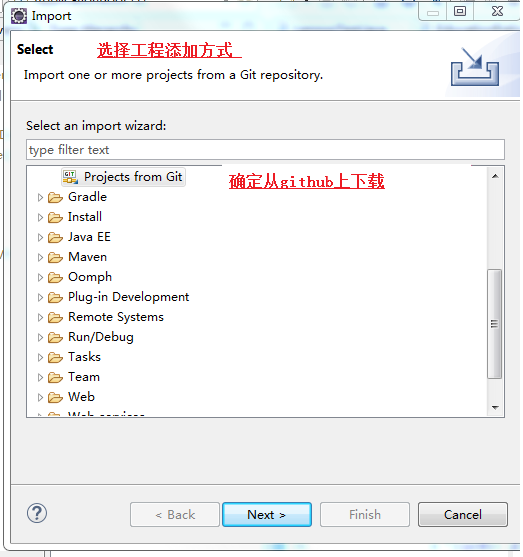
1. **起源动机**  
   2009年报,作者在印度基督大学的数据结构实验室 与Rohini 老师说,自己 能用 linklist 开发 大富翁游戏 , 为了证明, 在书店花了500卢布买了本 Java how to program 6th 第一次接触 java.  
     
   2012年作者在加州路德大学的计算机视觉课为了设计etl节点处理像素流, 把作业设计成了API包, 这是<德塔数据分析算法引擎系统>最早模型. 作者更新在 313699483 qq ID 中.  
     
   2013年后的 ETL Unicorn 设计出来之后, 作者逐步设计 语音, 音像, 电影, 商业分析节点,于是将计算机视觉中的先贤们的算法 从 2维 设计成 1维 处理 , 慢慢扩充优化丰富这个算法包.
2. **应用特色:**  
   适用于所有图片, 电影, 音乐的流数据处理. 作者一直在做算法加速优化.  
   适用于所有基于 波动粒子 工程的数据处理(天文, 地理, 勘探, 通讯, 盲分, 交通).
3. **使用方法**

**3.1下载 java 开发软件:**

Eclipse: https://www.eclipse.org/

Intellij: <https://www.jetbrains.com/idea/>

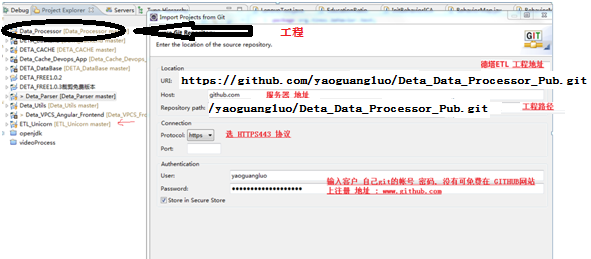
**3.2 导入 deta图灵 api ( API 是类库,接口 的意思, select 是选择 的意思 )**



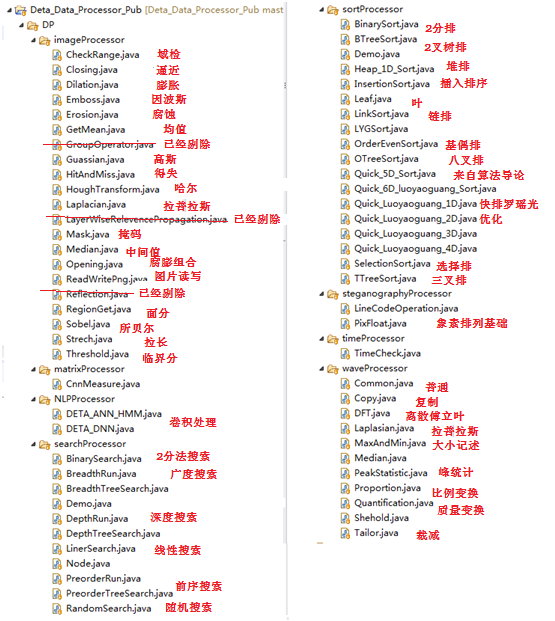
**3.3 点URI (uri是互联网传输的一种协议规范关键字)**



**3.4输入 Git导入目标地址 (git是版本持续化控制软件, repository 是 git工程的下载标识, host 是远程 主机, repository path 是git工程 在主机上下载链接, protocol 是是通信协议, port是端口, authentication是密钥, user是帐户名, password是密码, store in secure store 是记录保存)**



* 1. **代码文件全局**



* 1. **可下载的免费软件 和 图片例子集合:**

https://github.com/yaoguangluo/Deta\_Data\_Processor\_Pub

**Github 非个软申请版本:**

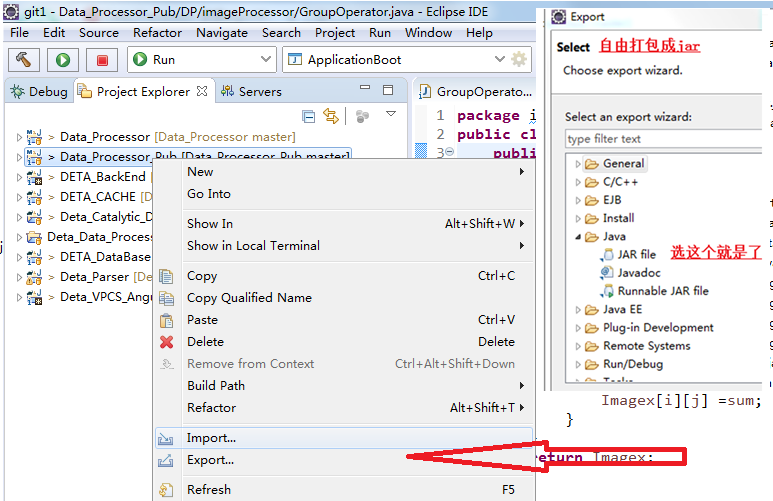
<https://github.com/yaoguangluo/Deta_Medicine>

<https://github.com/yaoguangluo/Data_Processor>

**国内:**

https://gitee.com/DetaChina/Data\_Processor

**3.8 可以任意 打包jar 作为商业 库销售和集成.( jar 是java的库的意思 , 可运行,可扩展, 可集成, export是 打包输出的意思)**



**4.功能注解：**

**4.1 API运行原理Flowchat**

各种数据分析系统

橙色为Deta数据分析算法API部分

各种数据分析系统

结束调用返回原来系统。

处理后的结果

音频处理

影像处理

像素处理

排序处理

调用德塔数据分析算法API

需要处理的1维或者2维数据

**4.2罗瑶光小高峰过滤快排4代原理**

快速排序的递归是基于2分拆分数组进行递归迭代的，假设数组的长为n， 那么 2分拆分将会有2种结果。当n 为偶数时，则为完整拆分。当n 为基数时，问题来了。n/2 将会有一定的概率 在2分递归的时候不对等。这个不对等产生的分配概率会形成许多短暂的运算小高峰时序。

概率的拆分如图：

基数产生缺陷

如果中间拆分时为基数，将会产生概率分配误差的噪声峰，基于左右对比来确定一个线性方向，可以快速降解这个峰值。另外当一个单元的while循环次数增多时，峰值的利用增大，有效的计算性能增加。于是确定线性方向x 选边缘最大数值。最后采用迪摩根分配律进行微分条件，性能再加速。于是比对算法快速排序有效的平滑这个问题：如下：

**private** **int** partition(**int**[] a, **int** lp, **int** rp) {

**int** x= a[lp]< a[rp]? a[lp]: a[rp];

**int** lp1= lp;

**while**(lp1< rp){

**while**(!(a[lp1]>x|| lp1>=rp)) {

lp1++;

}

**while**(a[rp]>x){

rp--;

}

**if**(lp1<rp){

**int** temp=a[rp];a[rp]=a[lp1];a[lp1]=temp;

}

}

a[lp]=a[rp];a[rp]=x;

**return** rp;

}

**4．3维度卷积计算原理：**

**4.3.1 给定一个需要处理的矩阵如下：**

44

46

45

41

43

42

34

36

35

31

33

32

14

16

15

24

26

25

23

22

21

13

12

11

**4.3.2 给一个卷积核算子如下：**

12

11

22

21

**4.3.3卷积处理方式：通过蓝色部分和橙色部分进行相应的内核计算操作，进行局部卷积微积分处理数据矩阵。**

44

46

45

41

434343

42

34

36

35

31

33

32

14

16

15

24

26

25

23

22

21

13

12

11

**注解：我再这里给出例子是 长为2宽为2的内核，根据实际情况可以自定义修改，如果像处理矩阵边缘数据，需要反射扩张增加原矩阵的长和宽。**

**4.3.4 先做红色和蓝色的部分卷积 相同颜色做相应算法运算如下**

**4.3.5从左到右进行相应卷积运算**

13

14

12

11

12

11

22

21

24

21

22

23

34

31

33

32

**4.3.6从左到右进行相应卷积运算，运算的虚线集为已经处理过的待输出数据。**

13

14

12

11

11

12

21

22

24

21

22

23

34

31

33

32

**4.3.7到了右边界 宏诺伊曼的思想从上到下处理。**

13

14

12

11

11

12

21

22

24

21

22

23

34

31

33

32

**4.3.8 然后再从左到右相应卷积处理。**

13

14

12

11

22

21

23

24

12

11

22

21

34

33

32

31

**注解：卷积算子有很多种，根据不同的工程应用，请选择合适的算法卷积算子**

**5. 适用范围**

适用于所有图片, 电影, 音乐的流数据处理. 作者一直在做算法加速优化.  
适用于所有基于 波动粒子 工程的数据处理(天文, 地理, 勘探, 通讯, 盲分, 交通).

**6. 注意**

注意1: 该作品免费版本使用权由国际软件研发协议apache-2.0 证书保护. 任何单位任意修改集成使用时请标注关键字:“罗瑶光”

注意2 本系统 基于先贤的算法理论思想进行Java编译, 排名不分先后.  
先贤名单: 祖冲之, 高斯, 傅立叶, 牛顿, 薛定锷, 拉普拉斯, 巴特沃斯, 所贝尔, 哈尔, 因波斯, 迪摩根, 霍夫 等.  
注意3 本系统 中的先贤算法思想 作者 无权所有, 有权应用. 同样客户和使用者 可以自由化应用该 软件系统, 但无权所有其Refer 的先贤算法 以及思想理论的国籍归属等一系列所属权修改.

注意4: 当前版本是1.0.0, 一直在优化中,有任何bug请直接联系作者.

QQ: 2080315360 (qq: 腾讯)

WECHAT: 15116110525 (WECHAT 微信)

TEL: 15116110525 (tel: 电话号码)

EMAIL: [2080315360@qq.com](mailto:2080315360@qq.com) ( email: 邮件地址)

**6.1 德塔数据分析实现集合涉及的先贤理论思想和发现。互联网链接：**

傅里叶思想：

<https://www.doc88.com/p-0973999656458.html>

<https://baike.sogou.com/v102060438.htm?fromTitle=%E5%82%85%E9%87%8C%E5%8F%B6>

拉普拉斯思想：

<https://baike.sogou.com/v815845.htm?fromTitle=%E6%8B%89%E6%99%AE%E6%8B%89%E6%96%AF%E7%AE%97%E5%AD%90>

<https://baike.sogou.com/v181283.htm?fromTitle=%E6%8B%89%E6%99%AE%E6%8B%89%E6%96%AF>

浮雕思想：

<https://en.wikipedia.org/wiki/Image_embossing>

高斯思想：

<https://baike.sogou.com/v7811633.htm?fromTitle=%E9%AB%98%E6%96%AF%E6%A0%B8%E5%87%BD%E6%95%B0>

腐蚀膨胀思想：

<https://baike.sogou.com/v178215318.htm?fromTitle=%E9%97%AD%E8%BF%90%E7%AE%97>

索贝尔思想：

<https://baike.sogou.com/v101294427.htm?fromTitle=Sobel%E7%AE%97%E5%AD%90>

Cnn：

<https://en.wikipedia.org/wiki/Convolutional_neural_network>

Rnn：

<https://en.wikipedia.org/wiki/Recursive_neural_network>

Ann：

<https://en.wikipedia.org/wiki/Artificial_neural_network>

Lwa:

<https://baike.sogou.com/v11009683.htm?fromTitle=bp%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C>

[https://baike.sogou.com/v11009683.htm?fromTitle=bp神经网络](https://baike.sogou.com/v11009683.htm?fromTitle=bp%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C)

BST:

<https://baike.sogou.com/v71134459.htm?fromTitle=AVL%E6%A0%91>

Quicksort:

<https://baike.sogou.com/v7638866.htm?fromTitle=%E5%BF%AB%E9%80%9F%E6%8E%92%E5%BA%8F>

Heapsort:

<https://baike.sogou.com/v4838322.htm?fromTitle=%E5%A0%86%E7%A7%AF%E6%8E%92%E5%BA%8F>

Selectsort:

<https://baike.sogou.com/v73874629.htm?fromTitle=%E9%80%89%E6%8B%A9%E7%B1%BB%E6%8E%92%E5%BA%8F%E6%B3%95>

Insertsort:

<https://baike.sogou.com/v7832934.htm?fromTitle=%E6%8F%92%E5%85%A5%E6%8E%92%E5%BA%8F>

**7. 感谢**

1 感谢 Renhat教授 提供了 Reflection 函数(已经移出个人著作权范围)  
2 感谢 同学 罗阳参与设计 group 函数(已经移出个人著作权范围)  
3 感谢 同学 高顺参与讨论 Hull算子不稳定熵(已经移出个人著作权范围)

4 感谢 Vicent Boucher 提供的LWA理论思想(已经移出个人著作权范围)  
5 感谢 同学 Mahesh参与讨论 linklist的C语言快速书写方式(已经移出个人著作权范围)

Deta 项目设计 采用 Mind Master 软件.

Deta 项目研发 采用 Eclipse IDE 软件.

Deta 项目测试 采用 JUNIT API 软件.

Deta 项目作品 主要采用 JAVA JDK8+.

Deta 项目编码和算法基础能力来自作者在印度基督大学 学习的 数据结构 课程. 作者长期使用 微星 windows 7操作系统开发此项目, 电脑装360 杀毒软件保证其高效研发环境. 感谢 github和gitee 备份, 节省了作者 大量的存储硬盘, 同时方便 查阅, 逻辑 的鼠标键盘 给作者 提供了迅捷 的输入输出 便利 .当然 电信的网络, 老爸,老妈, 都要感谢的.

**8 研发需要清单**

**8.1** Java 编辑器.

**8.2**  Jdk8+. Java 虚拟机运行环境.

**8.3** Junit 测试包.

**8.4** 一台连网的电脑.