

浮游动物识别

戴嘉伦 武斌 朱亚菲 王如晨

2015 年 6 月

目录

1 ZooScan 扫描仪	4
1.1 操作步骤	4
1.1.1 创建项目	4
1.1.2 扫描背景	4
1.1.3 准备样品	5
1.1.4 扫描样品	6
1.1.5 回收样品	7
1.2 注意事项	7
2 ZooProcess	8
3 Plankton Identifier(PkID)	8
3.1 主窗口	8
3.2 Learning	9
3.3 Evaluation	13
3.4 Prediction	17
3.5 Validation	19
3.6 Compilation	21
4 评价方法	23
4.1 论文中采用的评价方法	23
4.2 怎样得到混淆矩阵	25
4.3 PkID 中的评价参数	25
4.4 混淆矩阵 Confusion matrix (CM)	26
4.5 交叉验证 Cross Validation (CV)	29

4.5.1 训练集和测试集	29
4.5.2 常见的交叉验证方法	29
5 特征 (PkID)	30
5.1 位置特征	30
5.2 尺寸特征	31
5.3 灰度值特征	31
5.4 形状特征	32
5.5 自定义特征	33
6 识别结果分析	33
7 优缺点分析	37
8 竞赛	37
8.1 PASCAL VOC Challenges 2005-2012	37
8.2 IMAGENET Large Scale Visual Recognition Challenges 2010-2015	38
8.3 Labeled Faces in the Wild	38
9 特征 (Ours)	38
9.1 经典特征	38
9.1.1 HOG 特征	38
9.1.2 SIFT 特征	41
9.1.3 SURF	43
9.1.4 LBP	44
9.1.5 FV(Fisher vector)	45
9.1.6 哈尔特征 (Haar-like features)	45
9.2 其它特征	46
9.2.1 GIST	46
9.2.2 Harris-Laplace	46
9.2.3 GMM-HMM(Gaussian Mixture Model-Hidden Markov Model)	46
9.2.4 Gabor	46
9.2.5 DoG	46
9.2.6 local symmetry	46

9.2.7 Covariance Matrix	48
9.2.8 骨架特征	48
9.2.9 灰度特征	49
9.2.10 凸包	50
9.2.11 ICA(Independent Component Analysis)	50
9.2.12 SC	50
9.2.13 Binary SIFT (BSIFT) – Zhou, TIP 2014	51
9.2.14 COGE - PCM 2013 Best Paper, Mao , 2013	51
9.2.15 Ultra-Short Binary Descriptor (USB) – Zhang S., TIP , 2014	51
9.2.16 Topology-Preserving Hashing – Zhang L., MM 2012	51
10 学习算法	52
10.1 贝叶斯学习	52
10.2 SVM 方法	52
10.3 K-means	52
10.4 Decision tree	52
10.5 EM(Expectation Maximization Algorithm)	52
10.6 KNN	52
10.7 集成学习 (Ensemble learning)	53
10.7.1 AdaBoost	53
10.7.2 Bagging	53
10.7.3 Gasen	54

由法国国家科学研究院 Villefranche 海洋实验室的 Gorsky 等人发明 [5]、Hydroptic 公司生产的 ZooScan 浮游动物图像扫描分析系统是一种实验室成像系统 (即针对的是已经采集并固定保存的浮游生物样品)，主要用于对液体中的浮游动物样品进行计数、大小测量、种类鉴定以及生物量测定。ZooScan 系统采用非破坏性技术对液体浮游动物样本进行分析，样品可重复使用。

ZooScan 系统是由 ZooScan、ZooProcess 和 Plankton Identifier(PkID) 等共同组成的，ZooScan 是硬件部分，主要进行浮游动物样品扫描，形成数字图像。ZooProcess 和 PkID 是软件部分，分别以标准化的程序处理原始图像、对不同个体的形态参数进行自动测量和对图像中的浮游动物进行自动分类和计数。

1. ZooScan 扫描仪

ZooScan 扫描仪完成的任务：

- 扫描空白背景
- 扫描样品，获得原始图片和元数据信息

1.1 操作步骤

1.1.1 创建项目

- 打开“Image J”，在软件界面中，选择项目选项表最后的“CREATE NEW PROJECT”，创建一个新的项目。如图 1。

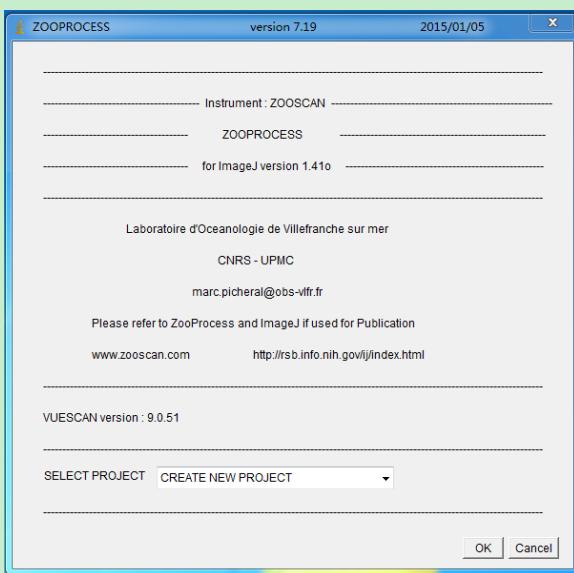


图 1: 创建项目

- 选择扫描选项。选择 2400dpi 分辨率和“Large”的扫描框 ($15cm \times 24cm$)。如图 2。
- 输入样品的元数据 (metadata)。在后面处理过程中，可以通过 ZooProcess 中的“EDIT and MODIFY metadata”工具来修改元数据。如图 3。

1.1.2 扫描背景

背景图像是一张空白图像，用于图像分析过程中。在与样品相同环境下 (自来水或是过滤海水)，先扫描背景再扫描样品。最好是在每个扫描任务开始时，都扫描一次背景图像。

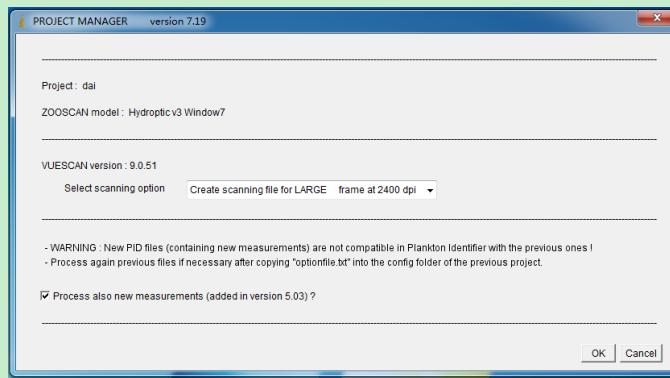


图 2: 选择分辨率和扫描框大小

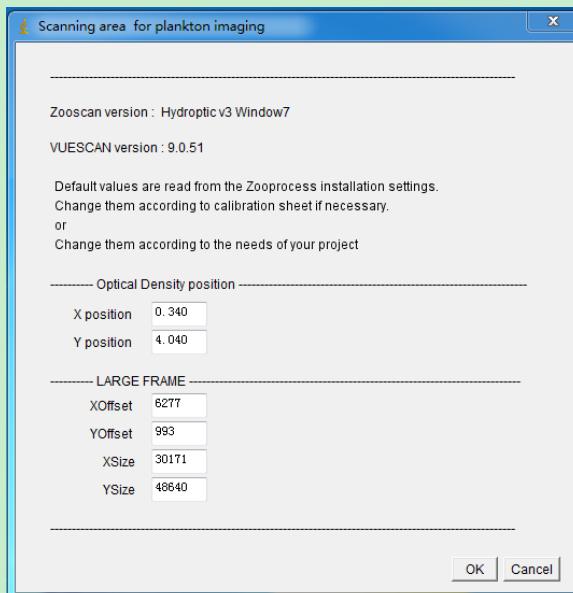


图 3: 输入扫描信息

- 用清水清洁和冲洗 ZooScan 托盘和表面玻璃，时不时地检查并清除在玻璃和扫描框上的污点。
- 倒一些清水（保持在室温）没过托盘，它可以防止扫描框刮擦托盘。
- 放置扫描框 ($15cm \times 24cm$)，这取决于之前 ZooProcess 中的“创建项目”中的选项。
- 在预扫描和实际扫描之间，等待 30 秒。

1.1.3 准备样品

- 存储几升清水，保持在室温，用来为 ZooScan 注水。
- 用筛子（网格间隙为 $100\mu m$ ）过滤掉防腐剂和海水中物质。样品通过间隙为 $1mm$ 和 $200\mu m$ 的两种网格，将浮游生物分成不同体型的两部分。

- 浮游生物被分为不同体型的两部分：一个为体型大的样品，另一个为体型小的样品。分别将分开后的样品，添加标签 $d1$ 和 $d2$ ，用于扫描后的数据处理过程中。如图 4。

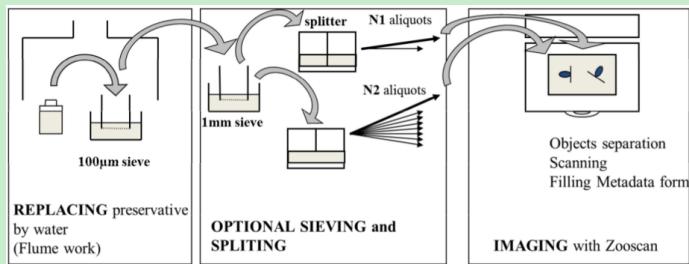


图 4: 准备样品

- 使上述两个分开后的样品中，保持只有 1000 – 1500 个浮游生物。

1.1.4 扫描样品

- 在扫描托盘中加入量水，放置扫描框 ($15cm \times 24cm$)，调整扫描框放置的位置（位置在扫描托盘中有标注）。
- 倒入样品，加入清水，直到没过扫描框的台阶。
- 将个体较大的浮游生物放在扫描框的中心，用木棒分离粘连浮游生物，避免浮游生物贴靠扫描框边缘。对于漂浮在水面的浮游生物，轻轻用木棒将浮游生物压入水中。如果存在无法没入水中的浮游动物，且数量不多，则将它们移出（[这一步是生成好的数据质量的关键步骤](#)）。如图 5。



图 5: 用木棒分离粘连浮游生物

- 检查托盘是否有气泡，顶盖玻璃的表面是否冷凝。
- 加载 ZooProcess，选择项目，点击扫描样品，选择两部分样品中的一个样品 (d_1 和 d_2)，写入相关元数据。

1.1.5 回收样品

- 清洁托盘，避免下次实验污染了样品。
- 移走并冲洗透明的扫描框，回收所有的标本。
- 清洁干燥扫描托盘。

1.2 注意事项

- 清水可以是自来水或过滤海水，保持在室温是为了避免在 ZooScan 的托盘中产生气泡或者顶盖玻璃上出现冷凝。因为自来水管中的自来水和房间的温度存在一定温差。
- ZooScan 提供 1200dpi 和 2400dpi 两种分辨率扫描。分辨率限制在 2400dpi，是由于 ZooScan 设计的光路需要空气入水和由水进入玻璃两次穿过界面，使成像分辨率收到限制。
- ZooScan 的扫描框有两个尺寸： $11\text{cm} \times 24\text{cm}$ 和 $15\text{cm} \times 24\text{cm}$ 。推荐使用 $15\text{cm} \times 24\text{cm}$ 的扫描框，具体使用哪个扫描框由 ZooProcess 中的选项决定。
- 扫描空白背景不仅可以去除灯光产生的异质性的斑点等，而且可以检验系统的稳定性。
- 在准备样品阶段，通常情况下，将样品分为两个或以上的小样品进行扫描。
- 在准备样品阶段，如何将浮游生物样品分成不同的小样品，取决于原样品中浮游生物的种类多少与体型大小。
- 扫描的浮游生物必须保持不动的状态，使用固定剂或将其麻醉。
- 扫描框上有 5mm 的小台阶，注入的水必须漫过这台阶的高度，避免在扫描后的图像边缘出现弯液面现象。如图 6。
- 在较大的框内 ($15\text{cm} \times 24\text{cm}$)，最多可以容纳 1000 – 1500 个浮游生物。
- 在 ZooProcess 中也可以对扫描图像中的粘连浮游生物进行分离，但是最好是在样品扫描之前用木棒进行分离。如图 7。

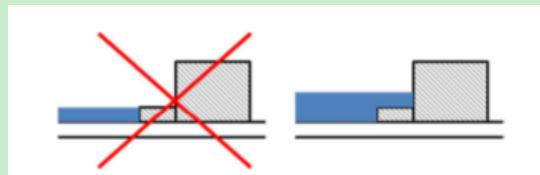


图 6: 框的台阶高度

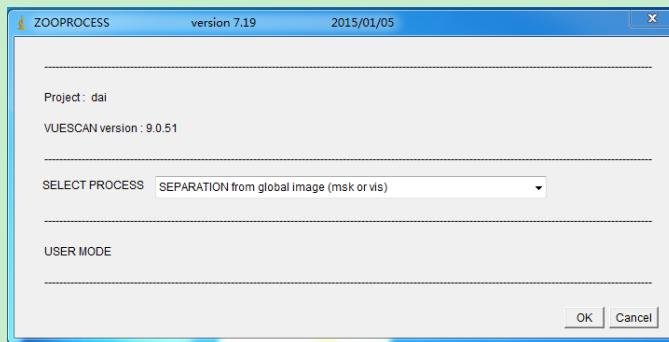


图 7: ZooProcess 的浮游生物分离

2. ZooProcess

ZooProcess 介绍详见<http://wbtwd2004.github.io/zooscan/2015/06/13/Zooprocess-Study.html>。

3. Plankton Identifier(PkID)

3.1 主窗口

打开 PkID 应用程序,显示的界面上共有 5 个按钮：“Learning”、“Evaluation”、“Prediction”、“Validation” 和 “Compilation”（图 8）。这 5 个部分可以相互独立地运行,但是会有运行的先后顺序。比如“Evaluation”和“Prediction”会用到“Learning”生成的文件,“Validation”用到“Prediction”生成的文件,“Compilation”用到“Validation”所生成的文件。因此,当你第一次使用 PkID 时你应该首先运行“Learning”。

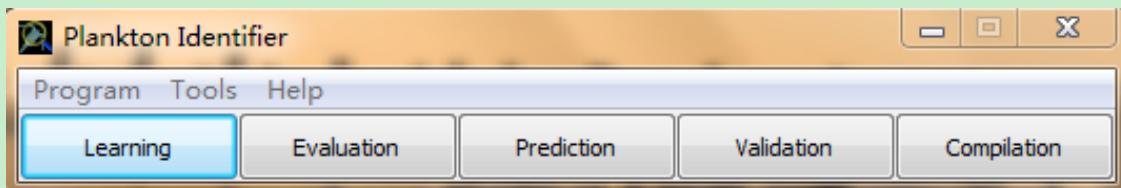


图 8: 主窗口

其他菜单:

Program > Settings 定义 Tanagra.exe 所在的路径以及存放缩略图、PID 文件和结果的默认文件夹路径。

Tanagra Path: 如果你安装了两个以上的 Tanagra 版本, 你可以选择想要使用的版本。点击 **Browse**, 浏览硬盘文件夹, 找到 Tanagra.exe, 点击 **OK**。

注: 如果你的 Tanagra 没有安装在 \Program Files\Tanagra 路径下, 或者你就根本没有安装 Tanagra, 这时当你运行 PkID 时就会自动弹出这个窗口 (图 9)。

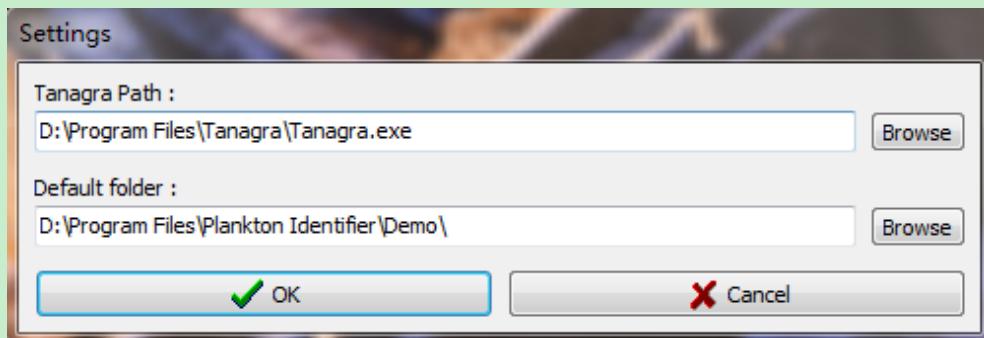


图 9: 设置窗口

Default folder: 默认文件夹是你第一次使用 PkID 执行一些步骤产生的文件所存放的地方。

Program > Exit 关闭 PkID。

3.2 Learning

这一步会生成一个学习文件以用作后续的自动识别。它对应于经专家鉴定的具有代表性的一些物体的子样本并且可以作为将来分析的参考。

1、文件夹选择窗口

当点击 **Learning** 按钮时, 会出现如图 10 所示的文件夹选择窗口。

选择一个空的文件夹以用来创建新的训练数据集, 你也可以通过点击右上角的按钮 (图 10: 1) 来创建一个新的文件夹然后给它命名 (图 10: 2), 最后点击 **OK** 按钮 (图 10: 3)。

你也可以选择一个已有的训练数据集 (其中包含已分类好的子文件夹以及包含物体元数据的一些 PID 格式的文件, 每个子文件夹代表一个种类, 其中存放属于该类的 jpg 格式的缩略图), 如图 10: 4。

注: 如果此处选择的文件夹结构不符, 或者包含了无效的数据, 那么将会无法打开, 并且在窗口的最下方会出现一行红色的警告信息以说明原因。

2、学习窗口

当选择了一个可以用来对缩略图进行分类的有效文件夹之后, 点击 **OK**, 会出现一个新的窗口 (图 11)。其中左边部分 (“Sample Set”) 是要通过浏览硬盘文件夹来选择未被分类的样本。右边部分

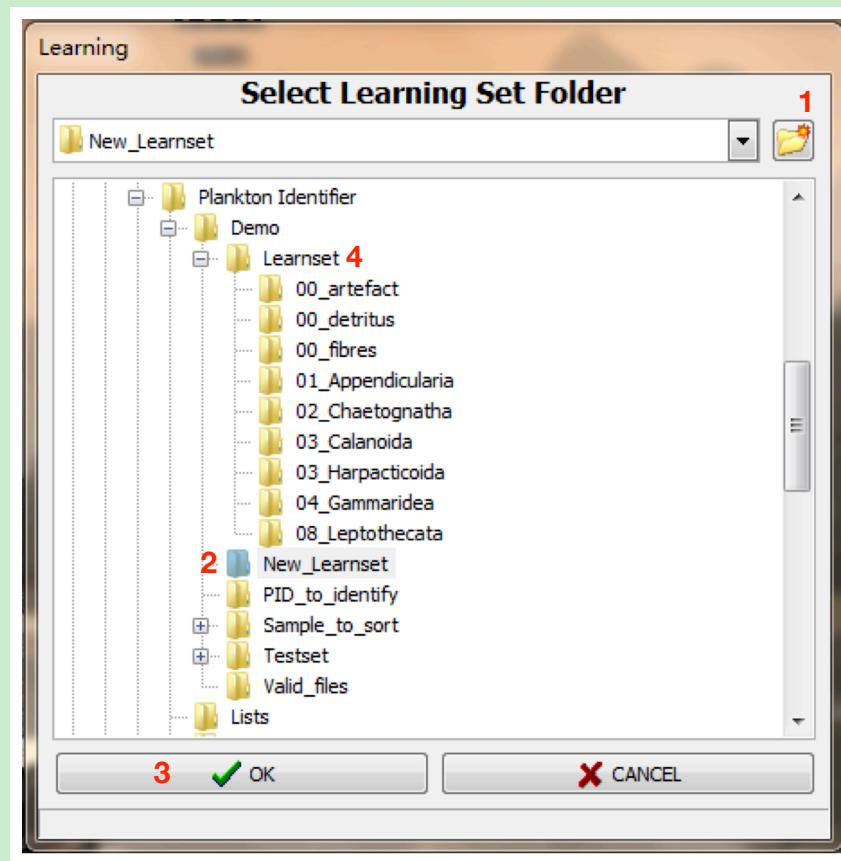


图 10: 训练集文件夹选择窗口

(“Learning Ser”) 是要将左边未被分类的样本拖到右边以完成分类 (创建子文件夹)。

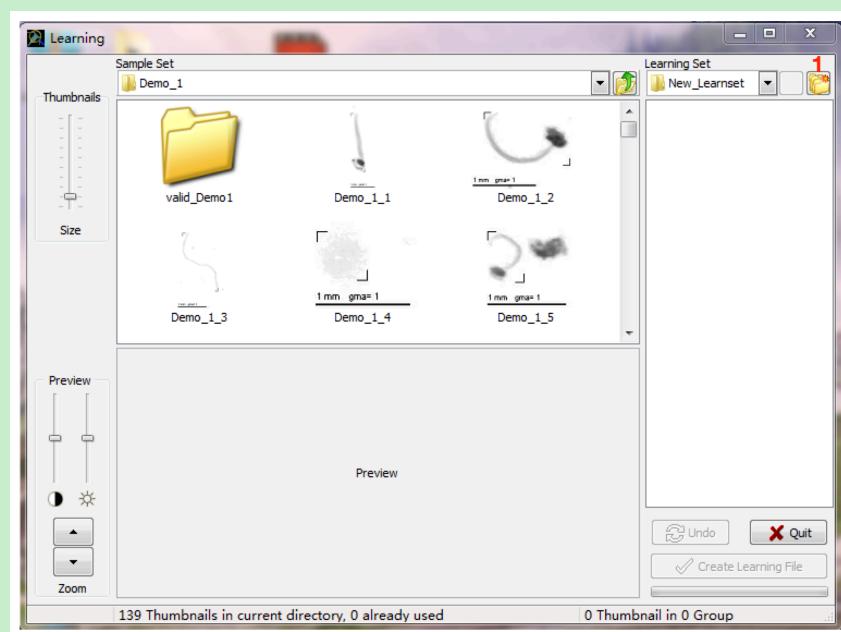


图 11: Learning 窗口

Sample Set

在窗口左边栏“Sample Set”处，浏览硬盘文件夹以打开一个包含未分类的缩略图和相应 PID 文件的文件夹。只有有有效名字（<Sample Name>_<Item Number>.jpg）的缩略图才会在左边栏中显示出来。如果名字是有效的但是 PID 文件中并没有包含该幅图像的一些数据，这时就会这个缩略图上方会出现一个问号并且当把鼠标放在问号上时会有相应的原因解释，这幅图也是无法使用的。

注：如果此处选择的文件夹结构不符，或者包含了无效的数据，那么将会无法打开，并且在窗口的最下方会出现一行红色的警告信息以说明原因。

类别（子文件夹）创建

在窗口右边栏“Learning Set”处，为了把缩略图归到对应的类别，需要在步骤 1 所选择的文件夹中创建一些子文件夹，每个文件夹代表一类。点击右上角的按钮（图 11: 1）创建新的文件夹，这时会出现一个新的窗口，可以给新创建的文件夹从给定的种类名字中选择一个（图 12）。如果给定的这些名字中没有合适的，你还可以选择另一个“Predefined Lists”或者先选择“New”作为名字，之后再在“Learning Set”这一栏中重新编辑命名。

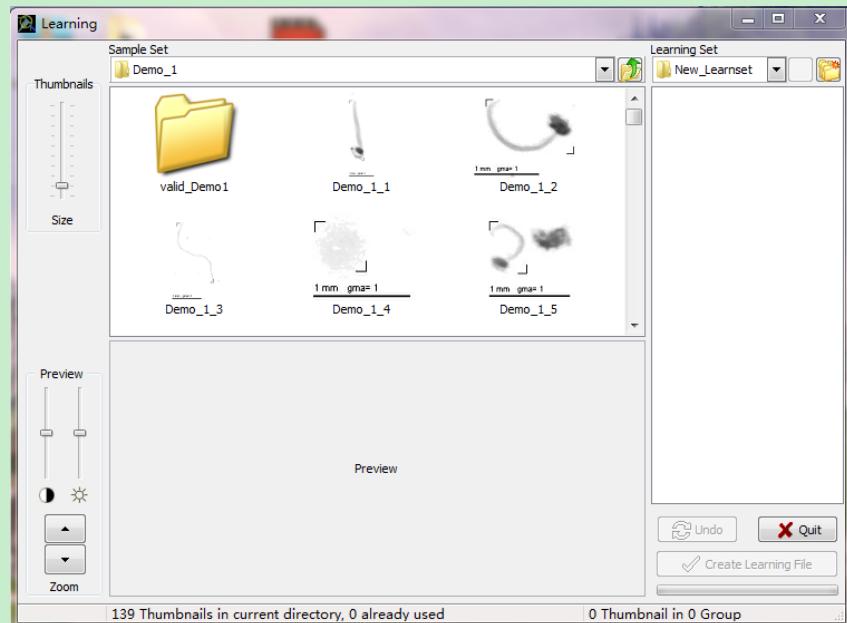


图 12: 创建训练集中分类类别的窗口

注 1: 所创建的子文件夹名字不能重复。已经用过的名字在“Create Group Folders”窗口中将不再显示。

注 2: 可以通过 *Tools>Import Name List* 菜单来自定义你自己的种类名字清单。

对缩略图进行归类

选中一个缩略图，下方会显示它的预览，如图 13: 1。你可以通过左侧的 zoom 按钮（图 13: 2）

将预览结果放大，还可以通过调节左侧对比度和亮度的状态条来显示更多细节。

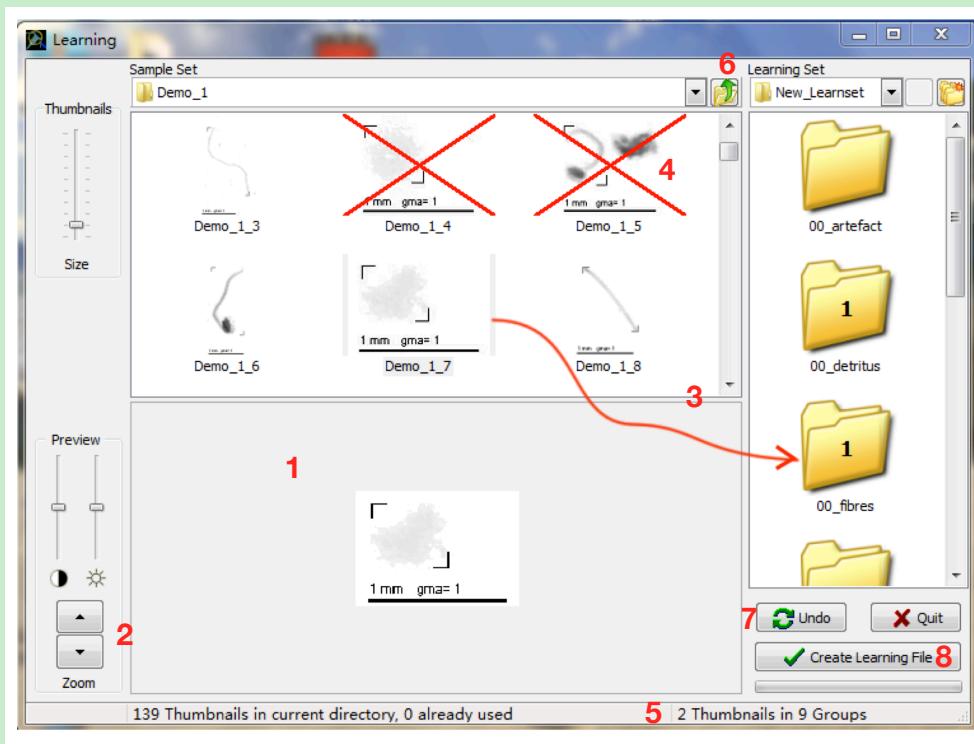


图 13: 对样本文件中的缩略图进行分类

把缩略图拖到相应的子文件夹中（图 13: 3），这时缩略图会被复制到那个子文件夹中，而不是被剪切过去了。对应的 PID 文件也被复制到了右边文件夹中，但在这个窗口中是看不见的，以免视觉上的混淆。一旦一个缩略图已经用在了所创建的数据集中，上面就会出现一个红色的叉（图 13: 4），表示它不能再被使用了。

一次选中多个缩略图也是可以的，可以通过按住 Ctrl 键来完成。在选中多个缩略图的时候，如果你想看每一个缩略图的预览效果，可以从右下角往左上角选择。

每一个子文件夹中缩略图的数目会显示在这个子文件夹上，并且随着你的操作而更新。已经被归类了的缩略图数目以及还没有完成归类的缩略图数目都被显示在了窗口的最下方（图 13: 5）。

你可以对多个样本集中的缩略图进行归类以创建自己的训练数据集。点击上面的一个按钮（图 13: 6) 转到你想要操作的文件夹路径下。

Cancel action

当你在训练数据集的创建过程中执行了一些误操作之后，可以有以下两个办法取消：(1) 用 Undo 按钮（图 13: 7）(2) 打开右侧的子文件夹，选中缩略图然后用 DEL 键删除。其中 Undo 键可以用来取消删除操作或子文件夹删除操作，DEL 键可以用来删除一个所有缩略图都被清除了的空子文件夹。

创建学习文件

一旦你觉得你创建的每一个类别中已经归类了足够多的样本，你就可以点击“Create Learning File”按钮（图 13: 8）了，点击之后会出现一个保存对话框，上面显示了所要保存的目标文件夹路径以及学习文件的名字，默认的名字为 Learn_<number> 格式。点击“Save”按钮，所有的学习工作就完成了。这时又会出来一个对话框询问你是否要继续分类，如果你选择“No”，这个学习窗口就会被关闭回到主窗口。

3.3 Evaluation

这一步是要帮助你评估一下基于上一步中创建的训练数据集所建立的预测模型对训练数据集中的物体的识别率有多高。最后会生成一个包含识别结果的文本文件以及一个包含了数据分析信息的 html 报告。点击主窗口中的 Evaluation 按钮，会出现如下的界面（图 14）。

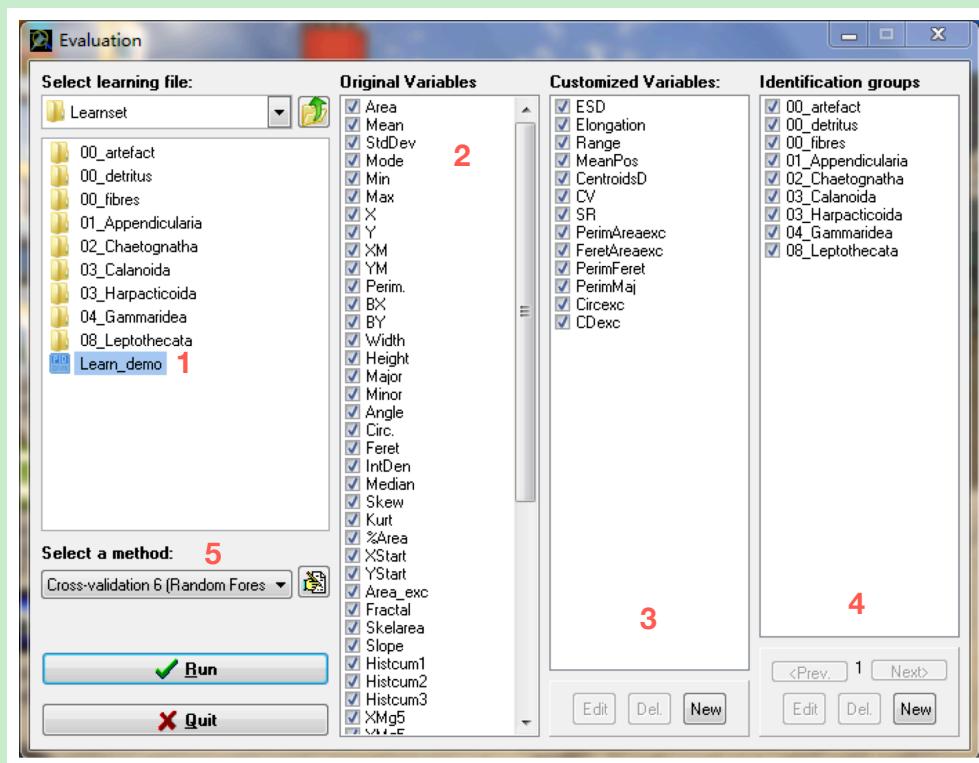


图 14: Evaluation 窗口

1、选取学习文件（图 14: 1）

浏览硬盘文件夹，选择你想要用来进行数据分析的学习文件。

注 1：选择学习文件后才能激活其它部分。

注 2：双击 PID 文件，将会自动在 PID viewer（如果已安装）或者文本编辑器（例如 Windows 下的 Notepad）中打开，从而可以校正文件内容。

2、初始变量（图 14: 2）

这里展示的是所选择的学习文件中的一些变量，你可以任意选取一些变量以用作分析，没有被选取的初始变量在计算时会被忽略，但不会从结果文件中移除。

注：对于那些用 ZooProcess 软件生成的 PID 文件，在计算时要被忽略的初始变量可以在附件中找到。

3、自定义变量（图 14: 3）

这一步是要根据已有的初始变量创建自定义变量。在你安装 PkID 之后会有 13 个自定义变量可供你选择是否要用它。没有被选取的初始变量在计算时会被忽略，并且不会在结果文件中显示。如果定义的某个变量不能从已经选取的初始变量计算，那么它会自动变成不可选取状态，并且显示成灰色。

要编辑一个已经存在的自定义变量，选中它，点击 Edit 以打开一个新窗口（15）。

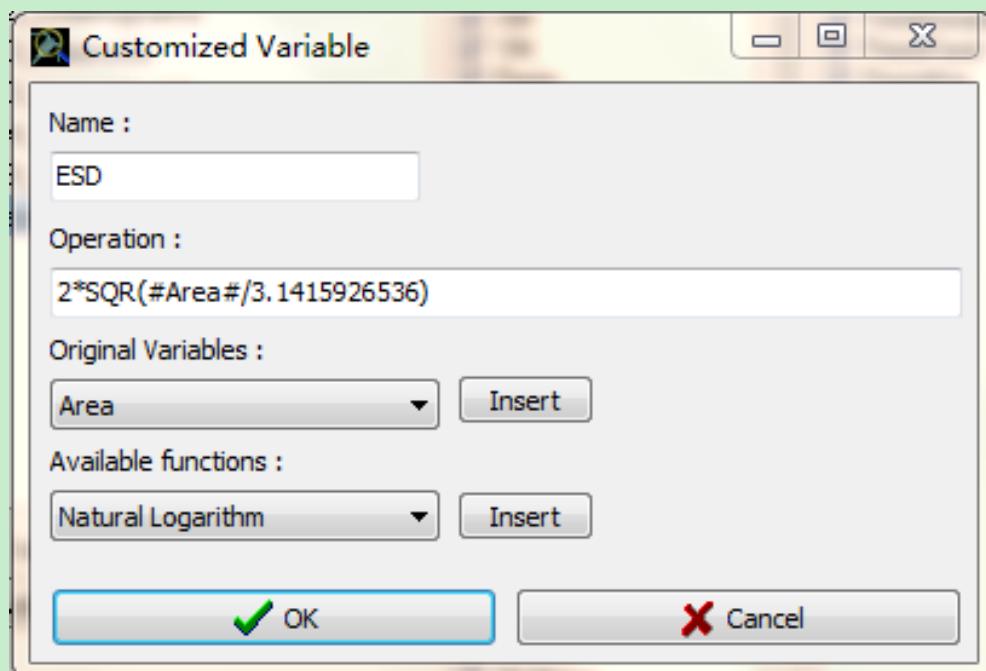


图 15：自定义变量窗口

选中已存在的自定义变量，点击 Del 可以将其删除。

要新建一个自定义变量，点击 New，出现如图 15 的变量自定义窗口。

1. 对新建的变量命名（命名必须与已有变量不同）
2. 在“Operation”下面一栏中输入计算公式
3. 按“OK”键

注：在编写公式时，一些基本的运算符（例如 +, -, \, *, ^）、括号、数字都是跟平常在键盘上敲的一样。要插入一个初始变量的话，可以在“Original Variables”下面一栏选中它，然后点击“Insert”。在

公式中用到初始变量时，需要在这个初始变量前后加上“#”号。在“Available functions”一栏中，可以选择一个函数进行插入。建议可以先看看已有的一些自定义变量是如何定义的，然后再编辑自己需要的公式。

4、Identification Groups（图 14: 4）

这里显示了在选取的学习文件中所定义的分类种类。默认的种类是不能被删除或编辑的，但是你可以通过将已有的类别合并来创建新的类别。

点击 New 可以创建新的类别，打开如图 16 所示的类别编辑窗口。

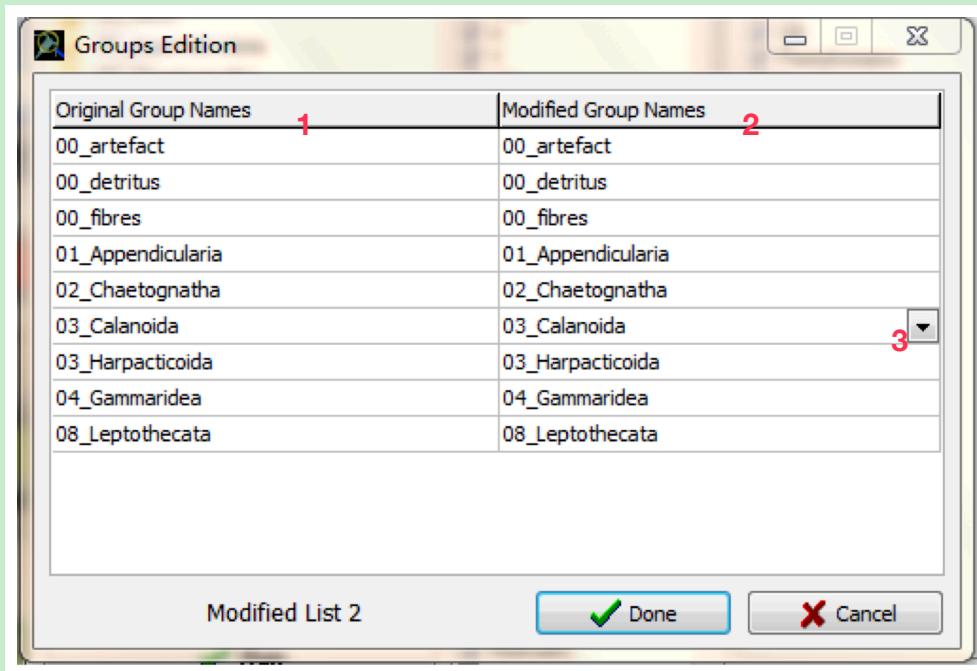


图 16: 分类类别编辑窗口

修改初始类别名字（图 16: 1）：

1. 在“Modified Group Names”一栏下方选择一个需要修改的名字（图 16: 2）
2. 给这个类别重新编辑命名，或者点击如图（图 16: 3）所示按钮，在名字列表中选择一个
3. 完成对类别的重新命名后，点击 Done

注：你可以创建很多的分类种类的列表，后续数据分析中用到的只会是当前这个列表（在 Evaluation 窗口点击 Run 时所看见的那个列表）。你可以用 Prev. 和 Next> 按钮来选择你想要的进行分析那个列表。在生成的结果文件中，初始名字会显示在“Ident1”这一列中，新的名字显示在“Ident2”这一列中。

5、选择一种方法（图 14: 5）

这里是要选择一种评价方法，用来检测用有监督学习方法和所选择的学习文件学习的模型的识别能力。PkID 中一共提供了两种评价方法。

k-fold cross-validation method: 用重采样技术来评价学习算法在学习文件上的识别准确率。初始的训练数据集被随机分为 k 个相同大小的子集。每一次循环过程中，将 $k - 1$ 个子集放在一起形成训练集，并且构造出预测模型，剩下的那 1 个子集被当成测试集来评价模型预测能力。一共循环 k 次。这样下来，每个子集都会有 1 次机会作为测试集， $k - 1$ 次机会作为训练集。将 k 次预测结果取平均，可以得到这个模型的最终预测能力。交叉验证过程会重复 n 次， n 次交叉验证的平均错误率在一个混淆矩阵中被计算得到。

PkID 一共实现了 8 种交叉验证方法，每一个都采用了不同的学习算法。

所有的交叉验证方法中所用的参数都是 $k = 2, n = 5$ 。要想改变这两个值：

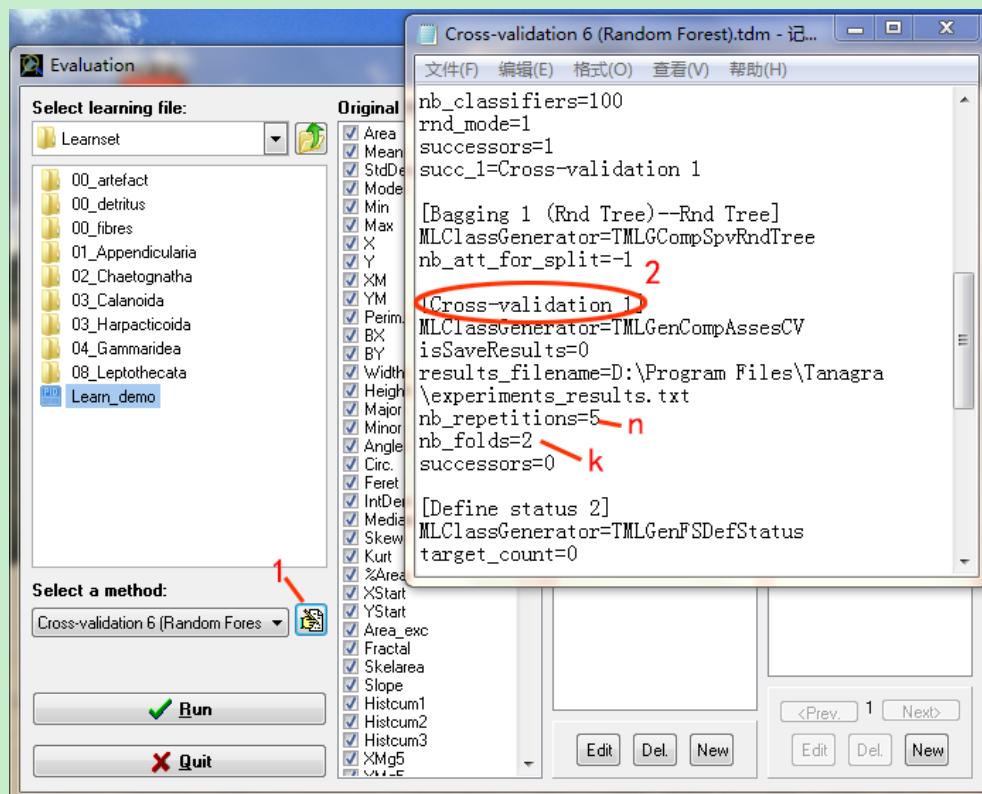


图 17: 分类类别编辑窗口

1. 点击编辑按钮（图 17: 1），打开相应的 tdm 文件
2. 在交叉验证部分修改 k 和 n 值图 17: 2
3. 在关闭 tdm 文件前保存修改

Test method: 在一个预先定义好的、独立的测试文件上对模型的准确率进行评价。在使用这个测试方法之前，需要创建一个特殊的文件，可以通过主窗口菜单中的 **Concatenate Learning Files** 来创建。相比于用两个不同的文件来作为训练集和测试集，我们更倾向于用一个文件来把它们联系起来，并且通过状态来表示所要扮演的角色，状态有“学习”和“测试”两种。

PkID 提供了两种测试方法。Test 1 可以通过在预先定义的测试文件上比较 8 个算法的准确率来选择最好的学习方法。Test 2 只用到随机森林算法。

注：通常来讲，训练集越大，所构造的分类模型越好，测试集越大，预测准确率越高。

Export to text file (no analysis): 生成一个文本文件，包含了所选的学习文件中所有初始变量和自定义变量，而不包含预测结果。这个文件还可以输入到任意的数据挖掘软件站进行分析。

6. 开始分析

当所有文件、变量、类别都选择好之后，点击 **Run** 按钮，将会出现一个保存页面，选择要保存的文件夹、文件名，点击 **Save**。

分析完成后（可能需要几分钟，这取决于样本大小和所选方法），结果和 html 报告被保存在了所选择的文件夹中，并且 html 报告会自动在网页中打开。这时会有一个对话框询问你是否要退出 Evaluation 窗口，如果你选择“Yes”，Evaluation 窗口就会被关闭并返回主窗口。

与初始 PID 文件相比，评价文件包含了以下新的列：

1. 对应于自定义变量的列
2. 包含了学习文件中分类类别的一列 (Ident)
3. 包含了修改后的类别名字的一列 (Ident2)
4. 用来表示状态的一列 (Learning 或 Test)

3.4 Prediction

这一步是要根据选择的学习文件中的种类对样本进行自动识别。结束后也会生成一个包含了自动识别结果的文本文件和一个包含了数据分析信息 html 报告。点击主窗口中的 **Prediction** 按钮，会打开一个新的窗口（图 18）：

- 1、选择学习文件（图 18: 1）
- 2、选择样本文件（图 18: 2）
- 3、选择学习算法（图 18: 3）

从 PkID 提供的 8 个学习算法中选择一个。

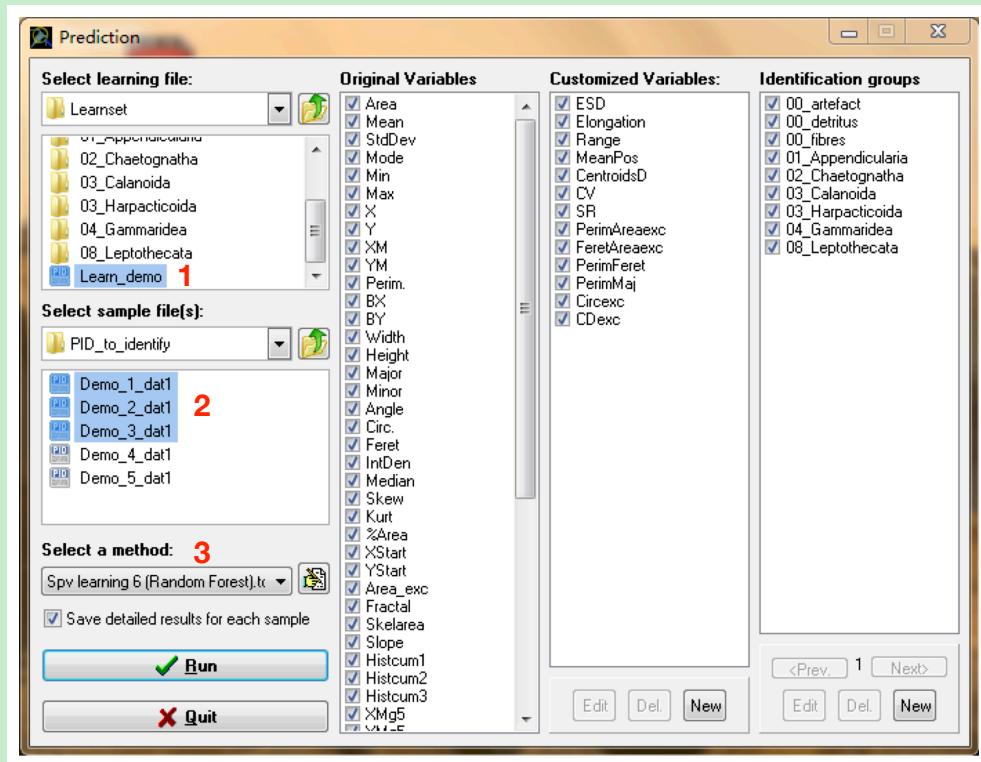


图 18: Prediction 窗口

- Spv learning 1 (5-NN)
- Spv learning 2 (C-SVC linear)
- Spv learning 3 (C-SVC RBF)
- Spv learning 4 (BVM)
- Spv learning 5 (C4.5)
- Spv learning 6 (Random Forest)
- Spv learning 7 (PLS)
- Spv learning 8 (Multilayer Perceptron)

4、开始分析

当所有文件、变量和分类类别都选好之后，点击 Run 按钮，会出现一个保存会话框。

3.5 Validation

这一步是将上一步生成的预测文件可视化，并且人为地对识别结果进行校正。这里可以有两个选择（图 19）：1) 用 Prediction 那一步生成的 Pred_.txt 文件来将预测结果可视化，真正实现缩略图的自动分类，你还可以对自动识别的结果进行进一步检查和校正。2) 打开一个已有的校正集来继续一个校正或进行二次校正。

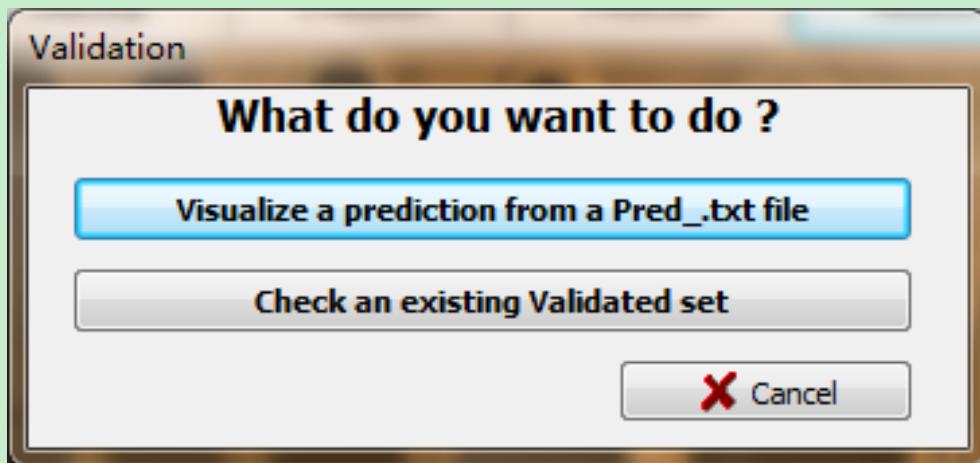


图 19: Validation 窗口：What do you want to do?

Visualize a prediction from a Pred_.txt file

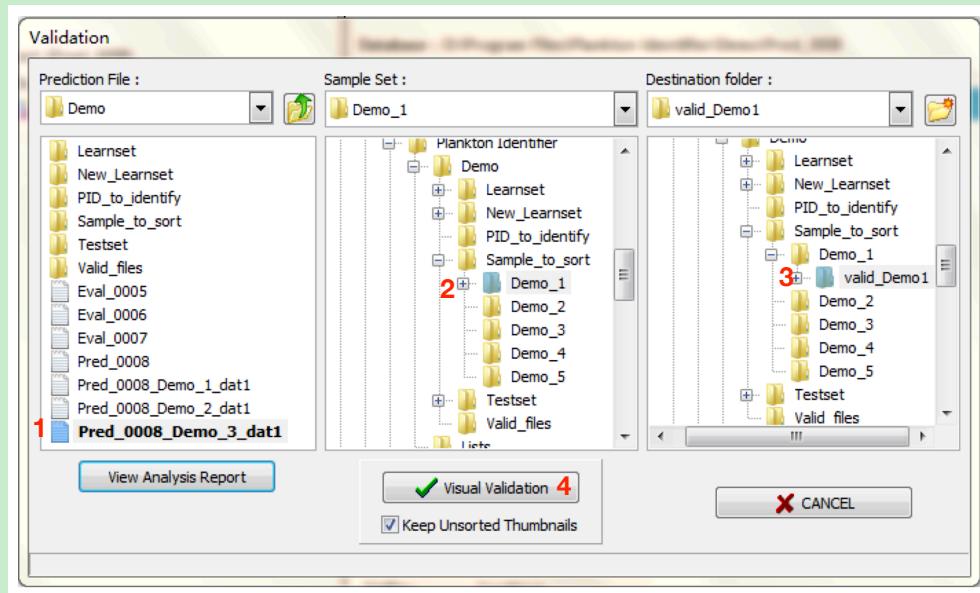


图 20: 校正选择窗口 1

1. 选择一个要用来校正的 Pred_.txt 文件（图 20: 1）

2. 选择一个包含未分类缩略图的文件夹作为“Sample Set”（图 20: 2）
3. 选择用来存放分类好了的缩略图的目标文件夹（图 20: 3）
4. 点击“Visual Validation”（图 20: 4）

注：在分类过程中，缩略图是被复制而不是剪切到了目标文件夹中。如果你不想保留未分类的缩略图，可以在图 ?? 处取消勾选 *Keep Unsorted Thumbnails*。

Check an existing Validated set

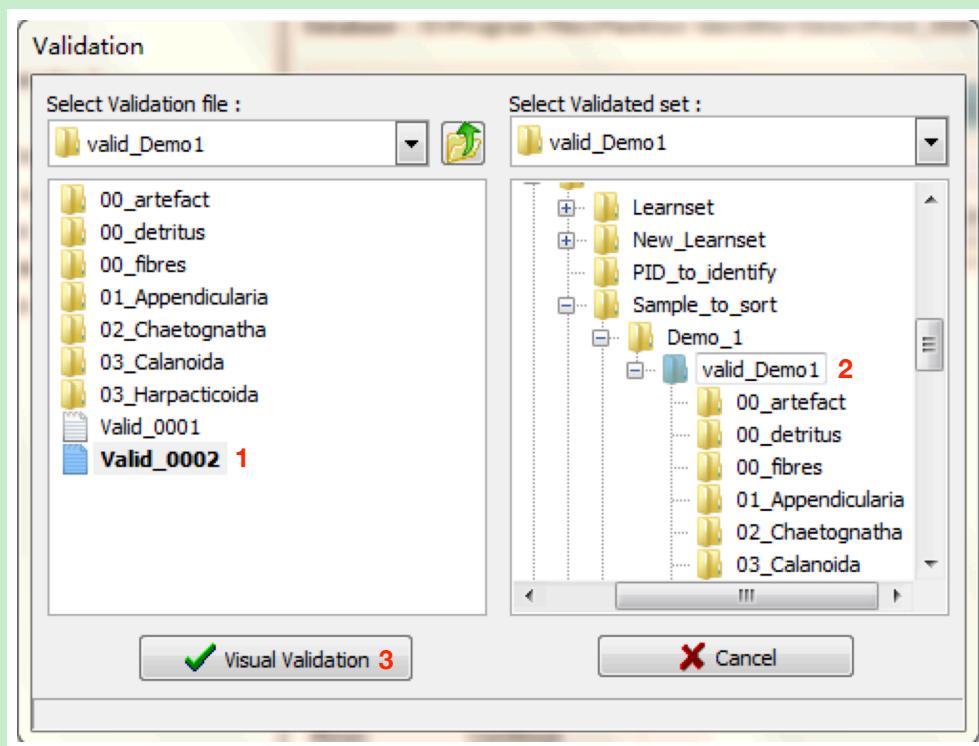


图 21: 校正选择窗口 2

1. 选择你想要检查的 Valid_.txt 文件（图 21: 1）
2. 选择一个包含已分类好的校正后的缩略图的文件夹（图 21: 2）
3. 点击“Visual Validation”按钮（图 21: 3）

Visual Validation

前面的两个选择最终都会打开如图 22 所示的校正窗口：

利用预测模型已经将每幅缩略图放到了其对应的类别文件夹中，你可以打开每个文件夹查看是否被正确分类了。

Thumbnails moving

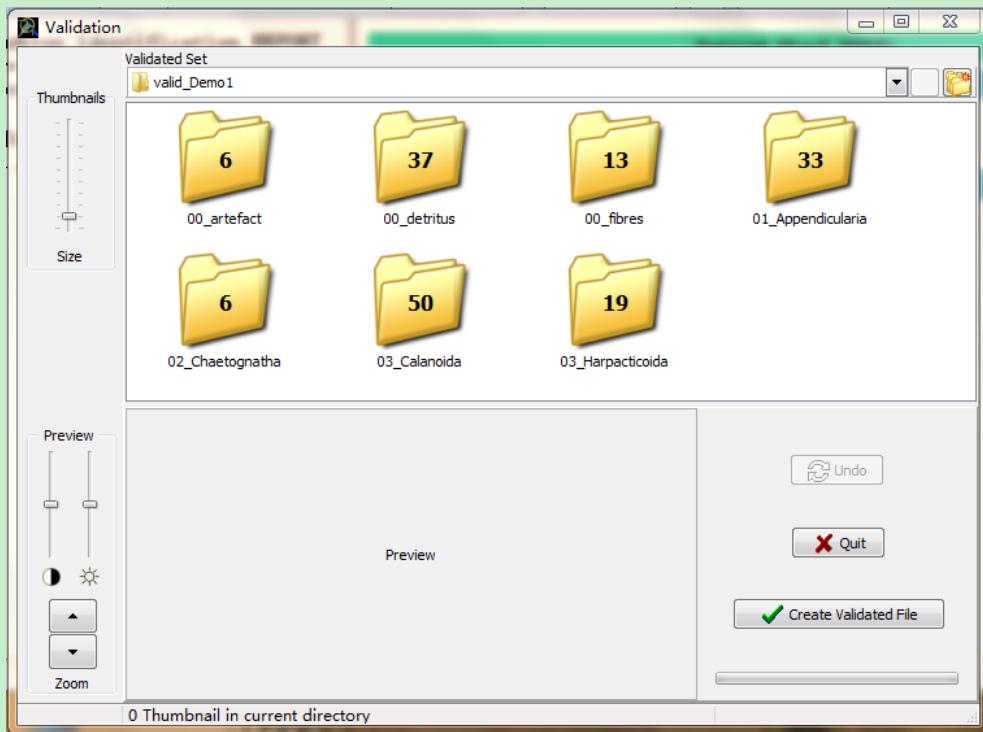


图 22: 校正窗口

3.6 Compilation

这一步是用来将上一步的生成的多个校正文件连起来，并且计算每个类别中的物体数目。在主窗口中点击 **Compilation**，会出现如下窗口（图 23）：

1、Create a concatenation file

1. 浏览硬盘文件夹，找到包含你所要连接的 Valid_.txt 文件的文件夹（图 23: ①）
2. 点击 **Concatenate**（图 23: ②），会出现一个保存会话框
3. 给连接后的文件进行命名，选择将其保存的文件夹路径
4. 点击 **Save** 按钮，连接就开始了（这可能需要几分钟，取决于你想要连接的文件数目）

注：要一次连接很多 Valid_.txt 文件，需要把它们放在同一个文件夹中，然后按“Ctrl”键同时选中它们。

Create a compilation file

1. 选择一个已经连接好的 Concat_.txt 文件。最终的连接文件会以粗体显示（图 24: ①）。

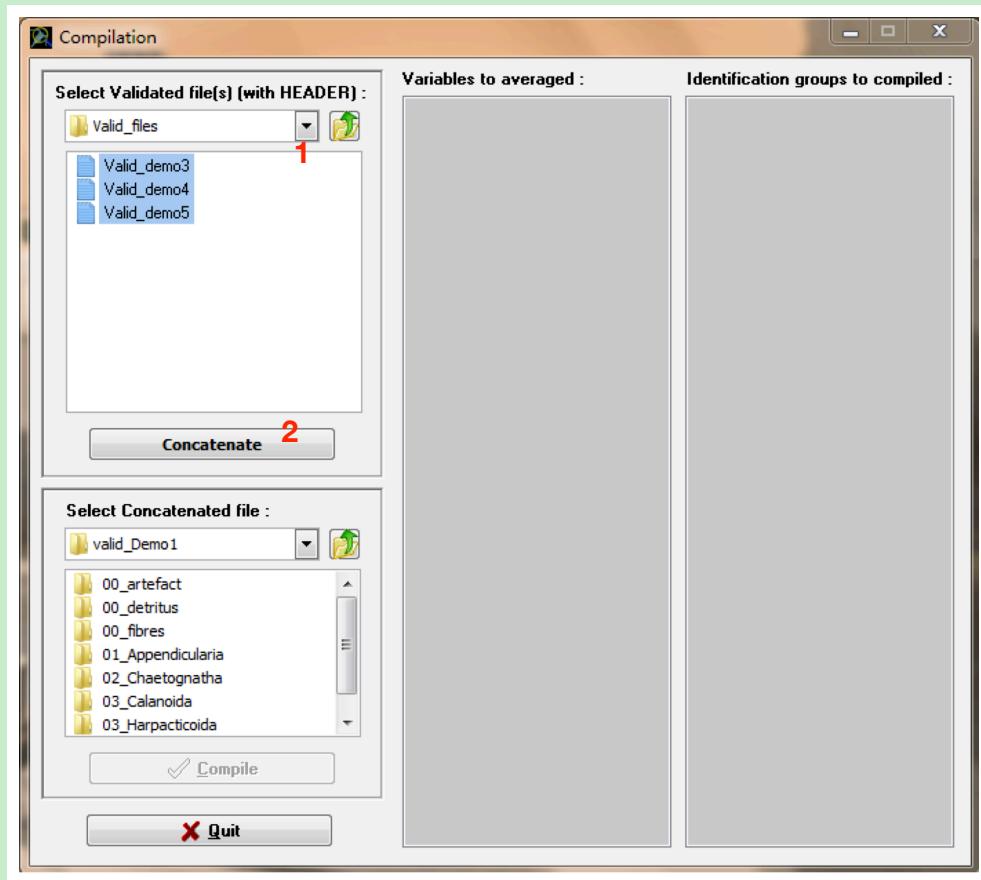


图 23: Compilation 窗口：连接

2. “Variables to averaged”一栏中可以勾选也可以不勾选相应的变量（图 24: 2），被打上勾的变量在编译文件中会被平均，没有打勾的会被去除。
 3. “Identification groups to compiled”一栏也可以挑选相应的类别打上勾（图 24: 3），只有打上勾的类别才会计算出它所包含的物体数目。
 4. 点击 Compile 按钮（图 24: 4），出现一个保存会话框
 5. 给编译后的文件进行命名，选择将其保存的文件夹路径
 6. 点击 Save，编译就开始了
- 编译后生成的文件如图 25。

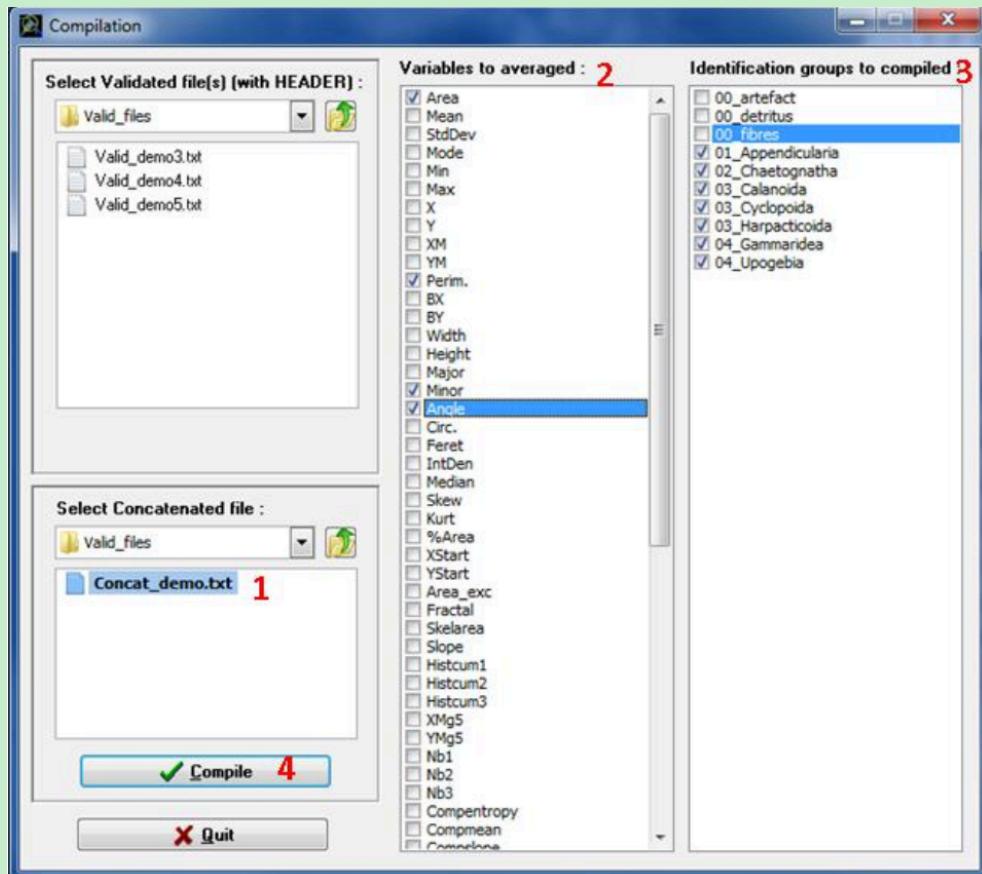


图 24: Compilation 窗口：编译

COMPILED FILE											
Label	GroupName	FracId	SubPart	Count	Area	Perim.	Minor	Angle	ESD	Elongation	
Demo_3	00_detritus	F1		128	55	1936.4	240.064	31.888	90.373	44.96	2.349
Demo_3	03_Harpacticoida	F1		128	11	1284.364	177.813	27.623	88.205	40.362	2.155
Demo_3	03_Calanoida	F1		128	45	2233.889	315.354	37.825	82.176	52.158	1.992
Demo_3	02_Chaetognatha	F1		128	6	5797.167	747.159	31.192	83.094	84.364	7.822
Demo_3	00_fibres	F1		128	10	1789.4	420.543	15.493	65.26	45.885	9.942
Demo_3	03_Cyclopoida	F1		128	1	1544	243.078	28.343	42.376	44.338	2.447
Demo_3	00_artefact	F1		128	6	1263.5	270.969	35.605	74.971	39.386	1.239
Demo_3	01_Appendicularia	F1		128	30	3211.9	468.046	30.148	83.982	62.01	4.562
Demo_4	00_artefact	F1		128	5	1873.6	319.886	38.802	50.847	47.521	1.667
Demo_4	03_Calanoida	F1		128	65	3063.2	348.628	42.982	81.698	61.048	2.052
Demo_4	04_Gammaridea	F1		128	4	7820	583.645	80.585	62.359	97.558	1.47
Demo_4	00_detritus	F1		128	43	2168.86	237.041	37.311	86.007	48.898	2.085
Demo_4	01_Appendicularia	F1		128	26	5943.577	660.8	42.251	76.959	83.938	4.445
Demo_4	03_Harpacticoida	F1		128	4	1370.75	186.223	28.762	102.387	41.719	2.179
Demo_4	04_Upogebia	F1		128	2	12262	906.794	57.27	41.864	124.772	4.771
Demo_4	03_Cyclopoida	F1		128	7	2078.429	252.343	30.891	100.26	50.725	2.69
Demo_4	00_fibres	F1		128	3	1744.667	518.907	11.963	85.451	47.033	18.579
Demo_5	00_detritus	F1		256	48	1575.125	202.747	31.077	85.253	42.98	2.194
Demo_5	00_fibres	F1		256	15	1171.333	383.601	14.173	99.428	37.625	9.063
Demo_5	00_artefact	F1		256	11	1304.182	353.829	29.674	62.284	39.575	2.066
Demo_5	01_Appendicularia	F1		256	13	3227.846	471.515	32.683	73.519	63.513	4.298
Demo_5	03_Harpacticoida	F1		256	13	1535.308	215.014	30.055	102.575	44.162	2.188
Demo_5	03_Calanoida	F1		256	41	2562.976	351.819	39.317	90.851	55.844	2.089
Demo_5	03_Cyclopoida	F1		256	2	2413	270.877	35.866	44.807	54.734	2.334

图 25: 编译文件

4. 评价方法

4.1 论文中采用的评价方法

文中采用混淆矩阵 (CM) (混淆矩阵介绍见 4.4) 对分类器的分类效果进行评价。评价时计算的
戴嘉伦 戚斌 朱亚菲 王如晨 CVBIQUC 是 CM 的召回率 (recall 即 the rate of true positives) 和 1-precision (low contamination 即 the rate of
· 23 · 23

false positives)。

论文中的具体介绍：

Evaluation of classifier performance requires the examination of a [Confusion matrix \(CM\)](#), which is a contingency table crossing true (manually validated) and predicted (assigned by the classifier) identification of objects. Correct interpretation of the CM requires the examination of each category separately, including the [rate of true positives](#) as well as [false positives](#).

- the rate of true positives (recall)

$$\text{true positives} = \frac{\text{number of objects correctly predicted}}{\text{total number actual objects}}$$

- the rate of false positives (low contamination)

$$\text{false positives} = \frac{\text{number of objects falsely assigned to a category}}{\text{total number of predicted objects}}$$

论文中的评价结果如图 26：

Table II: Confusion matrix for the 20 categories in the learning set used for machine classification of the 2007–2008 time series by the Random Forest algorithm

浮游生物的种类 Percentage in learning set	实际每种浮游生物的数量																							
	Aggregates	Aggregates_dark	Aperpendicularia	Bad focus	Bubbles	Chaetognatha	Cladocera	Copepoda_other	Oithona	Copepoda_small	Decapoda_large	Egg-like	Fibers	Medusae	Nectophores	Thaliacea	Limacina	Scratch	Pteropoda_other	Radiolaria	Total	Recall	1-Precision	
Aggregates	7.2	683	27	80	67	0	0	77	99	53	134	8	16	26	9	37	1	1	12	0	25	1355	0.50	0.42
Aggregates_dark	5.7	3	671	0	0	39	0	18	38	0	4	0	27	0	0	0	0	0	0	8	43	1064	0.63	0.40
Aperpendicularia	7.7	85	0	1243	0	0	22	0	2	2	0	5	0	73	0	18	1	0	0	2	2	1455	0.85	0.20
Bad focus	7.5	44	0	5	1230	0	0	5	27	2	12	0	1	5	0	47	0	0	20	0	12	1410	0.87	0.10
Bubbles	4.5	2	30	0	0	786	0	2	0	0	0	0	7	0	0	0	0	10	0	0	3	840	0.94	0.06
Chaetognatha	2.3	0	0	51	0	0	356	0	0	0	0	1	0	20	0	0	0	0	0	0	0	428	0.83	0.11
Cladocera	6.2	27	19	2	1	0	0	1028	22	0	2	0	5	0	0	1	0	1	0	0	62	1170	0.88	0.16
Copepoda_other	11.3	74	32	4	8	0	0	14	1608	119	237	24	1	0	0	4	0	0	6	2	2133	0.75	0.24	
Oithona	7.5	24	0	28	0	0	0	0	61	1256	38	0	0	9	0	0	0	0	0	4	0	1420	0.88	0.17
Copepoda_small	7.8	130	3	0	6	0	0	3	139	43	1141	0	0	0	0	0	0	0	0	0	0	1465	0.78	0.28
Decapoda_large	2.4	2	0	7	0	0	0	0	23	0	0	410	0	0	0	1	0	0	0	2	0	445	0.92	0.12
Egg-like	1.9	12	26	3	7	4	0	12	3	0	0	0	259	0	1	0	0	8	0	0	15	350	0.74	0.20
Fibers	6.4	10	0	84	0	0	21	0	7	14	0	0	0	1034	0	0	0	0	12	28	0	1210	0.85	0.13
Medusae	0.7	16	0	0	2	0	0	0	2	0	0	0	2	0	100	18	0	0	0	0	0	140	0.71	0.19
Nectophores	6.1	26	0	21	32	0	1	3	5	0	0	0	1	1	10	1029	20	0	1	0	5	1155	0.89	0.19
Thaliacea	1.4	8	0	5	6	0	2	0	0	0	0	0	0	0	3	108	123	0	0	0	0	255	0.48	0.17
Limacina	4.5	2	267	0	0	8	0	9	0	0	0	4	0	0	0	0	558	0	0	7	855	0.65	0.28	
Scratch	1.6	2	0	0	5	0	0	0	1	0	0	0	0	6	0	2	4	0	0	0	0	305	0.93	0.14
Pteropoda_other	2.3	9	0	2	1	0	0	3	55	23	23	20	0	11	0	0	0	0	0	291	2	440	0.66	0.15
Radiolaria	5.1	23	45	11	8	0	0	45	12	1	3	0	1	0	0	3	0	0	0	800	955	0.84	0.18	
Total	100	1182	1120	1546	1373	837	402	1219	2124	1513	1594	468	324	1185	123	1268	149	774	330	341	978	18850	0.78	0.19

Corresponding correct identifications (in bold) are in the diagonal.

检测到的每种浮游生物的数量

分类正确的数量 true positives rate(TPR)

图 26: Confusion matrix for the 20 categories in the learning set

4.2 怎样得到混淆矩阵

论文中提到三种得到 CM 的方法 (即采用什么训练集和测试集来生成):

1. Re-substitution CM

这个方法是采用的测试集和训练集为同一个数据集。在这个过程中，用产生的分类器对测试集进行分类时，得到的分类结果错误较少甚至可能没有错误，采用 CM 进行评价时就会低估分类器的错误率。

2. Cross-validation CM

这个方法是采用交叉验证 (交叉验证介绍见[4.5](#)) 的方法。在这个过程中，将一个数据集分成 n 个相等的子集，用其中 n-1 个子集来训练产生分类器，用剩下的 1 个子集来进行测试，重复进行 n 次来构建 CM。

3. Uses two equivalent and independent learning files describing the same categories with different objects

这种方法是用两个相等且相互独立的数据集分别作为训练集和测试集，根据测试结果建立 CM。两个相等的数据集即在两个集合中浮游生物的种类相同，但是每种浮游生物中的个体是不同的。

4.3 PkID 中的评价参数

PkID 中通过 5-折交叉验证 (介绍见[4.5.2](#)) 得到混淆矩阵，对分类器进行评价。5-折交叉验证将数据集分为 5 份，轮流将其中 4 份作为训练数据,1 份作为测试数据，进行试验。每次试验都会得出相应的误分类率 (error rate 介绍见[4.5.2](#)) 如图27。

CV error rate	
Range	
MIN	0.2044
MAX	0.2416
Trial Err rate	
1	0.2416
2	0.2159
3	0.2044
4	0.2068
5	0.2088

图 27: 5 次的误分类率

交叉验证得到的分类器总混淆矩阵 (五次分类的累计结果) 和误分类率 (五次分类误分类率的平均值) 如图28。

Error rate			0.2155														
Values prediction			Confusion matrix														
Value	Recall	1-Precision	Appendicularia	Bubble	Chaetognatha	CladoceraPenilia	Copepoda	Decapoda	Doliolida	Egg	Fiber	Gelatinous	Multiple	Nonbio	Pteropoda	Sum	
Appendicularia	0.8051	0.2039	2194	0	111	8	1	19	0	0	57	1	157	177	0	2725	
Bubble	0.8359	0.1049	0	606	0	0	0	0	0	29	0	0	3	87	0	725	
Chaetognatha	0.8934	0.0983	137	0	1559	0	5	2	0	0	7	1	18	14	2	1745	
CladoceraPenilia	0.8830	0.1044	0	0	0	4084	22	0	7	0	0	7	19	486	0	4625	
Copepoda	0.8419	0.2108	3	0	0	9	6205	145	0	0	0	1	234	745	28	7370	
Decapoda	0.7989	0.2157	4	0	0	0	339	2225	0	0	0	0	87	126	4	2785	
Doliolida	0.7474	0.1418	0	0	0	2	1	0	1065	0	0	156	7	194	0	1425	
Egg	0.7424	0.1401	0	66	0	13	2	1	1	1277	0	28	3	320	9	1720	
Fiber	0.7414	0.2069	59	0	18	0	0	0	0	0	1127	0	62	254	0	1520	
Gelatinous	0.6914	0.2042	6	0	0	72	10	5	93	39	0	2081	79	624	1	3010	
Multiple	0.3238	0.5093	267	0	23	51	491	202	9	23	45	91	1028	941	4	3175	
Nonbio	0.8406	0.2452	86	5	17	321	716	189	66	117	185	246	390	12937	115	15390	
Pteropoda	0.6627	0.1848	0	0	1	0	70	49	0	0	0	3	8	235	719	1085	
Sum			2756	677	1729	4560	7862	2837	1241	1485	1421	2615	2095	17140	882	47300	

图 28: 交叉验证得到的总混淆矩阵和误分类率

4.4 混淆矩阵 Confusion matrix (CM)

在机器学习中，混淆矩阵 (CM) 是一种比较简单的对学习算法性能进行评价的评估准则，而且是大多数指标的基础。常用的算法评估准则有：Confusion Matrix、ROC、Lift、Gini、K-S 等等。

混淆矩阵 (CM)¹²：混淆矩阵是一种评估分类器可信度的方法。在图像精度评价中，主要用于比较分类结果和实际测得值，可以把分类结果的精度显示在一个混淆矩阵里面。特别用于监督学习，在无监督学习一般叫做匹配矩阵。

混淆矩阵是一个 n 行 n 列的矩阵，n 代表类别的数量。矩阵的每一列代表预测的每一类的数量，每一行代表实际的每一类的数量。对角线上表示分类正确的每一类的数量。

例如：有 150 个样本数据，这些数据实际分为 3 类，每类 50 个。分类结束后得到的混淆矩阵如图29。例如：第一行说明类 1 的 50 个样本有 43 个样本分类正确，5 个错分为类 2，2 个错分为类 3；第一列说明类 1 有 43 个样本分类正确，类 2 的 2 个样本被错分为类 1，类 3 没有样本被错分为类 1；对角线上的数据表示，类 1、2、3 分别有 43、45、49 个样本被分类正确。

根据混淆矩阵可以导出以下几个参数³：

- true positives (TP)：正样本被识别出的数量
- true negatives (TN)：负样本被识别出的数量
- false positives (FP)：负样本被错误分为正样本的数量

¹<http://baike.baidu.com/view/2781781.htm>

²http://en.wikipedia.org/wiki/Confusion_matrix

³http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html

	类1	类2	类3
类1	43	5	2
类2	2	45	3
类3	0	1	49

图 29: 混淆矩阵例子

- false negatives (FN): 正样本被错误分为负样本的数量
- Accuracy: 准确率, 针对分类器的整个预测情况。

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Error rate: 误分类率, 针对分类器的整体预测情况。 (PkID 中使用的评价参数)

$$\text{Errorrate} = \frac{FP + FN}{TP + TN + FP + FN}$$

- The true positive rate (TPR) : 召回率, 就是正样本被识别出的概率。 (文中和 PkID 中使用的评价参数)

$$TPR = \frac{TP}{TP + FN}$$

- The false positive rate (FPR): 虚警率, 负样本被错误分为正样本的概率。

$$FPR = \frac{FP}{FP + TN}$$

- The true negative rate (TNR): 负样本被识别出的概率

$$TNR = \frac{TN}{FP + TN}$$

- The false negative rate (FNR) : 漏警率, 正样本被错误分为负样本的概率。

$$FNR = \frac{FN}{FN + TP}$$

- False discovery rate (FDR): 1-precision。 (文中和 PkID 中使用的评价参数)

$$FDR = \frac{FP}{FP + TP}$$

- Positive predictive value (PPV)

$$PPV = \frac{TP}{TP + FP}$$

- Negative predictive value (NPV)

$$NPV = \frac{TN}{TN + FN}$$

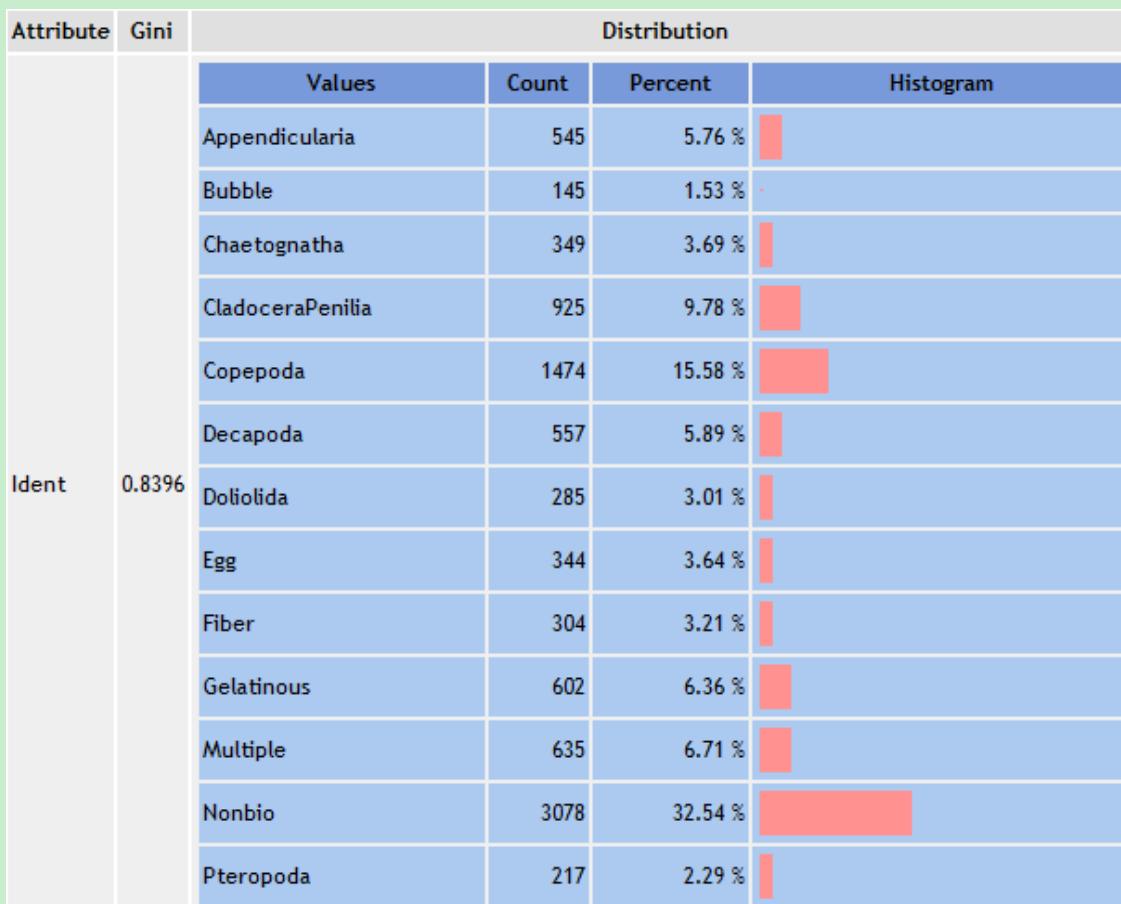


图 30: 数据集中每种类别的数目及所占比例

4.5 交叉验证 Cross Validation (CV)

交叉验证是用来验证分类器的性能一种统计分析方法，基本思想是把在某种意义下将原始数据进行分组，一部分做为训练集 (training set)，另一部分做为验证集 (validation set)，首先用训练集对分类器进行训练，在利用验证集来测试训练得到的模型，以此来做为评价分类器的性能指标。

4.5.1 训练集和测试集

在模式识别与机器学习的相关研究中，经常会将数据集分为训练集跟测试集这两个子集，前者用以建立模型，后者则用来评估该模型对未知样本进行预测时的精确度，正规的说法是泛化能力。怎么将完整的数据集分为训练集跟测试集，必须遵守如下要点：

1. 只有训练集才可以在模型的训练过程中，测试集则必须在模型完成之后才被用来评估模型优劣的依据。
2. 训练集中样本数量必须够多，一般至少大于总样本数的 50%。
3. 两组子集必须从完整集合中均匀取样。

注：其中最后一点特别重要，均匀取样的目的是希望减少训练集和测试集与完整集合之间的偏差，但却也不易做到。一般的作法是随机取样，当样本数量足够时，便可达到均匀取样的效果，然而随机也正是此作法的盲点，也是经常是在数据上做手脚的地方。举例来说，当辨识率不理想时，便重新取样一组训练集和测试集，直到测试集的识别率满意为止，但严格来说这样便算是作弊了。

在 MATALB 中使用 `cvpartition` 对数据集进行随机拆分，完成交叉验证。

4.5.2 常见的交叉验证方法

- Hold-Out Method

将原始数据随机分为两组，一组做为训练集，一组做为验证集。

- Double Cross Validation (2-fold Cross Validation, 记为 2-CV)

将数据集分成两个相等大小的子集，进行两回合的分类器训练。在第一回合中，一个子集作为 training set，另一个便作为 testing set；在第二回合中，则将 training set 与 testing set 对换后，再次训练分类器。

- K-fold Cross Validation (K-折交叉验证，记为 K-CV) (实验中采用的交叉验证方法) 将原始数据分成 K 组，将每个子集数据分别做一次验证集，其余的 K-1 组子集数据作为训练集，这样会

得到 K 个模型，用这 K 个模型最终的验证集的分类准确率的平均数作为此 K-CV 下分类器的性能指标。K 一般大于等于 2，实际操作时一般从 3 开始取。

- Leave-One-Out Cross Validation（记为 LOO-CV）

将每个样本单独作为验证集，其余的 N-1 个样本作为训练集，所以 LOO-CV 会得到 N 个模型，用这 N 个模型最终的验证集的分类准确率的平均数作为此下 LOO-CV 分类器的性能指标。相比于前面的 K-CV，LOO-CV 有两个明显的优点：

- 每一回合中几乎所有的样本皆用于训练模型，因此最接近原始样本的分布，这样评估所得的结果比较可靠。
- 实验过程中没有随机因素会影响实验数据，确保实验过程是可以被复制的。

5. 特征 (PkID)

PkID 中用到的特征一共有 67 个：Area, Mean, StdDev, Mode, Min, Max, X, Y, XM, YM, Perim., BX, BY, Width, Height, Major, Minor, Angle, Circ., Feret, IntDen, Median, Skew, Kurt, %Area, XStart, YStart, Area_exc, Fractal, Skelarea, Slope, Histcum1, Histcum2, Histcum3, XMg5, YMg5, Nb1, Nb2, Nb3, Compentropy, Compmean, Compslope, CompM1, CompM2, CompM3, Symetrieh, Symetriev, Symetriehc, Symetrievc, Converperim, Convarea, Fcons, ThickR, Tag, ESD, Elongation, Range, MeanPos, CentroidsD, CV, SR, PerimAreaexc, FeretAreaexc, PerimFeret, PerimMaj, Circexc, CDexc

从训练集的 PID 文件文件中看到，Compentropy, Compmean, Compslope, CompM1, CompM2, CompM3 这 6 个特征在所有图像上的值都为 0，在训练分类器时是不起作用的。这 6 个特征的具体含义也没有找到。

5.1 位置特征

BX 能够包围物体，且平行于图像两条边的最小外界矩形的左上角顶点的 X 坐标

BY 能够包围物体，且平行于图像两条边的最小外界矩形的左上角顶点的 Y 坐标

Height 能够包围物体，且平行于图像两条边的最小外界矩形的高

Width 能够包围物体，且平行于图像两条边的最小外界矩形的宽

XStart 图像最左上角像素点的 X 坐标

YStart 图像最左上角像素点的 Y 坐标

XM 物体灰度重心的 X 坐标

YM 物体灰度重心的 Y 坐标

XMg5 gamma 值为 51 时的物体灰度重心的 X 坐标

YMg5 gamma 值为 51 时的物体灰度重心的 Y 坐标

X 物体重心点的 X 坐标

Y 物体重心点的 Y 坐标

5.2 尺寸特征

Area 包含物体的最小矩形面积

Perim 物体最外层边缘的长度

Major 物体内切椭圆的长轴

Minor 物体内切椭圆的短轴

Feret Maximum feret diameter (最大费雷特径)，沿物体边缘任意两个点的最长距离

Area_exc 去掉物体空洞后的表面积，空洞是指灰度值与背景相同的部分

%area 物体表面积中空洞所占的百分比，即背景所占的比例

5.3 灰度值特征

Min 物体内部所有像素点的最小灰度值 (0 = black)

Max 物体内部所有像素点的最大灰度值 (255 = white)

Mean 物体内的平均灰度值；物体中所有像素点的灰度值的总和除以总的像素个数

IntDen Integrated density (总密度)。物体内像素点的灰度值的总和 ($IntDen = Area * Mean$)

StdDev 物体内像素的灰度值的标准差

Mode Modal grey value within the object

Skew 灰度直方图的偏斜度，是灰度分布是否符合正态分布的检验参数

Kurt 灰度直方图的峰值

Mean_exc 物体内部去掉空洞后的平均灰度值 ($Mean_exc = IntDen/Area_exc$)

Median 物体内像素的灰度值的中位数

Slope 归一化的灰度累计直方图的斜率

Histcum1 灰度累计直方图的值为 25% 时所对应的灰度值

Histcum2 灰度累计直方图的值为 50% 时所对应的灰度值

Histcum3 灰度累计直方图的值为 75% 时所对应的灰度值

5.4 形状特征

Fractal 物体边界的分形维数 (Berube and Jebrak, 1999), 表明物体边界的不规则程度

Skelarea 骨架像素的表面积 (在二值图像中, 不断地从物体边缘处减去像素点直到仅剩一个像素的宽度, 最后所得图形的像素点数)

Circ $Circularity = (4 * \pi * Area) / Perim^2$; 表征物体接近圆的程度, 值等于 1 时, 说明物体为正圆形, 值越接近 0, 物体体形越长。

Angle 浮游动物主轴与图片 x 轴形成的夹角, 在图片切割后旋转图片测量相关参数使用

Symetrieh 水平对称

Symmetriev 垂直对称

Symmetriehc

Symmetrievc

5.5 自定义特征

$$\text{ESD} \quad 2 \times \sqrt{\frac{\text{Area}}{\pi}}$$

$$\text{Elongation} \quad \frac{\text{Major}}{\text{Minor}}$$

$$\text{Range} \quad \text{Max} - \text{Min}$$

$$\text{MeanPos} \quad \frac{\text{Mean} - \text{Max}}{\text{Max} - \text{Min}}$$

$$\text{CentrodisD} \quad \sqrt{(XM - X)^2 + (YM - Y)^2}$$

$$\text{CV} \quad 100 \times \frac{\text{StdDev}}{\text{Mean}}$$

$$\text{SR} \quad 100 \times \frac{\text{StdDev}}{\text{Max} - \text{Min}}$$

$$\text{PerimAreaexc} \quad \frac{\text{Perim}}{\sqrt{\text{Area_exc}}}$$

$$\text{FeretAreaexc} \quad \frac{\text{Feret}}{\sqrt{\text{Area_exc}}}$$

$$\text{PerimFeret} \quad \frac{\text{Perim}}{\text{Feret}}$$

$$\text{PerimMaj} \quad \frac{\text{Perim}}{\text{Major}}$$

$$\text{Circexc} \quad \frac{4 \times \pi \text{Area_exc}}{\text{Perim}^2}$$

$$\text{CDexc} \quad \frac{\sqrt{(XM - X)^2 + (YM - Y)^2}}{\sqrt{\text{Area_exc}}}$$

6. 识别结果分析

采用交叉验证的方法来评价分类器的性能好坏，所选用的数据集中的类别如图 30 所示，其中有 10 个是浮游动物类，包括 Appendicularia（被囊类）、Chaetognatha（毛颚类）、CladoceraPenilia（枝角

类喙)、Copepoda (桡足类)、Decapoda (十足类)、Doliolida (海樽类)、Egg (蛋类)、Pteropoda (翼足类)、Gelatinous (凝胶纤维)、Multiple。3 个非浮游动物类: Nonbio (非生物类)、Bubble (气泡)、Fiber (纤维)。将该数据集随机均分为 n 份, 其中每个子集数据分别做一次验证集, 其余的 $n - 1$ 组子集数据作为训练集, 这样会得到 n 个模型, 用这 n 个模型最终的验证集的分类准确率的平均数作为此分类器的性能指标。

采用随机森林学习算法, 67 个特征的评价结果如图 31。

Error rate			0.2155														
Values prediction			Confusion matrix														
Value	Recall	1-Precision		Appendicularia	Bubble	Chaetognatha	CladoceraPenilia	Copepoda	Decapoda	Doliolida	Egg	Fiber	Gelatinous	Multiple	Nonbio	Pteropoda	Sum
Appendicularia	0.8051	0.2039	Appendicularia	2194	0	111	8	1	19	0	0	57	1	157	177	0	2725
Bubble	0.8359	0.1049	Bubble	0	606	0	0	0	0	0	29	0	0	3	87	0	725
Chaetognatha	0.8934	0.0983	Chaetognatha	137	0	1559	0	5	2	0	0	7	1	18	14	2	1745
CladoceraPenilia	0.8830	0.1044	CladoceraPenilia	0	0	0	4084	22	0	7	0	0	7	19	486	0	4625
Copepoda	0.8419	0.2108	Copepoda	3	0	0	9	6205	145	0	0	0	1	234	745	28	7370
Decapoda	0.7989	0.2157	Decapoda	4	0	0	0	339	2225	0	0	0	0	87	126	4	2785
Doliolida	0.7474	0.1418	Doliolida	0	0	0	2	1	0	1065	0	0	156	7	194	0	1425
Egg	0.7424	0.1401	Egg	0	66	0	13	2	1	1	1277	0	28	3	320	9	1720
Fiber	0.7414	0.2069	Fiber	59	0	18	0	0	0	0	0	1127	0	62	254	0	1520
Gelatinous	0.6914	0.2042	Gelatinous	6	0	0	72	10	5	93	39	0	2081	79	624	1	3010
Multiple	0.3238	0.5093	Multiple	267	0	23	51	491	202	9	23	45	91	1028	941	4	3175
Nonbio	0.8406	0.2452	Nonbio	86	5	17	321	716	189	66	117	185	246	390	12937	115	15390
Pteropoda	0.6627	0.1848	Pteropoda	0	0	1	0	70	49	0	0	0	3	8	235	719	1085
			Sum	2756	677	1729	4560	7862	2837	1241	1485	1421	2615	2095	17140	882	47300

图 31: Random Forest

采用随机森林算法, 去掉以下位置特征 (XM, YM, BX, BY, Width, Height, Angle, XStart, YStart, XMg5, YMg5, X, Y) 的评价结果如图 32, 可以看到, 与用所有特征相比, 去除不必要的位置特征会使评价结果有所提高, 对所有种类的 Recall 值、Precision 值求平均 (除了 Multi 类), 得到的平均的 Recall 值为 0.799, 平均的 Precision 值为 0.161。PID 文件中的位置特征大多是 ImageJ 自带的, 是用来生成缩略图所需的参数, 在训练分类器时并没有使用。

采用 K-NN 算法, 去掉位置特征后的评价结果如图 33。

采用 C-SVC linear 算法, 去掉位置特征后的评价结果如图 34。

采用 C-SVC RBF 算法, 去掉位置特征后的评价结果如图 35。

采用 BVM RBF 算法, 去掉位置特征后的评价结果如图 36。

采用 PLS 算法, 去掉位置特征后的评价结果如图 37。

采用 Multilayer perceptron 算法, 去掉位置特征后的评价结果如图 38。

Error rate			0.2074														
Values prediction			Confusion matrix														
Value	Recall	1-Precision	Appendicularia	Bubble	Chaetognatha	CladoceraPenilia	Copepoda	Decapoda	Doliolida	Egg	Fiber	Gelatinous	Multiple	Nonbio	Pteropoda	Sum	
Appendicularia	0.8172	0.2052	2227	0	113	8	0	19	0	0	32	0	171	155	0	2725	
Bubble	0.8455	0.0959		613	0	0	0	3	0	0	37	0	0	2	70	0	725
Chaetognatha	0.8991	0.0957	132	0	1569	0	1	2	0	0	4	1	21	14	1	1745	
CladoceraPenilia	0.8902	0.0918	0	0	0	4117	23	0	4	0	0	4	18	459	0	4625	
Copepoda	0.8503	0.2087	3	0	0	9	6267	129	0	0	0	4	233	697	28	7370	
Decapoda	0.8014	0.2130	2	0	0	0	309	2232	0	0	0	0	100	139	3	2785	
Doliolida	0.7761	0.1236	0	0	0	1	0	0	1106	0	0	148	9	161	0	1425	
Egg	0.7709	0.1004	0	54	0	11	6	0	0	1326	0	23	6	287	7	1720	
Fiber	0.7336	0.2019	48	0	16	0	1	0	1	0	1115	1	64	274	0	1520	
Gelatinous	0.7106	0.1962	6	0	0	58	10	4	71	20	0	2139	111	587	4	3010	
Multiple	0.3320	0.5154	286	1	21	48	500	211	9	10	48	106	1054	879	2	3175	
Nonbio	0.8462	0.2332	98	10	15	281	731	180	71	80	198	230	377	13023	96	15390	
Pteropoda	0.6470	0.1673	0	0	1	0	69	59	0	1	0	5	9	239	702	1085	
			Sum	2802	678	1735	4533	7920	2836	1262	1474	1397	2661	2175	16984	843	47300

图 32: Random Forest (without position variables)

Error rate			0.9424															
Values prediction			Confusion matrix															
Value	Recall	1-Precision	Appendicularia	Bubble	Chaetognatha	CladoceraPenilia	Copepoda	Decapoda	Doliolida	Egg	Fiber	Gelatinous	Multiple	Nonbio	Pteropoda	Sum		
Appendicularia	1.0000	0.9424	2725	0	0	0	0	0	0	0	0	0	0	0	0	0	2725	
Bubble	0.0000	1.0000	725	0	0	0	0	0	0	0	0	0	0	0	0	0	725	
Chaetognatha	0.0000	1.0000	1745	0	0	0	0	0	0	0	0	0	0	0	0	0	1745	
CladoceraPenilia	0.0000	1.0000	4625	0	0	0	0	0	0	0	0	0	0	0	0	0	4625	
Copepoda	0.0000	1.0000	7370	0	0	0	0	0	0	0	0	0	0	0	0	0	7370	
Decapoda	0.0000	1.0000	2785	0	0	0	0	0	0	0	0	0	0	0	0	0	2785	
Doliolida	0.0000	1.0000	1425	0	0	0	0	0	0	0	0	0	0	0	0	0	1425	
Egg	0.0000	1.0000	1720	0	0	0	0	0	0	0	0	0	0	0	0	0	1720	
Fiber	0.0000	1.0000	1520	0	0	0	0	0	0	0	0	0	0	0	0	0	1520	
Gelatinous	0.0000	1.0000	3010	0	0	0	0	0	0	0	0	0	0	0	0	0	3010	
Multiple	0.0000	1.0000	3175	0	0	0	0	0	0	0	0	0	0	0	0	0	3175	
Nonbio	0.0000	1.0000	15390	0	0	0	0	0	0	0	0	0	0	0	0	0	15390	
Pteropoda	0.0000	1.0000	1085	0	0	0	0	0	0	0	0	0	0	0	0	0	1085	
			Sum	47300	0	0	0	0	0	0	0	0	0	0	0	0	0	47300

图 33: K-NN (without position variables)

Error rate			0.2587														
Values prediction			Confusion matrix														
Value	Recall	1-Precision	Appendicularia	Bubble	Chaetognatha	CladoceraPenilia	Copepoda	Decapoda	Doliolida	Egg	Fiber	Gelatinous	Multiple	Nonbio	Pteropoda	Sum	
Appendicularia	0.8411	0.1921	2292	0	86	1	11	11	0	1	41	4	123	155	0	2725	
Bubble	0.8400	0.2793	0	609	0	1	19	0	0	23	0	0	2	71	0	725	
Chaetognatha	0.8940	0.0856	130	0	1560	0	0	9	0	0	16	0	17	13	0	1745	
CladoceraPenilia	0.7968	0.2031	0	0	0	3685	19	0	5	0	0	31	5	880	0	4625	
Copepoda	0.8001	0.2891	0	0	0	17	5897	187	0	0	1	0	182	1007	79	7370	
Decapoda	0.6700	0.2586	1	0	0	0	496	1866	0	0	0	0	94	318	10	2785	
Doliolida	0.7530	0.1721	0	0	0	7	0	0	1073	0	0	128	3	214	0	1425	
Egg	0.6459	0.1513	0	155	0	24	29	1	0	1111	0	65	0	324	11	1720	
Fiber	0.7362	0.2197	55	0	22	0	0	0	0	8	1119	0	44	272	0	1520	
Gelatinous	0.6970	0.2175	5	0	0	112	19	10	130	30	1	2098	63	540	2	3010	
Multiple	0.2995	0.4584	244	0	21	115	589	176	13	9	52	83	951	909	13	3175	
Nonbio	0.8014	0.2880	107	81	16	662	969	192	75	118	204	272	266	12334	94	15390	
Pteropoda	0.4323	0.3083	3	0	1	0	247	65	0	9	0	0	6	285	469	1085	
			Sum	2837	845	1706	4624	8295	2517	1296	1309	1434	2681	1756	17322	678	47300

图 34: C-SVC linear (without position variables)

Error rate			0.2276														
Values prediction			Confusion matrix														
Value	Recall	1-Precision	Appendicularia	Bubble	Chaetognatha	CladoceraPenilia	Copepoda	Decapoda	Doliolida	Egg	Fiber	Gelatinous	Multiple	Nonbio	Pteropoda	Sum	
Appendicularia	0.8488	0.1927	2313	0	81	0	0	7	0	0	50	0	133	141	0	2725	
Bubble	0.8138	0.1884		590	0	0	0	10	0	0	13	0	0	12	100	0	725
Chaetognatha	0.8613	0.0807	130	0	1503	0	1	1	0	0	5	0	23	82	0	1745	
CladoceraPenilia	0.8971	0.1369		0	0	0	4149	37	0	10	0	0	14	12	403	0	4625
Copepoda	0.8404	0.2508		0	0	0	14	6194	166	0	0	1	1	191	755	48	7370
Decapoda	0.7461	0.2170		0	0	0	0	392	2078	0	0	0	0	96	216	3	2785
Doliolida	0.8007	0.1434		0	0	0	0	0	0	1141	0	0	122	9	153	0	1425
Egg	0.6663	0.1377		0	102	0	10	33	1	2	1146	0	57	5	337	27	1720
Fiber	0.7007	0.2064		61	0	19	0	0	0	0	4	1065	0	41	330	0	1520
Gelatinous	0.7110	0.1766		5	0	1	74	12	2	95	30	5	2140	58	588	0	3010
Multiple	0.3049	0.4929		253	2	16	64	568	182	9	13	36	57	968	993	14	3175
Nonbio	0.8242	0.2572		103	33	15	496	851	182	75	110	180	205	354	12685	101	15390
Pteropoda	0.5189	0.2553		0	0	0	0	170	35	0	13	0	3	7	294	563	1085
			Sum	2865	727	1635	4807	8268	2654	1332	1329	1342	2599	1909	17077	756	47300

图 35: C-SVC RBF (without position variables)

Error rate			0.2304														
Values prediction			Confusion matrix														
Value	Recall	1-Precision	Appendicularia	Bubble	Chaetognatha	CladoceraPenilia	Copepoda	Decapoda	Doliolida	Egg	Fiber	Gelatinous	Multiple	Nonbio	Pteropoda	Sum	
Appendicularia	0.8334	0.1857	2271	0	77	0	0	8	0	0	50	0	149	170	0	2725	
Bubble	0.8028	0.1698		0	582	0	0	8	0	0	21	0	0	10	104	0	725
Chaetognatha	0.8407	0.0768	125	0	1467	0	1	0	0	0	9	0	26	117	0	1745	
CladoceraPenilia	0.8951	0.1257		0	0	0	4140	31	0	10	0	0	19	15	410	0	4625
Copepoda	0.8353	0.2401		0	0	0	11	6156	184	0	0	1	3	193	783	39	7370
Decapoda	0.7494	0.2207		0	0	0	0	352	2087	0	0	0	0	114	228	4	2785
Doliolida	0.7881	0.1342		0	0	0	1	0	0	1123	0	0	133	6	162	0	1425
Egg	0.6645	0.1387		0	85	0	10	32	2	0	1143	0	56	14	347	31	1720
Fiber	0.6796	0.2162		57	0	19	0	1	0	0	0	1033	0	30	380	0	1520
Gelatinous	0.7043	0.1821		5	0	0	76	15	1	84	32	5	2120	56	616	0	3010
Multiple	0.2986	0.5148		237	1	12	65	529	187	7	14	35	51	948	1079	10	3175
Nonbio	0.8300	0.2690		94	33	14	432	818	168	73	105	185	208	383	12774	103	15390
Pteropoda	0.5143	0.2510		0	0	0	0	158	41	0	12	0	2	10	304	558	1085
			Sum	2789	701	1589	4735	8101	2678	1297	1327	1318	2592	1954	17474	745	47300

图 36: BVM RBF (without position variables)

Error rate			0.6159														
Values prediction			Confusion matrix														
Value	Recall	1-Precision	Appendicularia	Bubble	Chaetognatha	CladoceraPenilia	Copepoda	Decapoda	Doliolida	Egg	Fiber	Gelatinous	Multiple	Nonbio	Pteropoda	Sum	
Appendicularia	0.1402	0.4299	382	0	692	16	121	29	0	0	8	2	165	1310	0	2725	
Bubble	0.0000	1.0000		0	0	0	0	324	0	0	0	0	0	0	401	0	725
Chaetognatha	0.7840	0.4309	52	0	1368	0	6	3	0	0	24	23	190	79	0	1745	
CladoceraPenilia	0.0357	0.7519		0	0	0	165	45	0	0	0	0	0	0	4415	0	4625
Copepoda	0.3512	0.7135	7	0	2	3	2588	4	0	0	0	3	45	4718	0	7370	
Decapoda	0.0251	0.7222		0	0	5	0	2075	70	0	0	0	6	141	488	0	2785
Doliolida	0.0035	0.9254		0	0	0	21	0	0	5	0	0	78	0	1321	0	1425
Egg	0.0012	0.5000		0	0	0	5	440	0	0	2	0	5	0	1268	0	1720
Fiber	0.0329	0.4444		101	0	179	3	1	0	0	0	50	3	7	1176	0	1520
Gelatinous	0.0635	0.6987		0	0	2	202	75	7	25	2	0	191	24	2482	0	3010
Multiple	0.0542	0.7882		98	0	111	51	876	94	25	0	1	240	172	1507	0	3175
Nonbio	0.8559	0.5968		30	0	32	199	1776	19	12	0	7	82	60	13173	0	15390
Pteropoda	0.0000	1.0000		0	0	13	0	706	26	0	0	0	1	8	331	0	1085
			Sum	670	0	2404	665	9033	252	67	4	90	634	812	32669	0	47300

图 37: PLS (without position variables)

Error rate			0.2597														
Values prediction			Confusion matrix														
Value	Recall	1-Precision	Appendicularia	Bubble	Chaetognatha	CladoceraPenilia	Copepoda	Decapoda	Doliolida	Egg	Fiber	Gelatinous	Multiple	Nonbio	Pteropoda	Sum	
Appendicularia	0.7930	0.2334	2161	0	147	2	1	11	2	1	71	27	156	144	2	2725	
Bubble	0.5338	0.3418		0	387	0	0	12	0	0	99	0	0	9	217	1	725
Chaetognatha	0.8590	0.1644		131	0	1499	0	1	12	3	0	35	24	23	16	1	1745
CladoceraPenilia	0.9174	0.1295		1	0	0	4243	32	0	3	10	0	26	28	282	0	4625
Copepoda	0.8168	0.2393		4	6	1	23	6020	228	1	26	4	3	271	693	90	7370
Decapoda	0.7311	0.2801		15	2	3	2	338	2036	0	17	1	5	132	174	60	2785
Doliolida	0.7642	0.2929		1	0	0	3	0	0	1089	1	2	149	16	164	0	1425
Egg	0.6645	0.3808		0	104	0	11	14	2	3	1143	3	93	6	328	13	1720
Fiber	0.7441	0.3125		32	2	35	0	1	1	6	1	1131	3	42	265	1	1520
Gelatinous	0.6940	0.2821		17	6	11	60	11	5	182	52	5	2089	97	466	9	3010
Multiple	0.3020	0.5540		282	12	47	82	450	229	45	48	101	105	959	796	19	3175
Nonbio	0.7693	0.2427		170	66	45	448	846	199	206	357	291	379	401	11839	143	15390
Pteropoda	0.3871	0.4466		5	3	6	0	188	105	0	91	1	7	10	249	420	1085
			Sum	2819	588	1794	4874	7914	2828	1540	1846	1645	2910	2150	15633	759	47300

图 38: MultilayerPerceptron (without position variables)

7. 优缺点分析

PkID 主要用了灰度特征、尺寸特征和形状特征。其中灰度特征跟灰度图的质量有很大关系，并不能完全准确地将不同种类区分开来，这里可以尝试加入一些灰度对比度特征。对于尺寸特征，每幅图像中都有一个 5mm 的标尺，可以依据标尺来度量物体的一些长度特征。人在进行浮游动物分类时，主要是依据形状特征，所以形状特征应该是最重要的，但 PkID 中只采用了少数简单的形状特征，因而不能充分地表示某一类浮游动物。

对于具体哪些特征是有用的哪些是没用的，还得通过具体的实验才能看出来。单从缩略图来看，这三类特征都是有用的，都有区分作用，但是还不充足，还应该再加入一些更合理的特征。

8. 竞赛

8.1 PASCAL VOC Challenges 2005-2012

PASCAL VOC 挑战赛是视觉对象的分类识别和检测的一个基准测试，提供了检测算法和学习性能的标准图像注释数据集和标准的评估系统。

挑战赛主要分为三个部分：图像的分类、识别、分割，另外还有一个‘动态’分类项目，一个由 Image Net 举行的大规模识别竞赛和人类身体部位识别的附加赛项目。

分类就是让算法找出测试图片都是属于哪一个标签，对测试的图片进行分类，将图片对号入座。检测则是检测出测试图片中由委员会特别圈定的内容，看看算法能否正确的符合圈定的内容。分割是对图片进行像素级分割，也就是识别出的特定物体用一种颜色表示，其他的则作为背景。动作分类则是

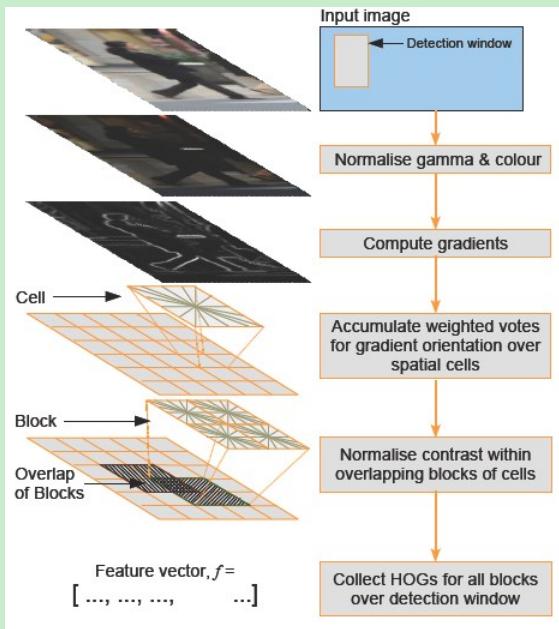


图 39: HOG 特征

在静态图片中预测人类的动作，比如有一张人类跑步的图片，算法根据身体各部位的位置特征判别这个动作是‘running’。人类轮廓识别就是识别标示出来的人体部位，这对于一张图片有几个人或者两个人身体部分纠缠在一起的图片识别有重要意义。

8.2 IMAGENET Large Scale Visual Recognition Challenges 2010-2015

8.3 Labeled Faces in the Wild

9. 特征 (Ours)

9.1 经典特征

9.1.1 HOG 特征

HOG[4] (Histogram of Oriented Gradients, 方向梯度直方图) 是一种在计算机视觉和图像处理中用来进行物体检测的特征描述子，它通过统计图像局部区域的梯度方向直方图来构成特征。HOG 特征结合 SVM 分类器已经被广泛应用于图像识别中，尤其在行人检测中获得了极大的成功。

主要思想：局部目标的表象和形状能够被梯度或边缘的方向密度分布很好地描述。

具体实现方法：将图像分成小的连通区域，我们把它叫细胞单元 (cell)。然后采集细胞单元中各像素点的梯度的或边缘的方向直方图，最后把这些直方图组合起来就可以构成特征描述器。具体实现流程如图39。

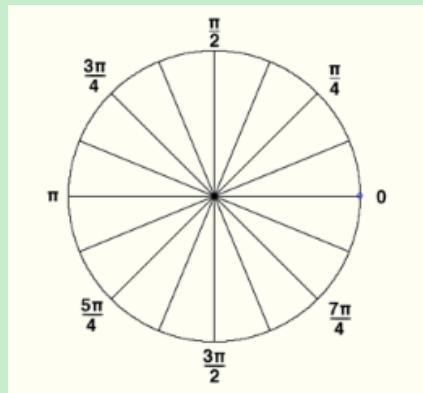


图 40: 梯度方向 bin

算法步骤:

(1) 图像归一化。采用 Gamma 校正法对输入图像进行颜色空间的标准化；目的是调节图像的对比度，降低图像局部的阴影和光照变化所造成的影响，同时可以抑制噪音的干扰。

Gamma 压缩公式：

$$I(x, y) = I(x, y)^{\text{gamma}}$$

比如可以取 $\text{Gamma} = \frac{1}{2}$ 。

(2) 利用一阶微分计算每一个像素的梯度（包括大小和方向）；目的是为了捕获轮廓信息，同时进一步弱化光照的干扰。

采用模板 $[-1 \ 0 \ 1]$ （实验表明采用模板 $[-1 \ 0 \ 1]$ 求得的梯度效果最好）计算水平和垂直方向的梯度：

$$G_h(x, y) = f(x + 1, y) - f(x - 1, y) \quad \forall x, y$$

$$G_v(x, y) = f(x, y + 1) - f(x, y - 1) \quad \forall x, y$$

计算梯度值和梯度方向：

$$M(x, y) = \sqrt{G_h(x, y)^2 + G_v(x, y)^2}$$

$$\theta(x, y) = \arctan \frac{G_h(x, y)}{G_v(x, y)}$$

(3) 将图像划分成小 cells（例如 6×6 像素/cell）。

(4) 统计每个 cell 的梯度直方图，可形成每个 cell 的描述子；

构建每个单元的梯度方向直方图，将 cell 的梯度方向 360 度根据需要分成 m 个 bin，例如图 40 分成了 16 个 bin。然后根据每个像素点的梯度方向，采用加权投票的方式得到直方图，即每一票都是带权值的，这个权值是根据该像素点的梯度幅度计算出来的。

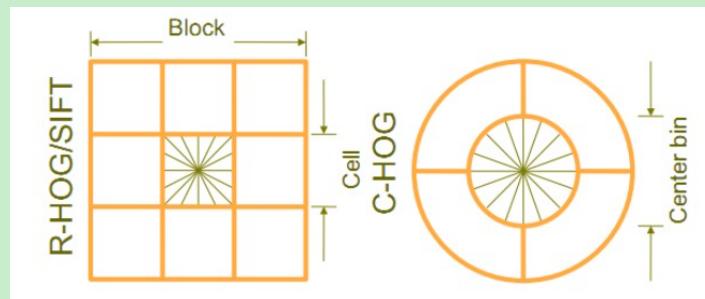


图 41: 矩形区间和环形区间

(5) 将每几个 cell 组成一个块 (block) (例如 3×3 个 cell/block)，一个 block 内所有 cell 的梯度直方图串联起来便得到该 block 的 HOG 特征描述子。

把各个细胞单元组合成大的、空间上连通的区间 (blocks)。这样以来，HOG 描述器就变成了由各区间所有细胞单元的直方图成分所组成的一个向量。这些区间是互有重叠的，这就意味着：每一个细胞单元的输出都多次作用于最终的描述器。区间有两个主要的几何形状——矩形区间 (R-HOG) 和环形区间 (C-HOG) 如图41。

(6) 将图像内的所有 block 的 HOG 特征直方图串联起来就可以得到该图像的 HOG 特征描述子了。这个就是最终的可供分类使用的特征向量了。

最终可以得到一个 $m \times n \times \alpha$ 个数据组成的高维向量，其中 m 表示每个 cell 中方向 bin 的数目， n, α 分别表示 block 的个数以及一个 block 中 cell 的数目。

在行人检测中的应用：

用一个窗口 (OpenCV 采用 64×128) 在图像中滑动，计算窗口对应的 HOG 特征。窗口中块的大小为 16×16 ，块的步长为 8×8 ，单元的大小为 8×8 。把每一个胞元投影到 9 个 bin，那么每一个单元对应的向量就是 9 维，每个 bin 对应该 9 维向量的一个数。该窗口特征向量的维数为：
 $9 \times \frac{16}{8} \times \frac{16}{8} \times (\frac{64-16}{8} + 1) \times (\frac{128-16}{8} + 1) = 3780$ 。

代码：

采用 vl_feat 中的 HOG 特征函数：

```

1     HOG = vl_hog(img, cellsize); //img表示输入的图像, cellsize为单元的大小
2     image = vl_hog('render', hog); //将HOG特征转换为图像

```

应用 (DPM(Deformable Part Model))： DPM 是一个非常成功的目标检测算法，连续获得 VOC (Visual Object Class) 07,08,09 年的检测冠军。目前已成为众多分类器、分割、人体姿态和行为分类的重要部分。2010 年 Pedro Felzenszwalb 被 VOC 授予“终身成就奖”。DPM 可以看做是 HOG (Histograms of Oriented Gradients) 的扩展，大体思路与 HOG 一致。先计算梯度方向直方图，然后

用 SVM (Support Vector Machine) 训练得到物体的梯度模型 (Model)。有了这样的模板就可以直接用来分类了，简单理解就是模型和目标匹配。DPM 只是在模型上做了很多改进工作。

9.1.2 SIFT 特征

SIFT (Scale invariant feature transform, 尺度不变特征变换) 是一种检测局部特征的算法，该算法通过求得一幅图像中的特征点及有关尺度和方向的描述子得到特征并进行图像特征点匹配，它由 David Lowe 在 1999 年 [7] 所发表，2004 年 [8] 完善总结。

主要思想：SIFT 算法是一种提取局部特征的算法，在尺度空间寻找极值点，提取位置、尺度、旋转不变量。

算法步骤：

(1) 构建尺度空间，检测极值点，获得尺度不变性。

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。用不同尺度（标准差）的高斯函数对图像进行平滑，然后比较平滑后图像的差别，差别大的像素就是特征明显的点。

原始图像经过不同尺度的高斯模糊所得的一组图像空间定义为：

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

其中 $I(x, y)$ 是原始图像， σ 大小决定图像的平滑程度， $G(x, y, \sigma)$ 是尺度可变高斯函数：

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

为了有效地在尺度空间检测到稳定的关键点，提出了高斯差分尺度空间。利用不同尺度的高斯差分核与图像卷积生成。

$$D(x, y, \sigma) = [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma)$$

(2) 精确定位出特征关键点的位置。

(3) 为关键点指定方向参数。

利用关键点邻域像素的梯度方向分布特性，我们可以为每个关键点指定方向参数方向，从而使描述子对图像旋转具有不变性，我们通过求每个极值点的梯度来为极值点赋予方向。

$$m(x, y) = \sqrt{[L(x+1, y) - L(x-1, y)]^2 + [L(x, y+1) - L(x, y-1)]^2}$$

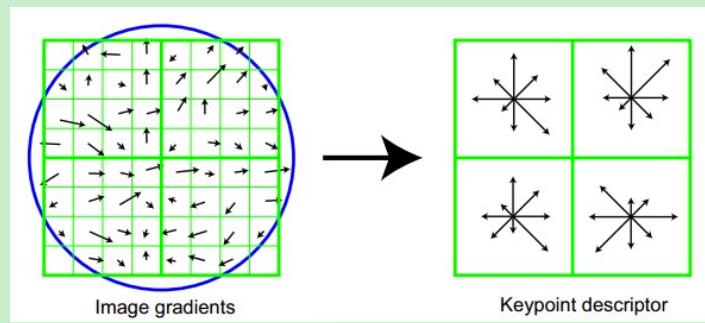
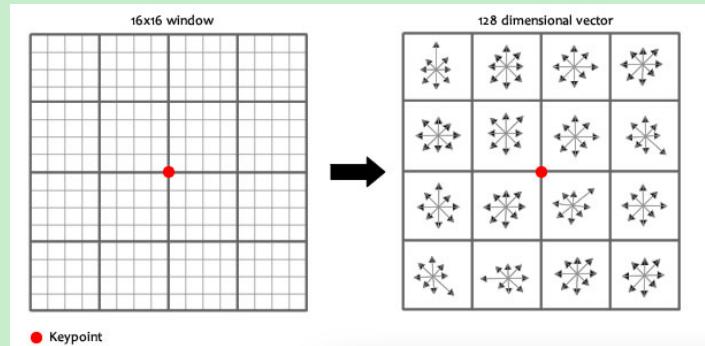


图 42:



$$\theta(x, y) = \alpha \tan 2 \frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)}$$

方向直方图的生成：确定关键点的方向采用梯度直方图统计法，统计以关键点为原点，一定区域内的图像像素点对关键点方向生成所做的贡献。梯度直方图的范围是 0 ~ 360 度，其中每 45 度一个柱，总共 8 个柱。以特征点为中心取 16*16 的邻域作为采样窗口，将采样点与特征点的相对方向通过高斯加权后归入包含 8 个 bin 的方向直方图，最后获得 4*4*8 的 128 维特征描述子。16x16 的图中其中 1/4 的特征点梯度方向及 scale，图42为其加权到 8 个主方向后的效果。

描述子采用 $4 \times 4 \times 4 = 128$ 维向量表征，综合效果最优。

(4) 关键点描述子产生。

1. 确定计算描述子所需的图像区域，图像区域的半径通过下式计算：

$$radius = \frac{3\theta_oct \times \sqrt{2} \times (d + 1)}{2}$$

2. 将坐标移至关键点主方向，如图43。

旋转后邻域内采样点的新坐标为：

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \sigma & -\sin \sigma \\ \sin \sigma & \cos \sigma \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

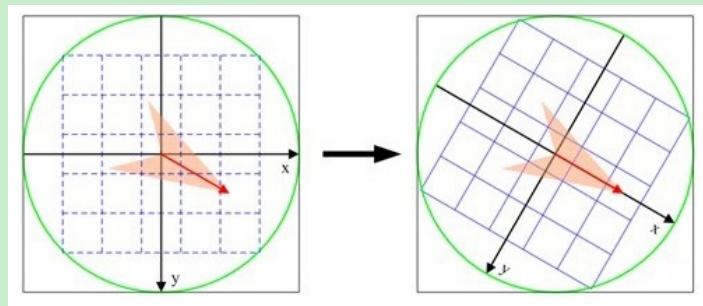


图 43: 将坐标移至关键点主方向

3. 在图像半径区域内对每个像素点求其梯度幅值和方向，然后对每个梯度幅值乘以高斯权重参数，生成方向直方图。
4. 在窗口宽度为 2×2 的区域内计算 8 个方向的梯度方向直方图，绘制每个梯度方向的累加值，即可形成一个种子点。然后再在下一个 2×2 的区域内进行直方图统计，形成下一个种子点，共生成 16 个种子点。
5. 描述子向量元素门限化及门限化后的描述子向量规范化。
6. 根据特征点的尺度对特征描述向量进行排序，SIFT 特征向量生成.

应用：

特征点检测匹配

代码：

采用 vl_feat 中的 sift 特征函数：

```
1 [f,d] = vl_sift(img);
```

9.1.3 SURF

SURF 与 SIFT 算法相似，SIFT 算法比较稳定，检测特征点更多，但是复杂度较高，而 SURF 要运算简单，效率高，运算时间短一点。

算法步骤：

- (1) 构建 Hessian 矩阵。采用的是 Hessian 矩阵行列式近似值图像。
- (2) 构造多尺度空间。

为了实现尺度不变性的特征点检测与匹配，SURF 算法与 SIFT 算法的第一步都是构造图像多尺度空间。SURF 算法则用高斯滤波与 Hessian 矩阵结合近似实现多尺度空间，计算复杂度相比 SIFT

降低多了。

(3) 根据非极大值抑制初步确定特征点。

(4) 主方向确定

(5) 构造特征点算术描述子, SURF 算法是选取一个 20S (单位) 的区域, 分成 4×4 分, 每一份中有 55S, 统计一份中的 dx , dy , $|dx|$, $|dy|$, 这样就得到 $4 \times 4 \times 4 = 64$ 的向量描述子。SIFT 是 128 个描述子, 比 SURF 复杂一点。

9.1.4 LBP

LBP (Local Binary Pattern, 局部二值模式)⁴是一种用来描述图像局部纹理特征的算子; 它具有旋转不变性和灰度不变性等显著的优点。它是首先由 T. Ojala, M.Pietikäinen, 和 D. Harwood 在 1994 年提出 [9], 用于纹理特征提取。而且, 提取的特征是图像的局部的纹理特征。

原始的 LBP 算子定义为在 3×3 的窗口内, 以窗口中心像素为阈值, 将相邻的 8 个像素的灰度值与其进行比较, 若周围像素值大于中心像素值, 则该像素点的位置被标记为 1, 否则为 0。这样, 3×3 邻域内的 8 个点经比较可产生 8 位二进制数 (通常转换为十进制数即 LBP 码, 共 256 种), 即得到该窗口中心像素点的 LBP 值, 并用这个值来反映该区域的纹理信息。如下图 44:

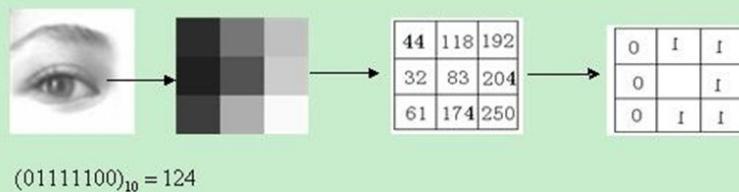


图 44: LBP 特征

对 LBP 特征向量进行提取的步骤:

1. 首先将检测窗口划分为 16×16 的小区域 (cell);
2. 对于每个 cell 中的一个像素, 将相邻的 8 个像素的灰度值与其进行比较, 若周围像素值大于中心像素值, 则该像素点的位置被标记为 1, 否则为 0。这样, 3×3 邻域内的 8 个点经比较可产生 8 位二进制数, 即得到该窗口中心像素点的 LBP 值;
3. 然后计算每个 cell 的直方图, 即每个数字 (假定是十进制数 LBP 值) 出现的频率; 然后对该直方图进行归一化处理。

⁴<http://blog.csdn.net/zouxy09/article/details/7929531>

4. 最后将得到的每个 cell 的统计直方图进行连接成为一个特征向量，也就是整幅图的 LBP 纹理特征向量；
5. 然后便可利用 SVM 或者其他机器学习算法进行分类了。

9.1.5 FV(Fisher vector)

Fisher vector 本质上是用似然函数的梯度 vector 来表达一幅图像。

主要思路：用生成式模型 (GMM) 对样本输入进行建模，进而得到样本的一种表示 (fisher vector)，再将这种表示 (fisher vector) 输入判别式分类器 (SVM) 得到图像分类结果。fisher vector 是 fisher kernel 中对样本特征的一种表示，它把一幅图片表示成一个向量。在高斯分布的基础上再找到变化的方向，可以更加准确的表示这一张图。

应用：图像的分类，目标识别等领域，特别是结合着 BOW model。

9.1.6 哈尔特征 (Haar-like features)

Haar-like 特征最早是由 Oren 等 [10] 应用于人脸表示，Viola 和 Jones [?] 在此基础上，使用 3 种类型 4 种形式的特征。

Viola 等提出的 Harr-like 特征如图 45：

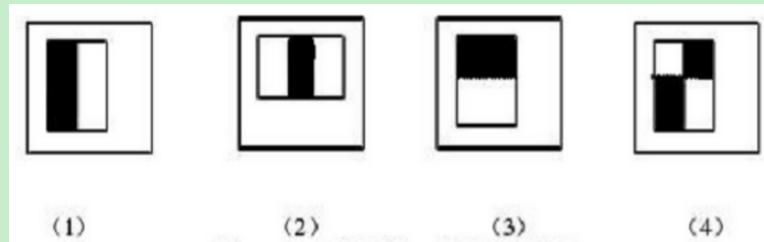


图 45: Viola 提出的 Haar-like 特征

将上面的任意一个矩形放到人脸区域上，然后，将白色区域的像素和减去黑色区域的像素和，得到的值我们暂且称之为人脸特征值，如果你把这个矩形放到一个非人脸区域，那么计算出的特征值应该和人脸特征值是不一样的，而且越不一样越好，所以这些方块的目的就是把人脸特征量化，以区分人脸和非人脸。

为了增加区分度，可以对多个矩形特征计算得到一个区分度更大的特征值，那么什么样的矩形特征怎么样的组合到一块可以更好的区分出人脸和非人脸呢，这就是 AdaBoost 算法要做的事了。

9.2 其它特征

9.2.1 GIST

GIST 是一个全局描述子，主要用于场景分类。

9.2.2 Harris-Laplace

9.2.3 GMM-HMM(Gaussian Mixture Model-Hidden Markov Model)

应用：语音识别

9.2.4 Gabor

Gabor 变换是 D.Gabor 1946 年提出的。Gabor 变换属于加窗傅立叶变换，Gabor 函数可以在频域不同尺度、不同方向上提取相关的特征。另外 Gabor 函数与人眼的生物作用相仿，所以经常用作纹理识别上，并取得了较好的效果。

Gabor 滤波器是一个用于边缘检测的线性滤波器。用 Gabor 函数形成的二维 Gabor 滤波器具有在空间域和频率域同时取得最优局部化的特性，与人类生物视觉特性很相似，因此能够很好地描述对应于空间频率(尺度)、空间位置及方向选择性的局部结构信息。同时，二维 Gabor 函数也类似于增强边缘以及峰、谷、脊轮廓等底层图像特征，这相当于增强了被认为是面部关键部件的眼睛、鼻子、嘴巴等信息，同时也增强了诸于黑痣、酒窝、伤疤等局部特征，从而使得在保留总体人脸信息的同时增强局部特性成为可能。它的小波特性说明了 Gabor 滤波结果是描述图像局部灰度分布的有力工具，因此，可以使用 Gabor 滤波来抽取图像的纹理信息。由于 Gabor 特征具有良好的空间局部性和方向选择性，而且对光照、姿态具有一定的鲁棒性，因此在人脸识别中获得了成功地应用。

应用：人脸识别、人脸表情识别、汉字识别

9.2.5 DoG

9.2.6 local symmetry

Reisfeld 等人 [11] 发表的论文中有关于局部对称性的描述。

Isotropic Symmetry Operator:

基于周围像素的灰度梯度，计算在给定点 p 处的对称性程度。如图 46 所示，点 p 处的对称性可以通过比较位于 p_i 和 p_j 处的像素对 i 和 j 的灰度梯度来度量，其中 $p = (p_i + p_j)/2$ 。每对像素对对点

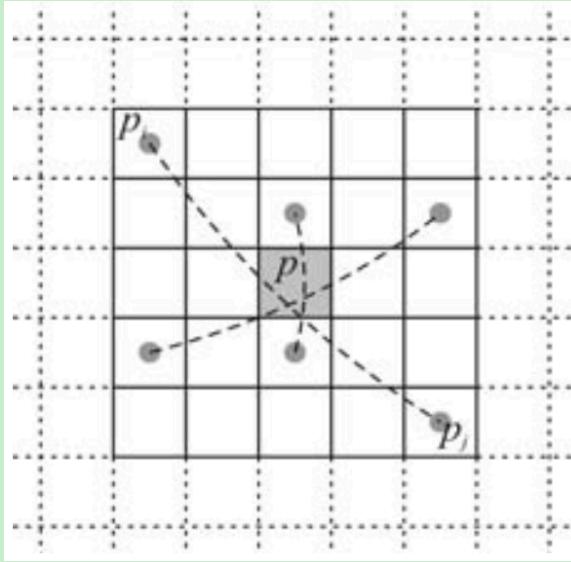


图 46: 用对称性算子来对像素对的梯度进行比较的三个例子

p 处的局部对称性的贡献为

$$c(i, j) = d(i, j, \sigma) \cdot p(i, j) \cdot m_i \cdot m_j \quad (1)$$

m_i 是像素点 i 的梯度的模, $d(i, j, \sigma)$ 是当标准差为 σ 时, 两个像素点间的距离的高斯加权函数, 对称性

$$p(i, j) = \left(1 - \cos(\gamma_i + \gamma_j)\right) \cdot \left(1 - \cos(\gamma_i - \gamma_j)\right) \quad (2)$$

其中, $\gamma_i = \theta_i - \alpha$, 见图 47。当点 p_i 、 p_j 关于点 p 对称时, $\gamma_i + \gamma_j = \pi$, 式 (2) 中的第一项取得最大值。如果只用第一项, 会使得在直线边缘上的点取得较大的值, 而这些点通常并不具有对称性。为了避免这个问题, 又加入了第二项, 表示具有相似梯度方向的像素对。最后, 对所有像素对的贡献进行求和得到点 p 处的 isotropic symmetry 值

$$M^{iso}(x, y) = \sum_{(i, j) \in \Gamma(p)} c(i, j) \quad (3)$$

为了使 symmetry operator 对有多个对称轴的对称图案更敏感, Reisfeld 等人 [?] 又提出了 radial symmetry operator, 作为 isotropic symmetry operator 的扩展。首先, 像素对的贡献的方向用 $\varphi(i, j) = (\theta_i + \theta_j)/2$ 来计算。然后, 对最大贡献为 $c(i, j)$ 的像素对 (i, j) , 对称方向被确定为

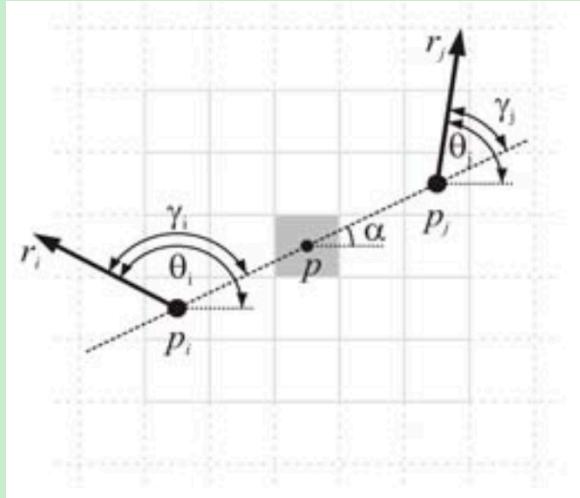


图 47: 每对像素对的贡献的几何表示

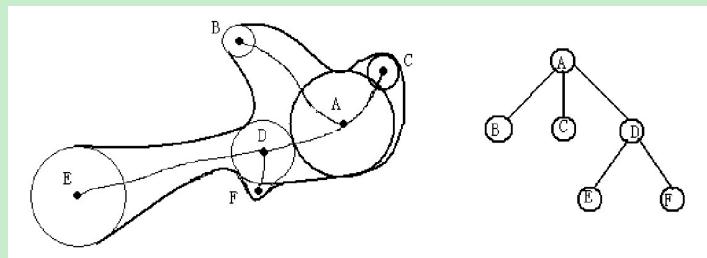


图 48: 骨架树

$\phi(p) = \varphi(i, j)$ 。这个值也用来提升有不相似方向的像素对的贡献

$$M^{rad} = \sum_{(i,j) \in \Gamma(p)} c(i, j) \cdot \sin^2(\varphi(i, j) - \phi(p)) \quad (4)$$

9.2.7 Covariance Matrix

9.2.8 骨架特征

理论：骨架又称为中轴，它的定义来自于烧草模型和最大圆盘模型。骨架不仅包含了物体的几何特征，也描述了物体的拓扑结构。经典的骨架化算法有中轴变换、细化算法、Voronoi 图算法等。在骨架的基础上可以通过奇点图、骨架树等对物体的结构特征进行描述。

骨架树：将骨架映射到一个树状结构（如图48）。仅有一个邻接点的骨架点成为端点，有两个以上邻接点的骨架点称为分支点，骨架上除了分支点和端点外的点称为连接点。根据骨架树深度优先搜索产生的节点序列中的所有节点用其孩子数替换，替换后得到的新序列即为树描述符。例如，图48的树描述符为 (3, 2, 0, 0, 0, 0)。

骨架化代码：<http://cs.smith.edu/~nhowe/research/code/>。

9.2.9 灰度特征

1) 灰度直方图 灰度直方图是灰度级函数，它表示图像中每种灰度级的像素的个数，反映图像中每种灰度出现的频率。灰度直方图的横坐标是灰度级，纵坐标是该灰度级出现的频率。

```
1 imhist(I,n);
```

2) 灰度共生矩阵 (GLCM)

灰度共生矩阵是对图像上保持某距离的两像素分别具有某种灰度的情况进行统计得到的。它不仅反映亮度的分布特性，也反映具有同样亮度或接近亮度的像素之间的位置分布特性。对灰度共生矩阵的理解，需要明确几个概念：方向、偏移量和灰度共生矩阵的阶数。

方向 一般计算过程会分别选在几个不同的方向来进行，通常为 0, 45, 90, 135。

偏移量

灰度共生矩阵阶数 灰度共生矩阵的阶数和灰度图像灰度值的阶数是一致的，即当灰度图像灰度值的阶数是 N 时，灰度共生矩阵为 $N \times N$ 。

取图像 ($N \times N$) 中任意一点 (x,y) 及偏离它的另一点 $(x+a,y+b)$ ，设该点对的灰度值为 (g_1,g_2) 。令点 (x,y) 在整个画面上移动，则会得到各种 (g_1,g_2) 值，设灰度值的级数为 256，则 (g_1,g_2) 的组合共有 256×256 种（共生矩阵的大小）。对于整个画面，统计出每一种 (g_1,g_2) 值出现的次数，然后排列成一个方阵，在用 (g_1,g_2) 出现的总次数将它们归一化为出现的概率 $P(g_1, g_2)$ ，这样的方阵称为灰度共生矩阵。

为了能更直观的以共生矩阵描述纹理情况，从共生矩阵导出一下反映矩阵情况的参数，典型的有一下几种：

能量 为灰度共生矩阵元素值的平方和，反映了图像灰度分布均匀程度和纹理粗细度。ASM 值大表明一种较均一和规则变化的纹理模式。

$$ASM = \sum_{i=0}^k \sum_{j=0}^k G(i,j)^2$$

对比度 反映了某个像素值及其领域像素值的亮度的对比情况。如果偏离对角线的元素有较大值，即

图像亮度值变化很快，则 CON 会有较大取值。

$$CON = \sum_i \sum_j (i - j)^2 G(i, j)$$

熵 表示图像的信息量，当共生矩阵中所有元素有最大的随机性、空间共生矩阵中所有值几乎相等时，共生矩阵中元素分散分布时，熵较大。它表示了图像中纹理的非均匀程度或复杂程度。

$$ENT = \sum_i \sum_j G(i, j) \log G(i, j)$$

自相关 反应了图像纹理的一致性。如果图像中有水平方向纹理，则水平方向矩阵的 COR 大于其余矩阵的 COR 值。

$$COR = \frac{\sum_i \sum_j j[(ij)G(i, j)] - u_i u_j}{s_i s_j}$$

```
1 [glcm, SI] = graycomatrix(Img);
```

9.2.10 凸包

9.2.11 ICA(Independent Component Analysis)

独立成分分析 (Independent Component Analysis, ICA) 是近年来出现的一种强有力的数据分析工具 (Hyvarinen A, Karhunen J, Oja E, 2001; Roberts S J, Everson R, 2001)。1994 年由 Comon 给出了 ICA 的一个较为严格的数学定义，其思想最早是由 Heranlt 和 Jutten 于 1986 年提出来的。ICA 从出现到现在虽然时间不长，然而无论从理论上还是应用上，它正受到越来越多的关注，成为国内外研究的一个热点。特别是从应用角度看，它的应用领域与应用前景都是非常广阔的，目前主要应用于盲源分离、图像处理、语言识别、通信、生物医学信号处理、脑功能成像研究、故障诊断、特征提取、金融时间序列分析和数据挖掘等。ICA 是一种用来从多变量（多维）统计数据里找到隐含的因素或成分的方法，被认为是主成分分析 (Principal Component Analysis, PCA) 和因子分析 (Factor Analysis) 的一种扩展。对于盲源分离问题，ICA 是指在只知道混合信号，而不知道源信号、噪声以及混合机制的情况下，分离或近似地分离出源信号的一种分析过程。

9.2.12 SC

SC[2] (Shape context, 形状上下文) 是基于物体轮廓样本点进行描述的。



图 49: 原始图像

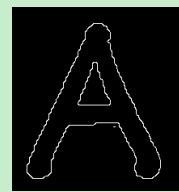


图 50: 边缘提取

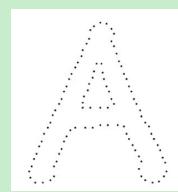


图 51: 采样点

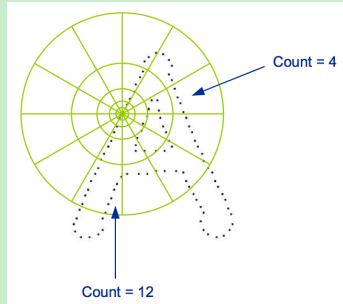


图 52: 对数极坐标空间

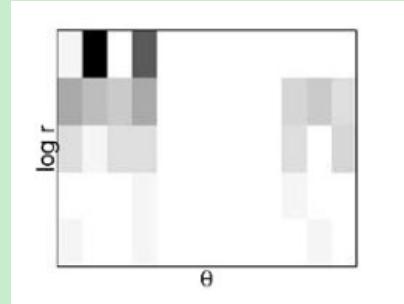


图 53: 直方图

主要思想：用直方图描述轮廓上点间的位置关系。

算法步骤：

1. 边缘提取 (如图50)。
2. 从边缘上均匀采样 n 个点，得到一个物体形状的点集合 (如图51)。
3. 通过对数极坐标空间 (如图52) 计算每个采样点与形状其他点之间的位置关系，采用直方图表示 (如图53)。 n 个采样点中，每个点都有一个直方图对其位置进行描述。

但由于形状上下文不能很好地解决物体类内部之间的变形，因此采用基于内部距离的形状上下文 [6]，内距离形状上下文中采用内距离代替欧式距离。

内距离形状上下文代码：http://www.dabi.temple.edu/~hbling/code_data.htm。

田奇

binary feature descriptors

9.2.13 Binary SIFT (BSIFT) – Zhou, TIP 2014

9.2.14 COGE - PCM 2013 Best Paper, Mao , 2013

9.2.15 Ultra-Short Binary Descriptor (USB) – Zhang S., TIP , 2014

9.2.16 Topology-Preserving Hashing – Zhang L., MM 2012

10. 学习算法

10.1 贝叶斯学习

贝叶斯学习是由英国学者 T. Bayesian (贝叶斯) [1] 于 1763 年提出的一种归纳推理的理论。

贝叶斯学习的基本公式如下：

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (5)$$

10.2 SVM 方法

SVM (Support Vector Machine, 支持向量机) 是一种有监督的统计学习方法，能够最小化经验误差和最大化几何边缘，被称为最大间隔分类器，可用于分类与回归分析。最早的相关论文为美国贝尔实验室 (Bell Lab., USA) 的 Corinna Cortes 和 Vladimir Vapnik 等人于 1995 年在 Machine Learning 杂志上提出 [3]，被引用次数已经超过 17600 次。

10.3 K-means

10.4 Decision tree

10.5 EM(Expectation Maximization Algorithm)

期望最大算法 (Expectation Maximization Algorithm, EM) 是一种从不完全数据或有数据丢失的数据集中求解概率模型参数的最大似然估计方法。

算法步骤：

1. 初始化分布参数。

2. 重复以下步骤直到收敛：

(a) E 步骤：估计未知参数的期望值，给出当前的参数估计。

(b) M 步骤：重新估计分布参数，以使得数据的似然性最大，给出未知变量的期望估计。

10.6 KNN

K 最近邻分类算法，是一个理论上比较成熟的方法，也是最简单的机器学习算法之一。该方法的思路是：如果一个样本在特征空间中的 k 个最相似 (即特征空间中最邻近) 的样本中的大多数属于某

一个类别，则该样本也属于这个类别。由于 KNN 方法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属类别的，因此对于类域的交叉或重叠较多的待分样本集来说，KNN 方法较其他方法更为适合。

10.7 集成学习 (Ensemble learning)

集成学习 (Ensemble learning) 是使用一系列学习器进行学习，并使用某种规则把各个学习结果进行整合从而获得比单个学习器更好的学习效果的一种机器学习方法。

主要思路：

在对新的实例进行分类的时候，把若干个单个分类器集成起来，通过对多个分类器的分类结果进行某种组合来决定最终的分类，以取得比单个分类器更好的性能。

10.7.1 AdaBoost

AdaBoost，是英文“Adaptive Boosting”（自适应增强）的缩写，由 Yoav Freund 和 Robert Schapire 在 1995 年提出。它的自适应在于：前一个基本分类器分错的样本会得到加强，加权后的全体样本再次被用来训练下一个基本分类器。同时，在每一轮中加入一个新的弱分类器，直到达到某个预定的足够小的错误率或达到预先指定的最大迭代次数。

主要步骤：

1. 初始化训练数据的权值分布。如果有 N 个样本，则每一个训练样本最开始时都被赋予相同的权值： $\frac{1}{N}$ 。
2. 训练弱分类器。具体训练过程中，如果某个样本点已经被准确地分类，那么在构造下一个训练集中，它的权值就被降低；相反，如果某个样本点没有被准确地分类，那么它的权值就得到提高。然后，权值更新过的样本集被用于训练下一个分类器，整个训练过程如此迭代地进行下去。
3. 将各个训练得到的弱分类器组合成强分类器。各个弱分类器的训练过程结束后，加大分类误差率小的弱分类器的权重，使其在最终的分类函数中起着较大的决定作用，而降低分类误差率大的弱分类器的权重，使其在最终的分类函数中起着较小的决定作用。

相关资料：<http://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/top10chapter.pdf>

10.7.2 Bagging

主要思路：

让学习算法训练多轮，每轮的训练集由从初始的训练集中随机取出的 n 个训练样本组成，某个初始训练样本在某轮训练集中可以出现多次或根本不出现，训练之后可得到一个预测函数序列 h_1, \dots, h_n ，最终的预测函数 H 对分类问题采用投票方式。

随机森林(Random Forest)是一种采用决策树作为基预测器的集成学习方法，2001 年由 Breiman [?] 提出，结合 Bagging 和随机子空间理论，集成众多决策树进行预测，通过各个决策树的预测值进行平均或投票，得到最终预测结果。

“视觉机器学习 20 讲”中提供了随机森林的源代码，是 c 与 matlab 的混合编程，但是只有在 windows 下编译好了的文件，在 mac 下无法跑。

DRFI 方法中用到随机森林⁵学习方法，下载下来解压后是在 randomforest-matlab 文件夹中，其中又包含两个文件夹，一个是 RF_Class_C，用来做分类的，另一个是 RF_Reg_C，用来做回归的。

RF_Class_C 文件夹中需要对.cpp 文件进行编译，生成 mexmaci64 文件，这一步可以通过在终端输入 make mex 命令实现。但是可能由于我电脑中的 fortran 版本不行，编译总是出错，目前还没有解决，后来在网页上⁶搜到了已编译好的 mexmaci64 文件，可以直接拿过来用。

tutorial_ClassRF.m 文件列举了 10 个例子，展示了用不同的参数跑随机森林学习算法的不同结果。

10.7.3 Gasen

GASEN[?]（基于遗传算法的选择性集成学习算法）。随机生成若干权向量，权向量的每个分量对应了一个个体学习器，这些权向量被遗传算法进化，得到一个最优权向量，它表示了各个体学习器在构成集成时的“重要性”，据此进行个体的选择。该方法是建立在遗传算法基础上，由于遗传算法计算效率低，使得 GASEN 计算量更大。

代码：http://lamda.nju.edu.cn/Default.aspx?Page=code_GASEN&NS=&AspxAutoDetectCookieSupport=1

参考文献

- [1] Mr. Bayes and Mr Price. An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfrs. *Philosophical Transactions (1683-1775)*, pages 370–418, 1763.

⁵<https://code.google.com/p/randomforest-matlab/>

⁶<https://code.google.com/p/randomforest-matlab/issues/detail?id=54>

- [2] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509–522, 2002.
- [3] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [4] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.
- [5] G Gorsky and Ph Grosjean. Qualitative and quantitative assessment of zooplankton samples. *globec int. Newsletter*, 9(1):5, 2003.
- [6] Haibin Ling and David W Jacobs. Shape classification using the inner-distance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):286–299, 2007.
- [7] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.
- [8] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [9] Timo Ojala, Matti Pietikainen, and David Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, number 1, pages 582–585, 1994.
- [10] Michael Oren, Constantine Papageorgiou, Pawan Sinha, Edgar Osuna, and Tomaso Poggio. Pedestrian detection using wavelet templates. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 193–199. IEEE, 1997.
- [11] Daniel Reisfeld, Haim Wolfson, and Yechezkel Yeshurun. Context-free attentional operators: the generalized symmetry transform. *International Journal of Computer Vision*, 14(2):119–130, 1995.