浮游动物识别

王如晨 朱亚菲

2015 年 7 月

目录

1. 竞赛

1.1 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，这对于一张图片有多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。

1.2 IMAGENET Large Scale Visual Recognition Challenges 2010-2015

ILSVRC 每年的竞赛内容包括：

- ILSVRC2014: Detection and Classification and localization
- ILSVRC2013: Detection and Classification and localization
- ILSVRC2012: Classification, Classification and localization and Fine-grained classification
- ILSVRC2011: Classification and Classification and localization
- ILSVRC2010: Classification

1.3 Labeled Faces in the Wild

2. 特征（Ours）

2.1 经典特征

2.1.1 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 [8] 所发表, 2004 年 [9] 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。

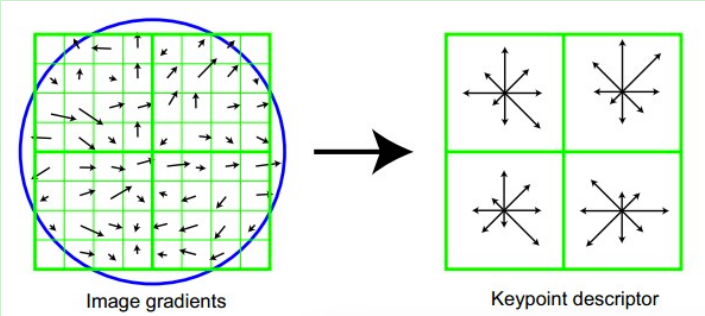
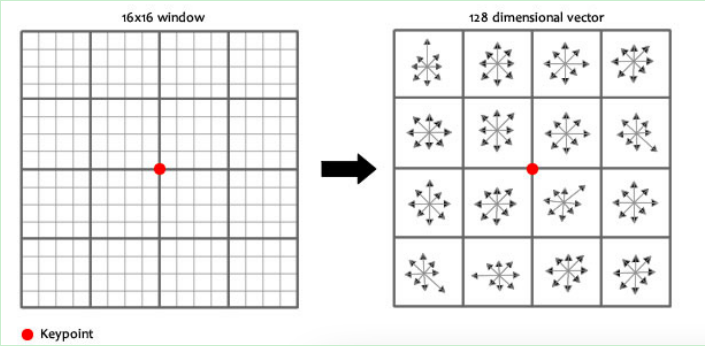


图 1:



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。

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

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

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 $4\times4\times8$ 的 128 维特征描述子。16×16 的图中其中 1/4 的特征点梯度方向及 scale，图1为其加权到 8 个主方向后的效果。

描述子采用 $4\times4\times8=128$ 维向量表征，综合效果最优。

4. 关键点描述子产生。

   (a) 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：

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

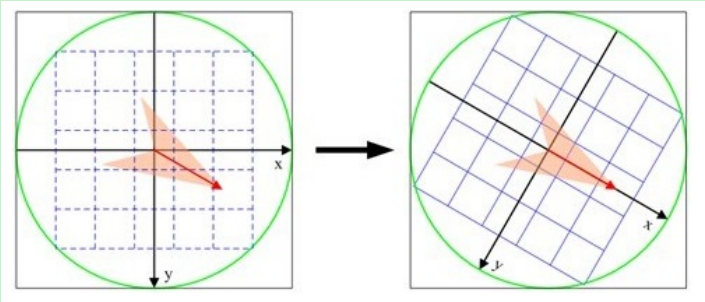


图 2: 将坐标移至关键点主方向

(b) 将坐标移至关键点主方向，如图2。

旋转后邻域内采样点的新坐标为：

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

(c) 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。

(d) 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。

(e) 描述子向量元素门限化及门限化后的描述子向量规范化。

(f) 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);
```

2.1.2 HOG 特征

HOG [5]（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

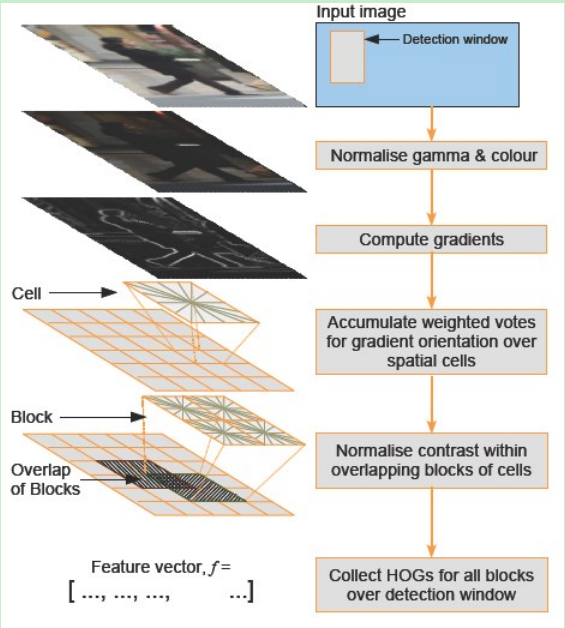


图 3: HOG 特征

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3。

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：

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

   比如可以取 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：

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

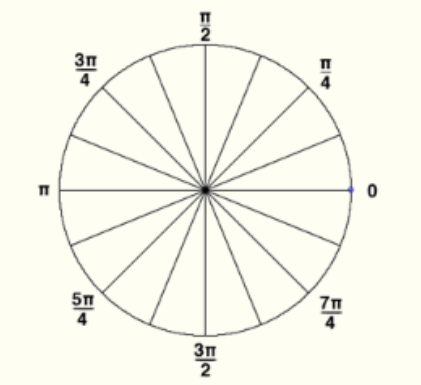


图 4: 梯度方向 bin

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

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4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。

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5。

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。

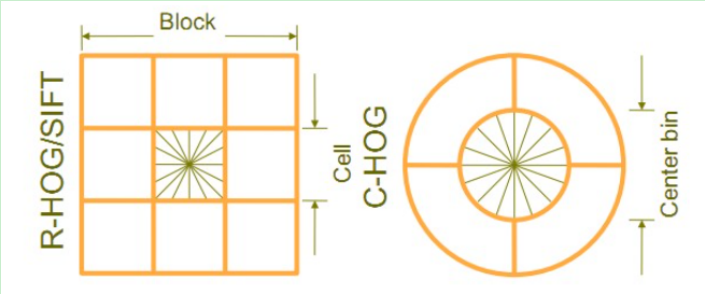


图 5: 矩形区间和环形区间

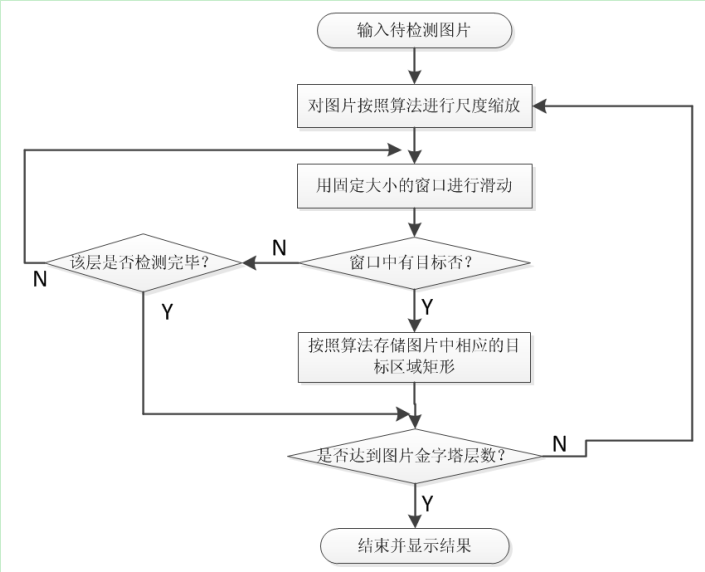


图 6: 行人检测流程

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: $9 \times \frac{16}{8} \times \frac{16}{8} \times (\frac{64-16}{8}+1) \times (\frac{128-16}{8}+1) = 3780$。

行人检测过程如图 6:

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（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（Histogrrams 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（Surpport 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。

2.1.3 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 5×5S，统计一份中的 $\sum d_x$，$\sum d_y$，$\sum |d_x|$，$\sum |d_y|$，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。

2.1.4 LBP

LBP（Local Binary Pattern，局部二值模式）[1]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 [10]，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

[1] http://blog.csdn.net/zouxy09/article/details/7929531

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7：

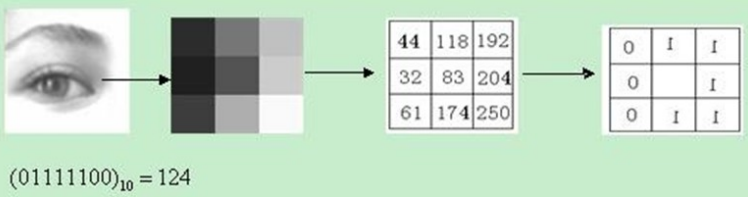


图 7: LBP 特征

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. 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。

2.1.5 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。

2.1.6 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

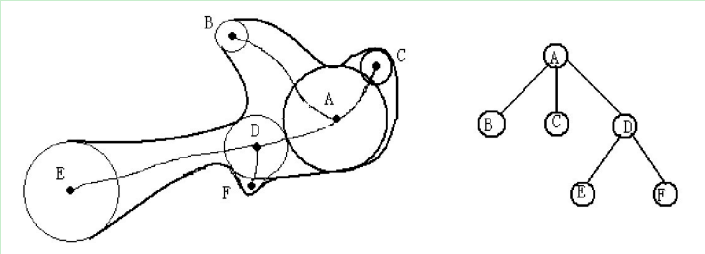


图 8: 骨架树

图 9: 原始图像

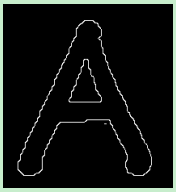


图 10: 边缘提取

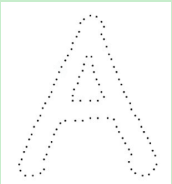


图 11: 采样点

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8)。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8的树描述符为（3，2，0，0，0，0)。

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

2.1.7 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。

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10)。
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11)。
3. 通过对数极坐标空间（如图12）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13)。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 [7]，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。

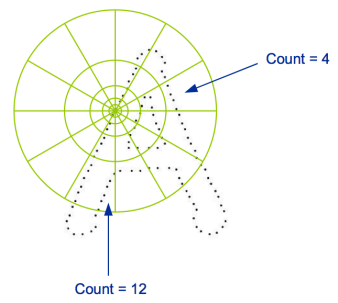


图 12: 对数极坐标空间

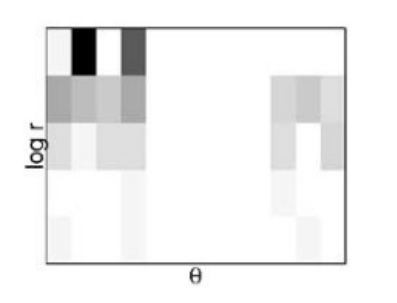


图 13: 直方图

2.1.8 哈尔特征（Haar-like features）

Haar-like 特征最早是由 Oren 等 [11] 应用于人脸表示，Viola 和 Jones [13] 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 14：

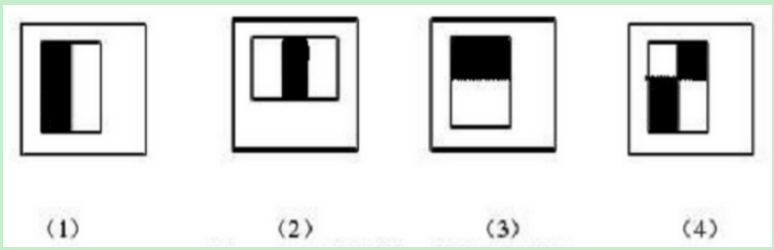


图 14: Viola 提出的 Haar-like 特征

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。

2.2 其它特征

2.2.1 local symmetry

Reisfeld 等人 [12] 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 15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

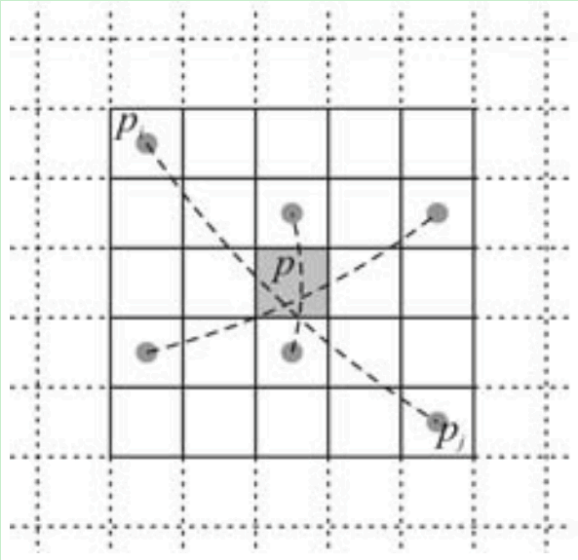


图 15: 用对称性算子来对像素对的梯度进行比较的三个例子

p 处的局部对称性的贡献为

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

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

$$p(i,j) = \Big(1 - cos(\gamma_i + \gamma_j)\Big) \cdot \Big(1 - cos(\gamma_i - \gamma_j)\Big) \tag{2}$$

其中，$\gamma_i = \theta_i - \alpha$，见图 16。当点 p_i、p_j 关于点 p 对称时，$\gamma_i + \gamma_i = \pi$，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

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

为了使 symmetry operator 对有多个对称轴的对称图案更敏感，Reisfeld 等人 [12] 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

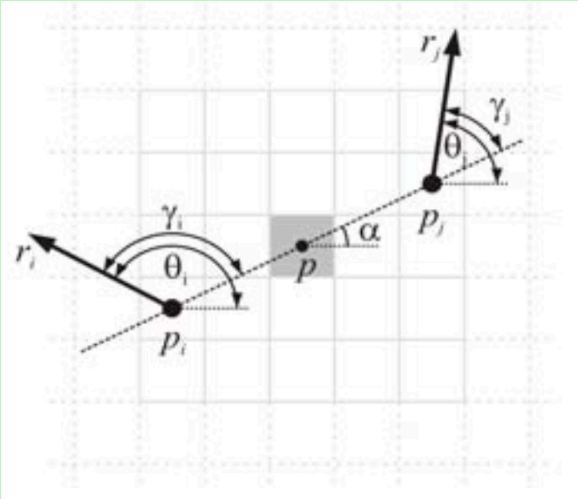


图 16: 每对像素对的贡献的几何表示

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

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

2.2.2 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 (g1,g2)。令点 (x,y) 在整个画面上移动，则会得到各种 (g1,g2) 值，设灰度值的级数为 256，则 (g1,g2) 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 (g1,g2) 值出现的次数，然后排列成一个方阵，在用 (g1,g2) 出现的总次数将它们归一化为出现的概率 $P(g1, g2)$，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。

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

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

2.2.3 GIST

GIST 是一个全局描述子，主要用于场景分类。

2.2.4 Harris-Laplace

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

应用：语音识别

2.2.6 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的应用。

应用：人脸识别、人脸表情识别、汉字识别

2.2.7 DoG

2.2.8 Covariance Matrix

2.2.9 凸包

2.2.10 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。

田奇

binary feature descriptors

2.2.11 Binary SIFT (BSIFT) – Zhou, TIP 2014

2.2.12 COGE - PCM 2013 Best Paper, Mao , 2013

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

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

3. 学习算法

3.1 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：

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

3.2 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 [4]，被引用次数已经超过 17600 次。

代码：SVMlight[2]和 LibSVM[3]

SVM 是一种典型的两类分类器，即它只回答属于正类还是负类的问题。当遇到多类别的时候，一般采取如下两种策略：

[2] http://svmlight.joachims.org/
[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm

- 一对多法：训练时依次把某个类别的样本归为一类，其他剩余的样本归为另一类，这样 k 个类别的样本就构造出了 k 个 SVM。分类时将未知样本分类为具有最大分类函数值的那类。

  比如我们有 5 个类别，第一次就把类别 1 的样本定为正样本，其余 2，3，4，5 的样本合起来定为负样本，这样得到一个两类分类器，它能够指出一篇文章是还是不是第 1 类的；第二次我们把类别 2 的样本定为正样本，把 1，3，4，5 的样本合起来定为负样本，得到一个分类器，如此下去，我们可以得到 5 个这样的两类分类器（总是和类别的数目一致）。到了有文章需要分类的时候，我们就拿着这篇文章挨个分类器的问：是属于你的么？是属于你的么？哪个分类器点头说是了，文章的类别就确定了。这种方法的好处是每个优化问题的规模比较小，而且分类的时候速度很快（只需要调用 5 个分类器就知道了结果）。但有时也会出现两种很尴尬的情况，例如拿一篇文章问了一圈，每一个分类器都说它是属于它那一类的，或者每一个分类器都说它不是它那一类的，前者叫分类重叠现象，后者叫不可分类现象。分类重叠倒还好办，随便选一个结果都不至于太离谱，或者看看这篇文章到各个超平面的距离，哪个远就判给哪个。不可分类现象就着实难办了，只能把它分给第 6 个类别了。更要命的是，本来各个类别的样本数目是差不多的，但“其余”的那一类样本数总是要数倍于正类（因为它是除正类以外其他类别的样本之和嘛），这就人为的造成了上一节所说的“数据集偏斜”问题。

- 一对一法：其做法是在任意两类样本之间设计一个 SVM，因此 k 个类别的样本就需要设计 $\frac{k(k-1)}{2}$ 个 SVM。当对一个未知样本进行分类时，最后得票最多的类别即为该未知样本的类别。LibSVM 中的多类分类就是根据这个方法实现的。

  每次选一个类的样本作正类样本，而负类样本则变成只选一个类，这就避免了偏斜。因此过程就是算出这样一些分类器，第一个只回答“是第 1 类还是第 2 类”，第二个只回答“是第 1 类还是第 3 类”，第三个只回答“是第 1 类还是第 4 类”，如此下去，你也可以马上得出，这样的分类器应该有 $\frac{5\times4}{2}=10$ 个（通式是，如果有 k 个类别，则总的两类分类器数目为 $\frac{k(k-1)}{2}$）。虽然分类器的数目多了，但是在训练阶段（也就是算出这些分类器的分类平面时）所用的总时间却比“一类对其余”方法少很多，在真正用来分类的时候，把一篇文章扔给所有分类器，第一个分类器会投票说它是“1”或者“2”，第二个会说它是“1”或者“3”，让每一个都投上自己的一票，最后统计票数，如果类别“1”得票最多，就判这篇文章属于第 1 类。这种方法显然也会有分类重叠的现象，但不会有不可分类现象，因为总不可能所有类别的票数都是 0。看起来够好么？其实不然，想想分类一篇文章，我们调用了多少个分类器？10 个，这还是类别数为 5 的时候，类别数如果是 1000，要调用的分类器数目会上升至约 500,000 个（类别数的平方量级）。

- 层次支持向量机：还是像一对一方法那样来训练。在分类时可以先问分类器“1 对 5”（意思是

它能够回答“是第 1 类还是第 5 类”），如果它回答 5，我们就再问“2 对 5”这个分类器，如果它还说是“5”，我们就继续问下去，就可以得到分类结果。好处在哪？我们其实只调用了 4 个分类器（如果类别数是 k，则只调用 k-1 个），分类速度飞快，且没有分类重叠和不可分类现象！缺点在哪？假如最一开始的分类器回答错误（明明是类别 1 的文章，它说成了 5），那么后面的分类器是无论如何也无法纠正它的错误的（因为后面的分类器压根没有出现“1”这个类别标签），其实对下面每一层的分类器都存在这种错误向下累积的现象。

3.3 K-means

3.4 Decision tree

3.5 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。

3.6 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。

3.7 ELM（极限学习机）

ELM 算法 [6] 是新加坡南洋理工大学的黄广斌教授提出来的。

代码：http://www.ntu.edu.sg/home/egbhuang/

3.8 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。

3.8.1 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

3.8.2 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 $h\_1, \ldots, h\_n$，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 [3] 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[4]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[5]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。

3.8.3 Gasen

GASEN [14] (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=&AspxAutoDetectCookieSuppor
1

4. 技术路线和方案

4.1 技术路线

在设计浮游动物分类系统过程中，需要完成特征和分类器的选择，我们的技术路线如图 17。

特征选取：根据浮游动物的形态特征和计算机视觉中分类识别常用的经典特征，我们选取的特征如图 17。

分类器的训练：根据经典学习方法和集成学习，我们选择的学习算法如图 17。

[4] https://code.google.com/p/randomforest-matlab/
[5] https://code.google.com/p/randomforest-matlab/issues/detail?id=54

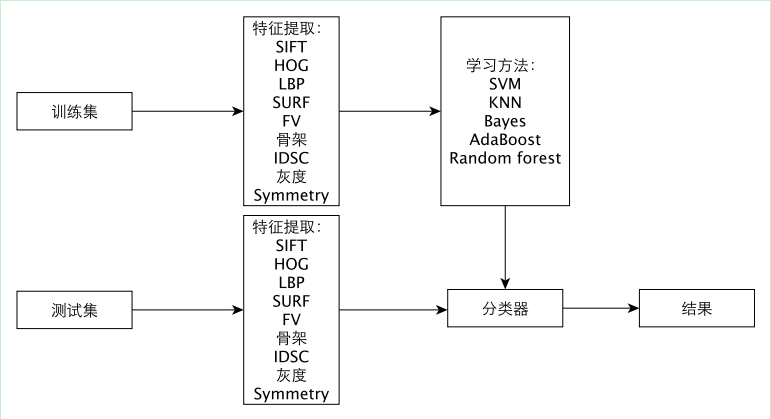


图 17: 技术路线

5. 实验

5.1 HOG+SVM

5.1.1 HOG

OpenCV 提取 HOG 特征所参考的网址：http://docs.opencv.org/3.0-rc1/d5/d33/structcv_1_1HOGDescriptor.html。

HOG 检测滑动窗口内部的参数包括：检测窗口的大小、窗口内 block 的大小、block 滑动的步长以及 cell 的大小。其中滑动窗口所包含的区域即我们需要提取 HOG 特征的区域。

与行人检测不同，在浮游动物分类问题中，没有复杂的背景，所以不需要用窗口进行滑动，判断哪个窗口中有目标，只需将整幅图像看成一个窗口，判断该窗口中的目标属于哪个类别。在本实验中，将图像 resize 为 64×64，以整幅图像作为一个窗口提取 HOG 特征，其中块（block）选用 16×16，block 滑动的步长选用 8×8，单元（cell）选用 8×8。则对于一幅图像，所提取的 HOG 特征向量维数为 $(\frac{64-16}{8}+1) \times (\frac{64-16}{8}+1) \times 4 \times 9 = 1764$ 维。每个 block 中的特征向量是对其中的 4 个 cell 按从上到下，从左到右的顺序将 9 维特征向量连起来得到的。而整个窗口中的特征又是对其中的所有 block 按从上到下，从左到右的顺序将特征连起来得到的。

5.1.2 SVM

在实验中，针对 SVM 核函数选用了 Linear 核和 RBF 核。

- Linear 核：主要用于线性可分的情形。参数少，速度快，对于一般数据，分类效果已经很理想了。
- RBF 核：主要用于线性不可分的情形。参数多，分类结果非常依赖于参数。

5.1.3 HOG+C-SVC Linear

实验得到的 **re-substitution Confusion Matrix**（自身验证自身）如图 18:

	Appendicularia	Bubble	Chaetognatha	CladoceraPenilia	Copepoda	Decapoda	Doliolida	Egg	Fiber	Gelatinous	Multiple	Nonbio	Pteropoda	Total	Recall	1-Precision
Appendicularia	544	0	0	0	0	0	0	0	0	0	0	1	0	545	0.998	0.151
Bubble	0	145	0	0	0	0	0	0	0	0	0	0	0	145	1	0
Chaetognatha	0	0	349	0	0	0	0	0	0	0	0	0	0	349	1	0.046
CladoceraPenilia	0	0	0	772	3	0	0	0	0	3	4	143	0	925	0.835	0.441
Copepoda	1	0	0	24	1142	55	0	0	0	34	94	123	1	1474	0.775	0.354
Decapoda	16	0	2	11	55	399	0	0	1	1	8	63	1	557	0.716	0.294
Doliolida	0	0	0	0	0	0	284	0	0	0	0	1	0	285	0.996	0.424
Egg	0	0	0	0	0	0	0	344	0	0	0	0	0	344	1	0.216
Fiber	0	0	0	0	0	0	0	0	304	0	0	0	0	304	1	0.116
Gelatinous	0	0	0	24	54	4	16	16	0	412	1	75	0	602	0.684	0.446
Multiple	31	0	5	38	99	16	2	0	8	14	308	114	0	635	0.485	0.496
Nonbio	48	0	10	513	415	91	191	79	31	280	196	1223	1	3078	0.397	0.298
Pteropoda	1	0	0	0	0	0	0	0	0	0	0	0	216	217	0.995	0.014
Total	641	145	366	1382	1768	565	493	439	344	744	611	1743	219	9460	0.837	0.254

图 18: 混淆矩阵（训练集与测试集相同）

对测试集进行预测得到的混淆矩阵如图 19:

	Appendicularia	Bubble	Chaetognatha	CladoceraPenilia	Copepoda	Decapoda	Doliolida	Egg	Fiber	Gelatinous	Multiple	Nonbio	Pteropoda	Total	Recall	1-Precision
Appendicularia	52	0	7	1	3	2	0	0	12	0	8	13	2	100	0.52	0.669
Bubble	0	96	0	0	1	0	0	1	0	0	1	1	0	100	0.96	0
Chaetognatha	38	0	44	0	1	1	0	0	11	0	1	2	2	100	0.44	0.537
CladoceraPenilia	0	0	0	55	6	1	4	0	0	9	7	18	0	100	0.55	0.626
Copepoda	0	0	0	10	43	12	0	0	2	5	12	16	0	100	0.43	0.703
Decapoda	4	0	1	12	21	22	0	0	2	5	7	23	3	100	0.22	0.651
Doliolida	0	0	0	13	0	0	63	0	0	9	1	14	0	100	0.63	0.182
Egg	0	0	0	0	0	1	0	91	0	1	0	7	0	100	0.91	0.081
Fiber	34	0	26	0	0	2	0	0	31	0	2	5	0	100	0.31	0.544
Gelatinous	0	0	0	16	9	2	4	5	0	29	7	28	0	100	0.29	0.574
Multiple	7	0	4	19	29	7	2	0	2	3	13	14	0	100	0.13	0.803
Nonbio	11	0	3	21	13	6	4	2	5	6	4	25	0	100	0.25	0.86
Pteropoda	11	0	10	0	19	7	0	0	3	1	3	13	33	100	0.33	0.175
Total	157	96	95	147	145	63	77	99	68	68	66	179	40	1300	0.5	0.493

图 19: 混淆矩阵（训练集与测试集不同）

5.1.4 C-SVC RBF

实验得到的 **re-substitution Confusion Matrix**（自身验证自身）如图 20:

对测试集进行预测得到的混淆矩阵如图 21:

5.2 LBP+SVM

SVM 核函数也选用了 Linear 核和 RBF 核。实验中，将图像划分为 16×16 的区域，统计每一个区域内的 LBP 直方图（直方图的 bin 设置为 256)。

	Appendicular	Bubble	Chaetognatha	CladoceraPenilia	Copepoda	Decapoda	Doliolida	Egg	Fiber	Gelatinous	Multiple	Nonbio	Pteropoda	Total	Recall	1-Precision
Appendicularia	545	0	0	0	0	0	0	0	0	0	0	0	0	545	1	0
Bubble	0	145	0	0	0	0	0	0	0	0	0	0	0	145	1	0
Chaetognatha	0	0	349	0	0	0	0	0	0	0	0	0	0	349	1	0
CladoceraPenilia	0	0	0	925	0	0	0	0	0	0	0	0	0	925	1	0
Copepoda	0	0	0	0	1474	0	0	0	0	0	0	0	0	1474	1	0
Decapoda	0	0	0	0	0	557	0	0	0	0	0	0	0	557	1	0
Doliolida	0	0	0	0	0	0	285	0	0	0	0	0	0	285	1	0
Egg	0	0	0	0	0	0	0	344	0	0	0	0	0	344	1	0
Fiber	0	0	0	0	0	0	0	0	304	0	0	0	0	304	1	0
Gelatinous	0	0	0	0	0	0	0	0	0	602	0	0	0	602	1	0
Multiple	0	0	0	0	0	0	0	0	0	0	635	0	0	635	1	0
Nonbio	0	0	0	0	0	0	0	0	0	0	0	3078	0	3078	1	0
Pteropoda	0	0	0	0	0	0	0	0	0	0	0	0	217	217	1	0
Total	545	145	349	925	1474	557	285	344	304	602	635	3078	217	9460	1	0

图 20: 混淆矩阵（训练集与测试集相同）

	Appendicularia	Bubble	Chaetognatha	CladoceraPenilia	Copepoda	Decapoda	Doliolida	Egg	Fiber	Gelatinous	Multiple	Nonbio	Pteropoda
Appendicularia	0	0	0	0	3	0	0	0	0	0	0	97	0
Bubble	0	3	0	0	0	0	0	0	0	0	0	97	0
Chaetognatha	0	0	0	0	0	0	0	0	0	0	0	100	0
CladoceraPenilia	0	0	0	0	1	0	0	0	0	0	0	99	0
Copepoda	0	0	0	0	37	0	0	0	0	0	0	63	0
Decapoda	0	0	0	0	3	0	0	0	0	0	0	97	0
Doliolida	0	0	0	0	1	0	0	0	0	0	0	99	0
Egg	0	0	0	0	0	0	0	0	0	0	0	100	0
Fiber	0	0	0	0	0	0	0	0	0	0	0	100	0
Gelatinous	0	0	0	0	9	0	0	0	0	0	0	91	0
Multiple	0	0	0	0	1	0	0	0	0	0	0	99	0
Nonbio	0	0	0	0	4	0	0	0	0	0	0	96	0
Pteropoda	0	0	0	0	6	0	0	0	0	0	0	94	0

图 21: 混淆矩阵（训练集与测试集不同）

5.2.1 C-SVC Linear

实验得到的 **re-substitution Confusion Matrix**（自身验证自身）如图 22:

	Appendicularia	Bubble	Chaetognatha	CladoceraPenilia	Copepoda	Decapoda	Doliolida	Egg	Fiber	Gelatinous	Multiple	Nonbio	Pteropoda	Total	Recall	1-Precision
Appendicularia	499	0	10	0	0	0	0	0	2	0	0	34	0	545	0.916	0.34
Bubble	0	145	0	0	0	0	0	0	0	0	0	0	0	145	1	0.02
Chaetognatha	25	0	324	0	0	0	0	0	0	0	0	0	0	349	0.928	0.122
CladoceraPenilia	0	0	0	698	15	2	0	0	0	0	7	203	0	925	0.755	0.39
Copepoda	3	0	1	30	858	18	0	0	1	0	25	538	0	1474	0.582	0.413
Decapoda	15	0	1	12	54	294	0	0	1	0	16	163	1	557	0.528	0.303
Doliolida	0	0	0	0	1	0	278	0	0	0	0	6	0	285	0.975	0.259
Egg	0	0	0	0	0	0	0	344	0	0	0	0	0	344	1	0.223
Fiber	64	0	19	0	0	0	0	0	221	0	0	0	0	304	0.727	0.166
Gelatinous	0	0	0	3	6	1	21	1	0	432	0	138	0	602	0.718	0.329
Multiple	73	1	10	25	74	29	2	0	23	7	177	214	0	635	0.279	0.425
Nonbio	77	2	4	377	442	73	74	98	17	204	83	1612	15	3078	0.524	0.456
Pteropoda	0	0	0	0	11	5	0	0	0	1	0	53	147	217	0.677	0.098
Total	756	148	369	1145	1461	422	375	443	265	644	308	2961	163	9460	0.739	0.273

图 22: 混淆矩阵（训练集与测试集相同）

对测试集进行预测得到的混淆矩阵如图 23:

5.2.2 C-SVC RBF

实验得到的 **re-substitution Confusion Matrix**（自身验证自身）如图 24:

	Appendicularia	Bubble	Chaetognatha	CladoceraPenilia	Copepoda	Decapoda	Doliolida	Egg	Fiber	Gelatinous	Multiple	Nonbio	Pteropoda	Total	Recall	1-Precision
Appendicularia	67	0	8	1	1	1	0	0	3	0	2	17	0	100	0.67	0.636
Bubble	0	96	0	0	1	0	0	2	0	0	0	1	0	100	0.96	0.01
Chaetognatha	35	0	50	0	0	0	0	0	6	0	0	9	0	100	0.5	0.474
CladoceraPenilia	0	0	0	62	6	1	0	0	0	0	3	28	0	100	0.62	0.496
Copepoda	0	0	0	5	37	6	0	0	0	0	3	49	0	100	0.37	0.678
Decapoda	3	0	1	7	17	25	0	0	0	0	3	44	0	100	0.25	0.51
Doliolida	0	0	0	11	0	0	54	0	0	9	1	25	0	100	0.54	0.1
Egg	0	0	0	0	0	2	0	89	0	0	0	9	0	100	0.89	0.082
Fiber	48	0	28	0	0	0	0	0	18	0	0	6	0	100	0.18	0.419
Gelatinous	1	0	0	12	3	3	4	4	0	45	0	28	0	100	0.45	0.262
Multiple	12	0	4	10	16	4	0	0	1	3	6	44	0	100	0.44	0.714
Nonbio	12	1	2	14	11	4	2	2	2	4	1	45	0	100	0.45	0.866
Pteropoda	6	0	2	1	23	5	0	0	1	0	2	32	28	100	0.28	0
Total	184	97	95	123	115	51	60	97	31	61	21	337	28	1300	0.51	0.404

图 23: 混淆矩阵（训练集与测试集不同）

	Appendicularia	Bubble	Chaetognatha	CladoceraPenilia	Copepoda	Decapoda	Doliolida	Egg	Fiber	Gelatinous	Multiple	Nonbio	Pteropoda	Total	Recall	1-Precision
Appendicularia	545	0	0	0	0	0	0	0	0	0	0	0	0	545	1	0
Bubble	0	145	0	0	0	0	0	0	0	0	0	0	0	145	1	0
Chaetognatha	0	0	349	0	0	0	0	0	0	0	0	0	0	349	1	0
CladoceraPenilia	0	0	0	925	0	0	0	0	0	0	0	0	0	925	1	0
Copepoda	0	0	0	0	1000	0	0	0	0	0	0	474	0	1474	0.67	0
Decapoda	0	0	0	0	0	557	0	0	0	0	0	0	0	557	1	0
Doliolida	0	0	0	0	0	0	285	0	0	0	0	0	0	285	1	0
Egg	0	0	0	0	0	0	0	344	0	0	0	0	0	344	1	0
Fiber	0	0	0	0	0	0	0	0	304	0	0	0	0	304	1	0
Gelatinous	0	0	0	0	0	0	0	0	0	602	0	0	0	602	1	0
Multiple	0	0	0	0	0	0	0	0	0	0	635	0	0	635	1	0
Nonbio	0	0	0	0	0	0	0	0	0	0	0	3078	0	3078	1	0.13
Pteropoda	0	0	0	0	0	0	0	0	0	0	0	0	217	217	1	0
Total	545	145	349	925	1000	557	285	344	304	602	635	3552	217	9460	0.97	0.01

图 24: 混淆矩阵（训练集与测试集相同）

5.3 SIFT+BoW+SVM

5.4 实验总结

5.4.1 遇到的问题

1. SIFT 实验结果不对，几乎全都分类错误。

2. 交叉验证代码没有找到，找到的交叉验证代码都不是 MATLAB 和 C 的，没有跑通。

3. 没有来得及调参数，对参数的影响效果也不是很了解。

4. 用测试集跑出来的实验结果在 Multiple 和 Nonbio 类上效果最差，Bubble 和 Egg 上效果最好。

5.4.2 方案

20150706~20150710 用三个经典特征（HOG, SIFT, LBP）和 SVM 分类器分别完成对浮游动物的分类，并对分类结果进行评价分析。

20150713~20150717 根据我们上周遇到的问题，决定在这周对这些问题进行进一步研究：

1. 用 MATLAB 版的 SIFT 和 bag of word 进行实验。

2. 调整特征以及 SVM 分类器的参数，分析对效果的影响；用交叉验证对 SVM 进行优化，分析效果。

3. 打算索要完整的 20 类浮游动物的数据集。

4. 调试交叉验证的程序。

20150720~20150724 仍然用三个经典特征（HOG、SIFT、LBP），分类器换为随机森林和 ELM 进行实验。

在回家之前，初步实现一套浮游动物分类系统。

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.

[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] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[5] 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.

[6] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 985–990. IEEE, 2004.

[7] 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.

[8] 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.

[9] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[10] 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.

[11] 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.

[12] Daniel Reisfeld, Haim Wolfson, and Yehezkel Yeshurun. Context-free attentional operators: the generalized symmetry transform. *International Journal of Computer Vision*, 14(2):119–130, 1995.

[13] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[14] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: many could be better than all. *Artificial intelligence*, 137(1):239–263, 2002.