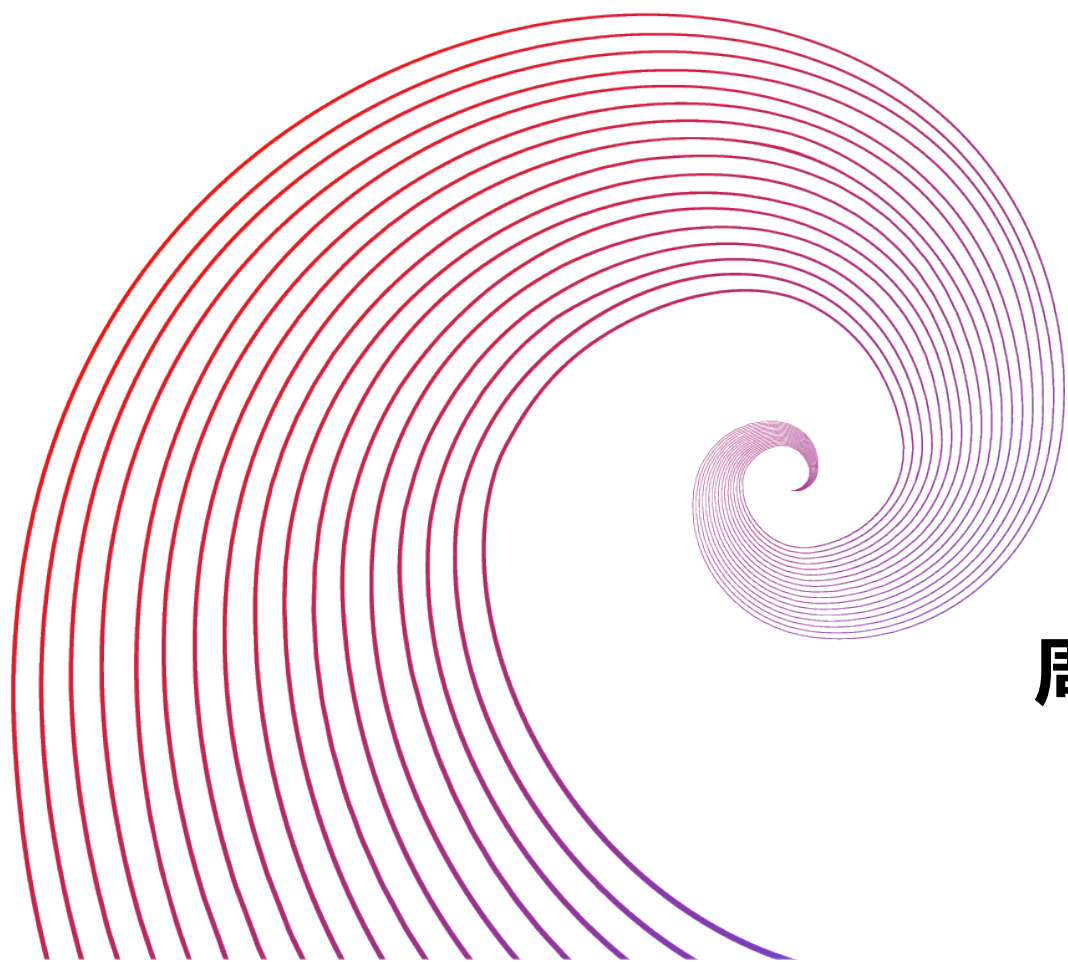
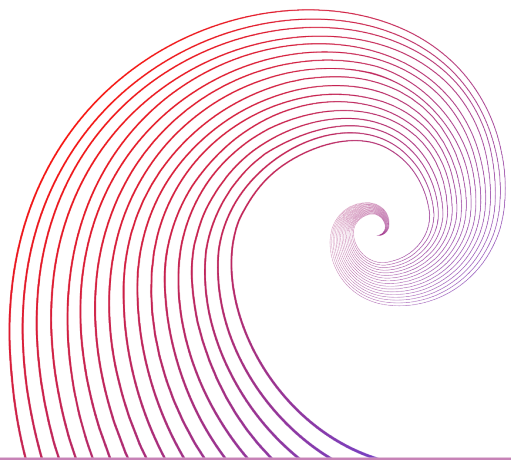


基于惯性大水滴滴水算法和支持向量机的粘连字符验证码识别



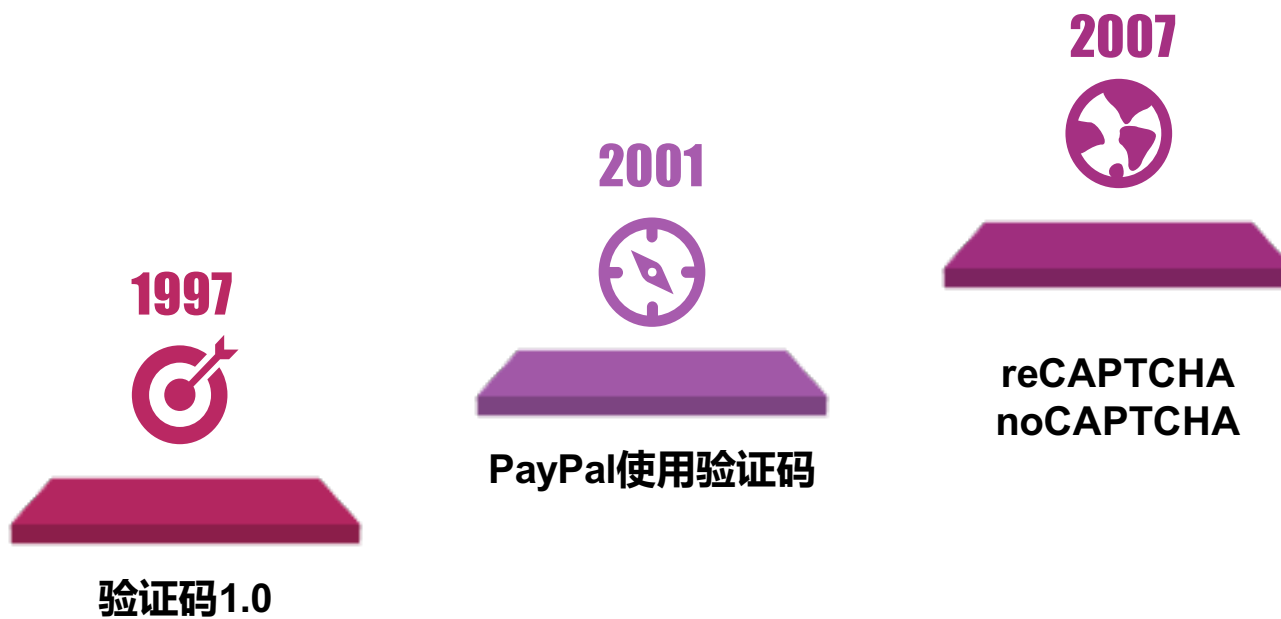
周慧文 白恒瑞 徐一博 赖其才



01

验证码简介

验证码发展历史



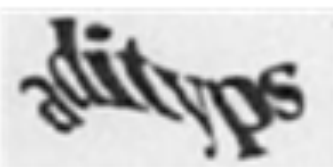
短语验证码类型

多字体



805301

旋转扭曲



adityps

空心



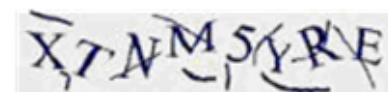
idMAspVd

双层



19F/99
qxw2c

噪线



X7NM,SYRE

面临的攻击：

- OCR：通过检测暗、亮的模式确定字符形状，然后用字符识别方法将形状翻译成计算机文字的过程
- 字符分割：把多行或多字符图像中的每个字符从整个图像中分割出来,成为单个字符
- 深度学习：学习抽象验证码特征

图片分割技术

01

竖直投影法

- 最简单的字符分割法，常用于字符没有粘连在一起的情况，如：车牌识别
- 首先将二值化的图像进行水平或竖直投影，根据相邻的波谷或极大极小值来确定字符边界，进行分割

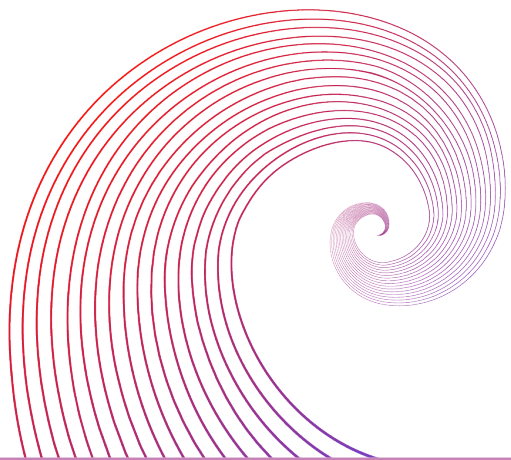


gehyX gehyX
g e h y X

CFS连通域分割法

- 基于字符的连通特征，即每个字符或者其一部分由单独的一个连通域组成
- 将不同的连通域加以标记，就可以分割出单独的字符
 - 要求字符不粘连

02



02

验证码预处理



预处理步骤



灰度化

彩色图像的像素点由R、G、B、三原色组成，其值在0-255之间。当 $R=G=B$ 时，则彩色表示一种灰度颜色。灰度图像是指只含有亮度信息，不含彩色信息的图像。



去噪

现实中的数字图像在数字化和传输过程中常受到成像设备与外部环境噪声干扰等影响，称为含噪图像或噪声图像。图像去噪是指减少数字图像中噪声的过程称为图像去噪。



二值化

二值化是图像分割的一种方法。在二值化图像的时候把大于某个临界灰度值的像素灰度设为灰度极大值，把小于这个值的像素灰度设为灰度极小值，从而实现二值化。

灰度化

✓ **分量法**

$$R=G=B = R \text{ or } G \text{ or } B$$

✓ **最大值法**

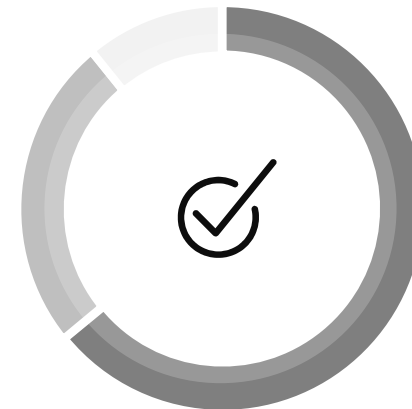
$$R=G=B = \text{Max} (R + G + B)$$

✓ **平均值法**

$$R=G=B = (R + G + B) / 3$$

✓ **加权平均值法**

$$R=G=B = 0.30 R + 0.59 G + 0.11 B$$



灰度化

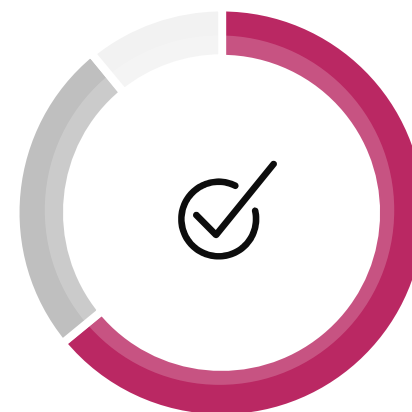
去噪

✓ 均值滤波

也称线性滤波，主要采用邻域平均法。均值滤波的核心思想是其将整个图像看成是由很多灰度恒定的小块组成，相邻像素间相关性很强，但噪声具有统计独立性，故可用邻域的均值替代原图像中的各个像素值。

✓ 中值滤波

中值滤波是基于排序统计理论的一种能有效抑制噪声的非线性信号处理技术，是把数字图像或数字序列中一点的值用该点的一个邻域中各点值的中值代替，让周围像素灰度值的差比较大的像素改取与周围的像素值接近的值，从而可以消除孤立的噪声点。



去噪

二值化

✓ 双峰法

灰度级直方图呈明显的双峰状，则选取两峰之间的谷底所对应的灰度级作为阈值。

✓ P参数法

所谓P分位法图像分割，就是在知道图像中目标所占的比率Ratio时，循环不同的灰度值对图像进行分割，使目标所占的比率与Ratio的差值足够小。

✓ 迭代法

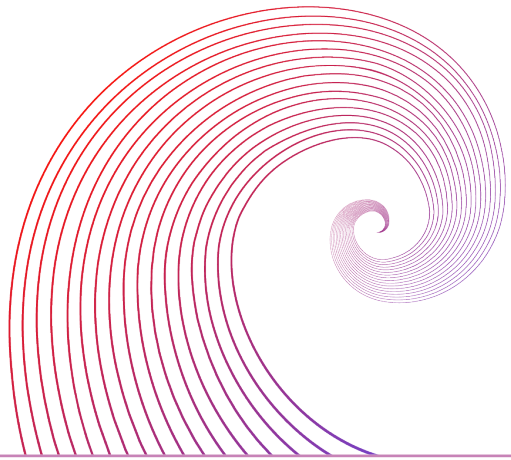
首先，初始化一个阈值，然后按照某种策略通过迭代不断更新这一阈值，直到满足给定的约束条件为止。

✓ OSTU法图像二值化

根据初始阈值把图像分为两类，然后计算两类之间的方差，更新阈值并重新计算类间方差，直到得到满足类间方差最大时的阈值。



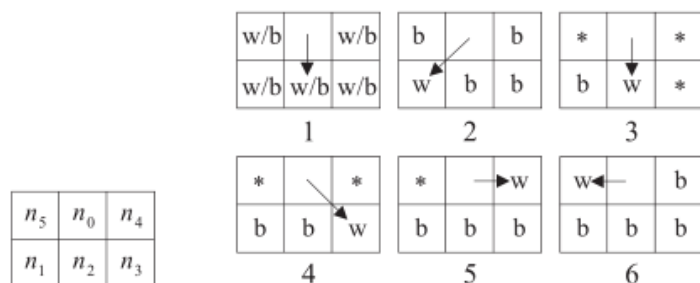
二值化



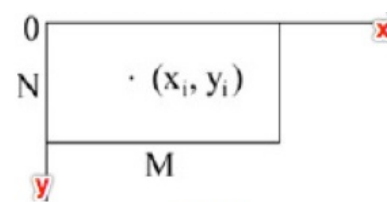
03

滴水算法

传统滴水算法



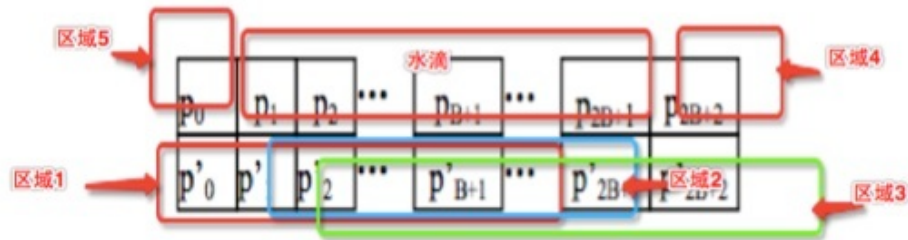
(a)水滴周围像素编号 (b)水滴下一滴落位置的选择



$$w_j = 6 - j, (j = 1, 2, 3, 4, 5) \Rightarrow W = \begin{cases} 4, \sum_{j=1}^5 z_j w_j = 0 \text{ or } 15 \\ \max_{j=1}^5 z_j w_j, \text{其他} \end{cases}$$

$$\begin{cases} (x_i - 1, y_i) & \text{if } W_i = 1 \\ (x_i + 1, y_i) & \text{if } W_i = 2 \\ (x_i + 1, y_i + 1) & \text{if } W_i = 3 \\ (x_i, y_i + 1) & \text{if } W_i = 6 \\ (x_i - 1, y_i + 1) & \text{if } W_i = 5 \\ (x_i + 1, y_i + 1) & \text{if } W_i = 4 \text{ and } \bar{I}_i = \bar{V}_{04} \\ (x_i, y_{i+1}) & \text{if } W_i = 4 \text{ and } \bar{I}_i = \bar{V}_{05}, 0 \end{cases}$$

大水滴惯性滴水算法



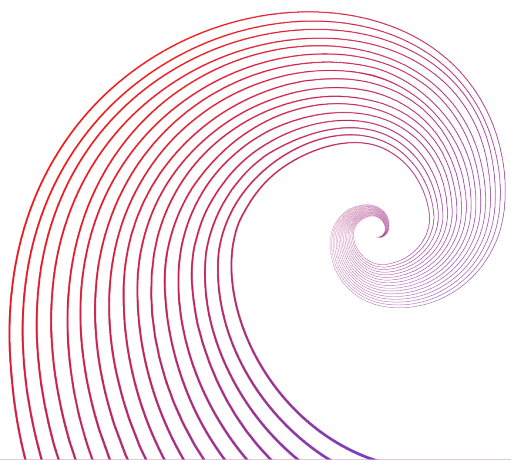
$$\Sigma = \sum_{j=1}^5 z'_j w_j$$

$$W_i = \begin{cases} 4 \\ 6 \\ \max_{j=1}^5 z'_j w_j \end{cases}$$

$$Z_j = \begin{cases} \max V(P^{j-1}, P'_{2B-1+j}) & j=1, 2, 3 \\ V(P_{2B+2}) & j=4 \\ V(P_0) & j=5 \end{cases}$$

if $\Sigma = 0$
if $\Sigma = 15$
else

$$\begin{cases} S(x_i - B - 1, x_i + B - 1, y_i) & \text{if } W_i = 1 \\ S(x_i - B + 1, x_i + B + 1, y_i) & \text{if } W_i = 2 \\ S(x_i - B + 1, x_i + B + 1, y_i + 1) & \text{if } W_i = 3 \\ S(x_i - B, x_i + B, y_i + 1) & \text{if } W_i = 6 \\ S(x_i - B - 1, x_i + B - 1, y_i + 1) & \text{if } W_i = 5 \\ S(x_i - B + 1, x_i + B + 1, y_i) & \text{if } W_i = 4 \text{ and } \bar{I}_i = \bar{V}_{04} \\ S(x_i - B, x_i + B, y_i + 1) & \text{if } W_i = 4 \text{ and } \bar{I}_i = \bar{V}_{05}, 0 \end{cases}$$



04

图片识别

支持向量机简介

01

是机器学习方法之一

- 广泛运用于各种分类器中，如自然语言处理、数据挖掘、图形处理

性能良好、准确度高、泛化力强

- 没有神经网络中存在的局部极小值点的问题

02

01

基本模型

- 特征空间上的间隔最大的线性分类器，其学习策略是间隔最大化，最终转化为一个凸二次规划问题的求解

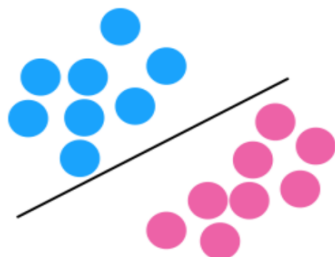
简要过程

01



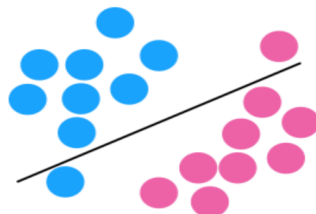
- 假设现在二维平面上有一堆红球和蓝球，我们需要将其分类

02



- 从图上看，我们很容易就能用一条直线将其分类.

03



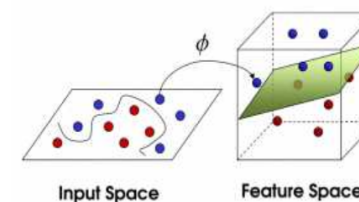
- 但是如果我们在这其中加小球，这根线就很可能不能分类了，所以最好是采用直线分法的时候，尽量采用直线两边没有小球的直线

04



- 但是有时我们会碰到无法用直线进行分类的情况

05



- 这个时候，我们将二维平面上的小球转化为三维平面上的小球，进行空间的平面的切割分类，能够应对这种情况

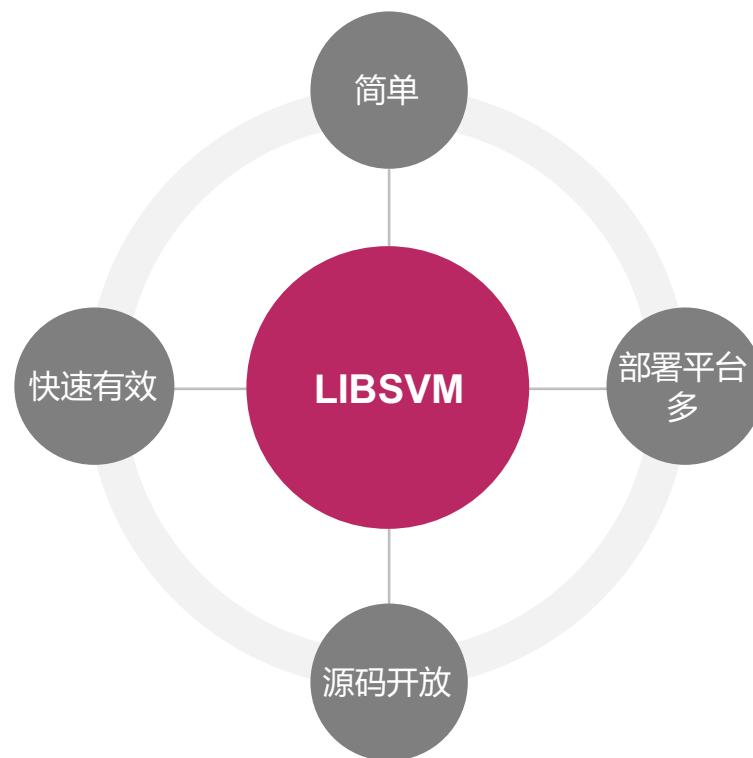
LIBSVM简介

在高维空间中非常高效

即使在数据维度比样本数量大的情况下仍然有效

在支持向量中使用的是训练集的子集,因此它也是高效利用内存的

通用性: 不同的核函数与特定的决策函数一一对应. 常见的内核已经提供, 也可以指定定制的内核.



具体流程

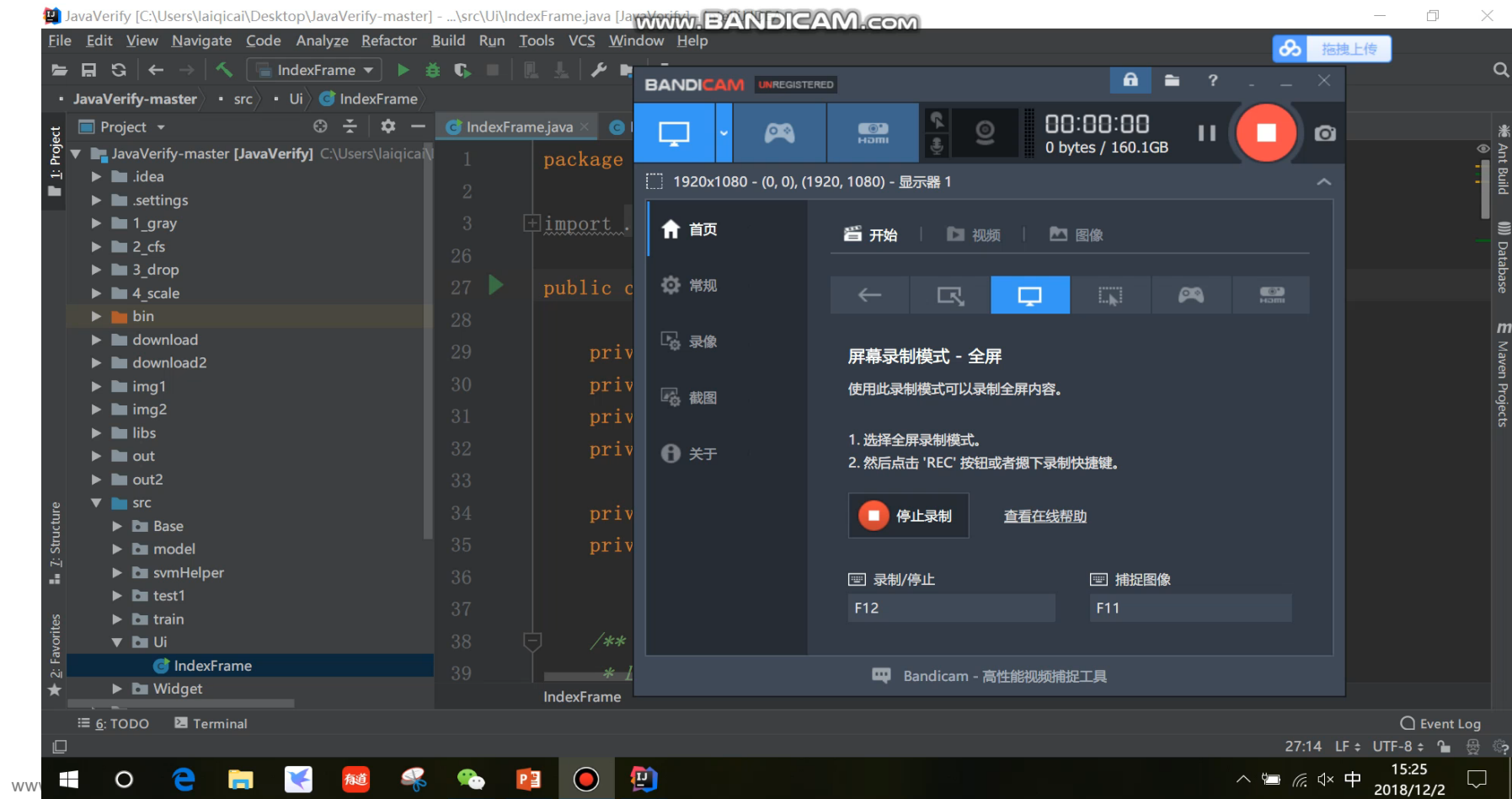
- › 把图片二值化、切割、产生字符块，然后人工识别，并标注上类标号，这一步比较耗时耗力，需要训练上万张图，才能达到90%以上的准确率

- › 对标注好类标号的图片按指定的宽高进行缩放，这一步是为了统一图片大小，使特征向量的维数保持一致，加大分类的准确性

- › 对缩放后的图片进行特征提取，由于是二值化的图片，因此其特征值只有0或者1，将每个像素进行标号，如16X16的图片可标记为256个标号，对应256个像素点，然后调用svm_train生成模型

- › 将特征保存为LibSvm的数据格式，调用LibSvm的svm_predict使用生成的模型对图片进行预测，并保存结果

演示



Thanks

