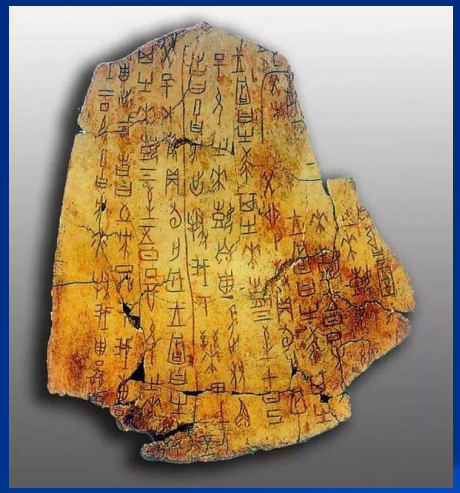
第四讲 笔式交互技术

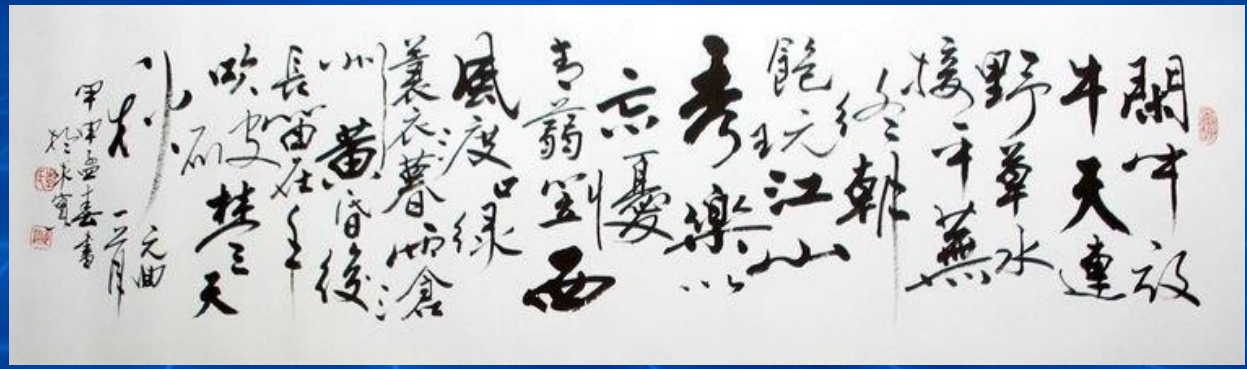
华东师范大学 王长波

提　纲

- 笔式交互的简介
- 笔式交互的技术
  - 笔式交互预处理
  - 传统汉字识别技术
  - 深度学习汉字识别
- 笔式交互的评测

文字是信息交流与感知世界的重要手段





汉字/字符的主要输入方式

- 人工方式
  - 键盘输入
  - 手写输入
  - 语音输入
- 自动输入
  - 印刷体汉字识别
  - 脱机手写汉字识别
  - 场景文字自动识别

笔式交互是字符信息最重要的交互方式！

笔式交互的应用领域

- 智能终端
  - 智能手机、平板电脑
  - 手写输入、文字录入及检索
- 文化知识
  - 书法识别
  - 数字图书馆
- 信息查询
  - 查询系统
  - 导航系统
- 大数据服务
  - 图文的挖掘
  - 文字的搜索
- 金融行业
  - 自主开户、票据识别
  - 移动金融、移动理财
- 教育产业
  - 板书、笔记
  - 题库搜索解答
- 智慧城市
  - 交通管理
  - 电子表格管理
- 智慧政务
  - 随申办
  - 税务、邮政、海关

字符识别是笔式交互的关键

- 按照识别对象的特征分类
  - 印刷体字符识别
  - 手写体字符识别
- 手写体字符识别
  - 脱机字符识别（off-line）
    - 机器用扫描方式识别已经写好的文本
  - 联机字符识别（on-line）
    - 用笔在输入板上写，边写边认

汉字/字符识别分类

- 技术难度比较
  - 西文识别
    - 比汉字容易：汉字类别、变化多
    - 汉字类别：楷书，(行楷)，行书，(行草)，草书
  - 印刷体识别
    - 相对手写体更容易
    - 已经有了大量实际应用，图书馆数字化
  - 联机手写体识别
    - 相对容易，PDA等的推广，大量应用
  - 脱机手写体识别——最难
    - 脱机手写体数字识别，如邮政编码的自动识别
    - 汉字等文字的脱机手写体识别还处在实验室阶段

联机手写识别的优势

- 【与脱机手写相比】具有
  - 笔划时间信息
  - 空间形状信息
  - 良好的可交互性
    - 用户可以修正错误
  - 可适应性
    - 系统适应：根据用户修正的错误进行学习
    - 用户适应：可以改变书写习惯

笔式交互信号采集方式

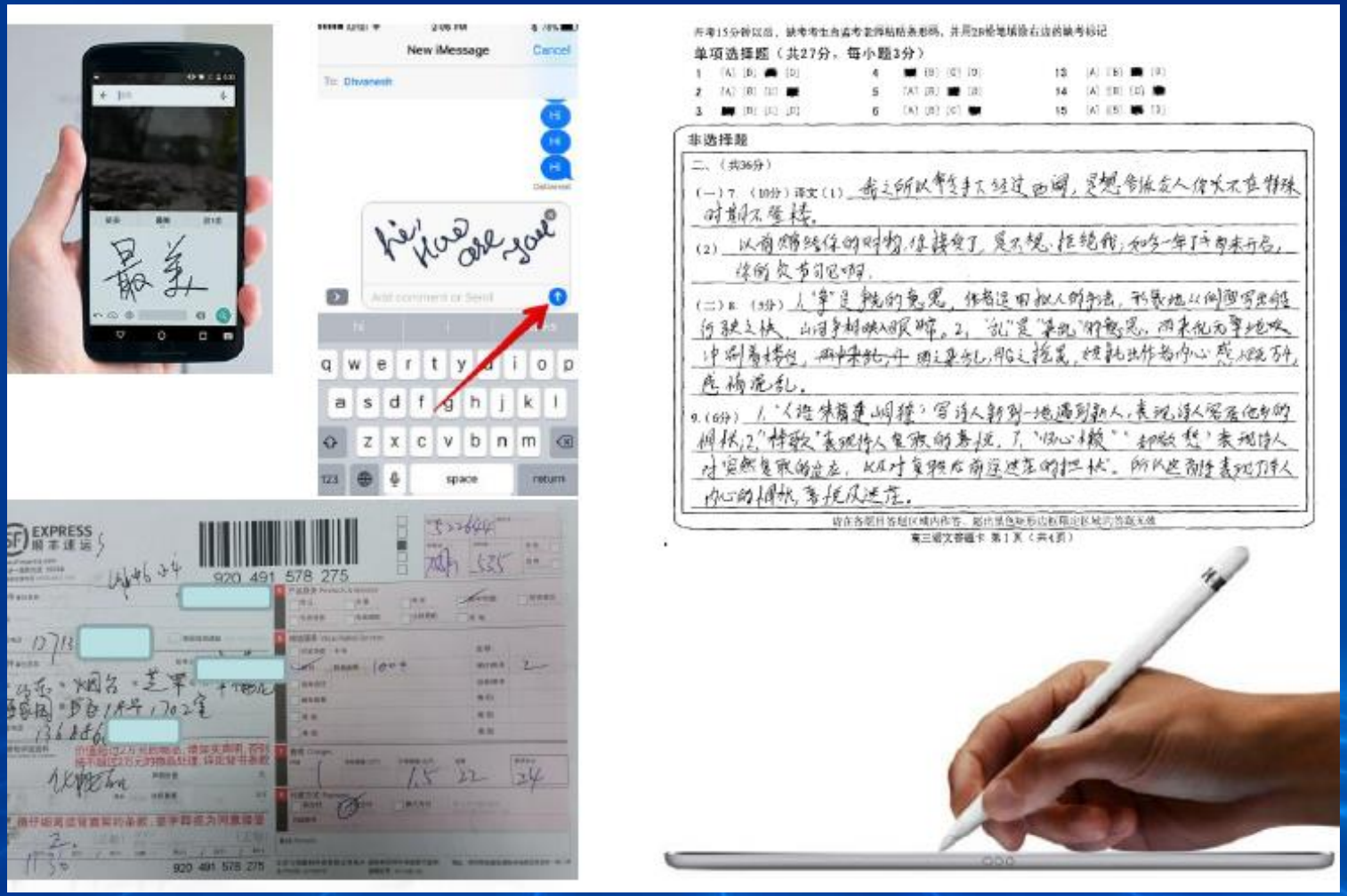
- 脱机字符识别
  - 扫描仪或者摄像设备
    - 数字图像信号
- 联机手写识别
  - 手写屏，手写输入板
    - 运动轨迹电信号，记录了笔划和笔顺信息
    - 电磁式或压电式，在书写时，笔在板上的运动轨迹（在板上的坐标）被转化为一系列的电信号，电信号可以串行地进入到计算机中，从这些电信号我们可以比较容易地抽取笔划和笔顺的信息

提　纲

- 笔式交互的简介
- 笔式交互的技术
  - 笔式交互预处理
  - 传统汉字识别技术
  - 深度学习汉字识别
- 笔式交互的评测

一、笔式交互预处理

笔式交互的识别（OCR）


New iMessage
Cancel
To: Dhvanesh
Hi
Hi
Hi
Delivered
Hi! How are you?
Add comment or Send
space
return
EXPRESS
920 491 578 275
920 491 578 275
单项选择题（共27分，每小题3分）
非选择题
二、（共36分）

OCR技术发展历史

- 西文OCR技术研究始于50年代
  - Optical Character Recognition (OCR)
  - 几乎所有的早期模式识别研究者都进行过字符识别的研究。随后的 30 多年来，字符识别一直是模式识别的重要内容之一
- 汉字OCR技术
  - 印刷体汉字的识别最早可以追溯到60年代
  - 1966年，IBM公司的Casey和Nagy发表了第一篇关于印刷体汉字识别的论文，在这篇论文中他们利用简单的模板匹配法识别了1，000个印刷体汉字

汉字OCR技术发展历史

- 70年代以来，日本人做了许多工作
  - 日本的常用汉字有2000个左右
  - 1977年东芝综合研究所研制了可以识别2000个汉字的单体印刷汉字识别系统
  - 80年代初期，日本武藏野电气研究所研制的可以识别2300个多体汉字的印刷体汉字识别系统，代表了当时汉字识别的最高水平
  - 日本的三洋、松下、理光和富士等公司也有其研制的印刷汉字识别系统
- 简评
  - 这些系统在方法上，大都采用基于KL数字变换的匹配方案，使用了大量专用硬件，其设备有的相当于小型机甚至大型机，价格极其昂贵，没有得到广泛应用

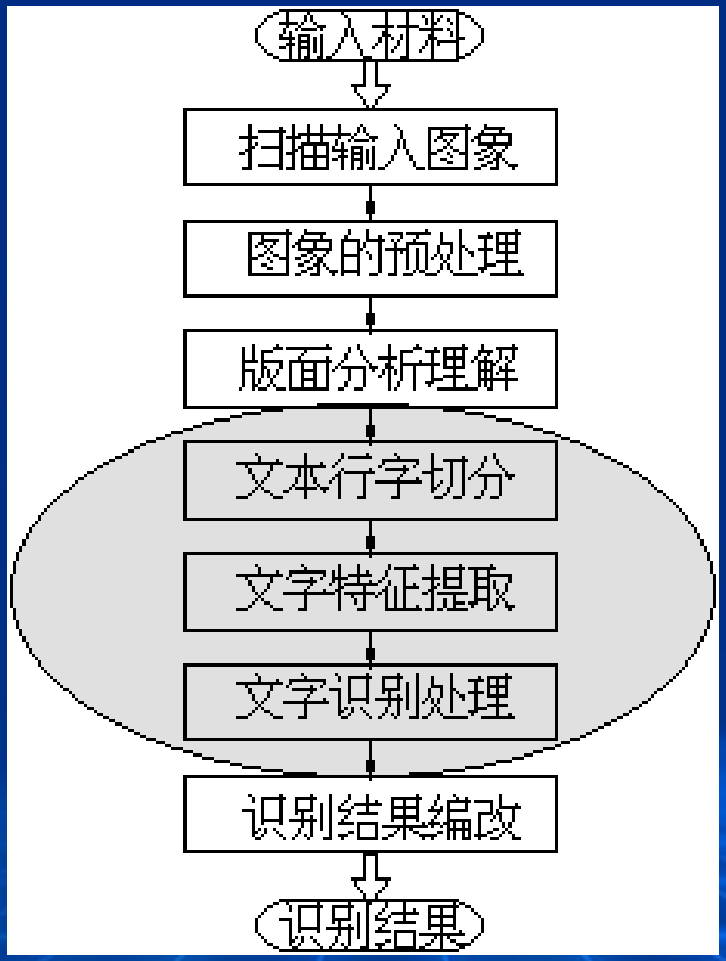
汉字OCR技术发展历史

- 我国自70年代后期开始字符识别方面的研究
- 80年代以后,台湾和香港发展的也很快
- 发展历程
  - 70年代末期到80年代末期
    - 算法和方案探索 ： 单体汉字识别
  - 90年代初期
    - 由实验室走向市场，初步实用
  - 90年代后期——混排
    - 多语言混排文本：如中英文
    - 多字体混排文本：如：宋，楷体，…）
    - 多字号混排文本：不同大小

当前进展状态

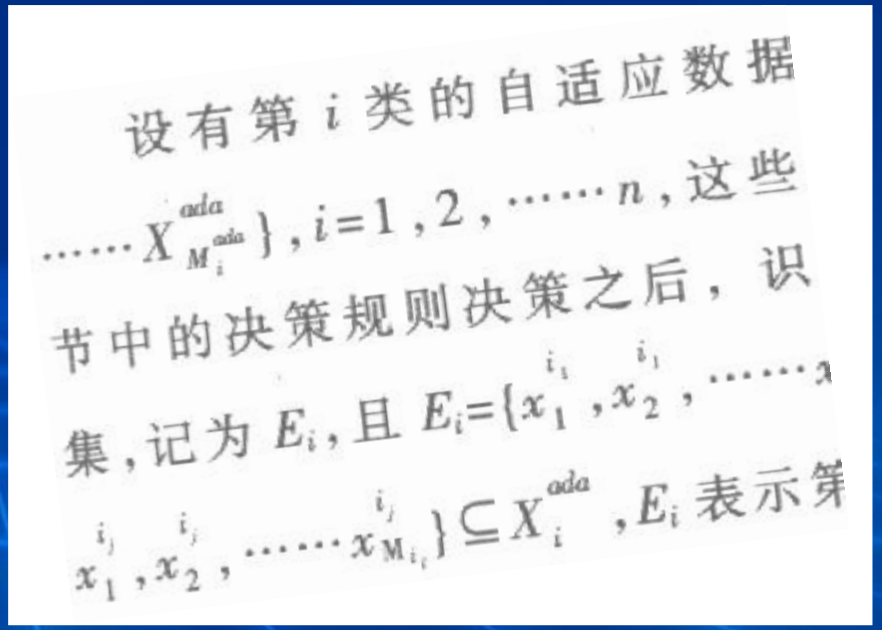
- 2000年代后
  - 识别率、鲁棒性的提高
  - 单纯OCR→文档分析
    - 多语混排，多字号，多字体
    - 版面分析
      - 文本的结构
      - 表格，图像（如插图），公式
  - 摄像设备（非扫描仪）
    - 名片→手机摄像→通讯录

OCR技术一般流程


输入材料
扫描输入图象
图象的预处理
版面分析理解
文本行字切分
文字特征提取
文字识别处理
识别结果编改
识别结果

流程简介

- （1）图像获取
  - 将文本转换为图象点阵
    - 扫描仪 (Scanner)
    - 其它光电扫描设备
      - 如传真机，摄象机
  - 不同获取设备的差异
    - 扫描仪：最优
    - 摄像机：识别难度大



设有第 i 类的自适应数据

$\cdots\cdots X_{M_i^{ada}}^{ada}\}, i=1,2,\cdots\cdots n$，这些

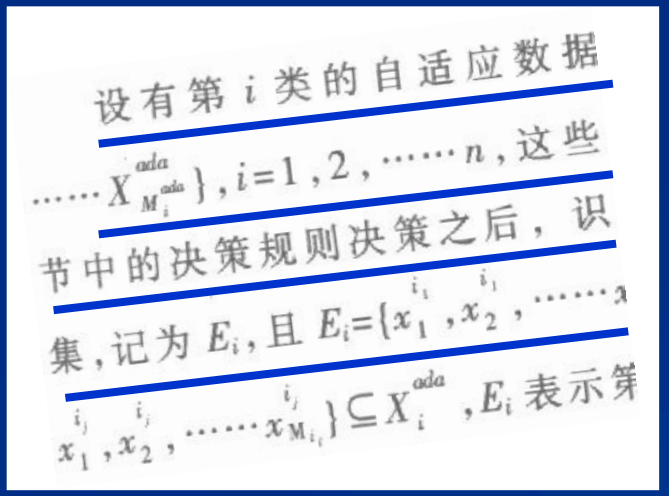
节中的决策规则决策之后，识

集，记为 E_i，且 $E_i=\{x_1^{i_1}, x_2^{i_1}, \cdots\cdots x$

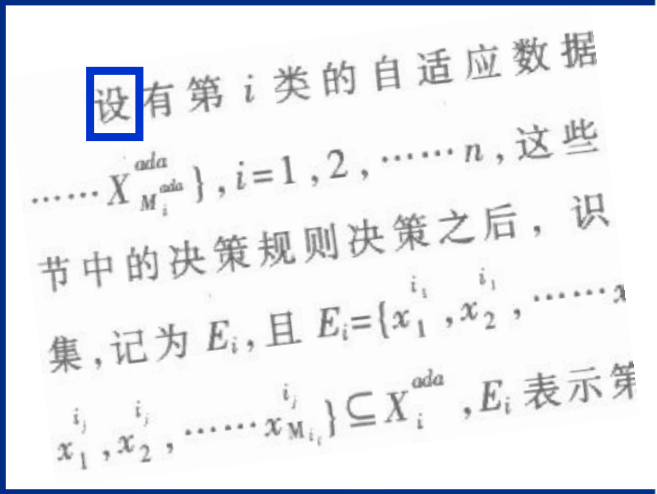
$x_1^{i_j}, x_2^{i_j}, \cdots\cdots x_{M_{i_j}}^{i_j}\} \subseteq X_i^{ada}$，$E_i$ 表示第

流程简介

- （2）图像预处理
  - 滤除干扰噪声
  - 倾斜校正
  - 各种滤波处理
- （3）版面分析
  - 完成对于文本图象的总体分析
  - 区分出文本段落及排版顺序，图象、表格的区域
    - 对于文本区域将进行识别处理
    - 对于表格区域进行专用的表格分析及识别处理
    - 对于图象区域进行压缩或简单存储。


设有第 i 类的自适应数据
$\cdots\cdots X_{M_i^{ada}}^{ada}\}$，$i=1,2,\cdots\cdots n$，这些
节中的决策规则决策之后，识
集，记为 E_i，且 $E_i=\{x_1^{i_1},x_2^{i_1},\cdots\cdots$
$x_1^{i_j},x_2^{i_j},\cdots\cdots x_{M_{i_j}}^{i_j}\}\subseteq X_i^{ada}$，$E_i$ 表示第

流程简介


设有第 i 类的自适应数据
$\cdots\cdots X_{M_i^{ada}}^{ada}\}$，$i=1,2,\cdots\cdots n$，这些
节中的决策规则决策之后，识
集，记为 E_i，且 $E_i=\{x_1^{i_1},x_2^{i_1},\cdots\cdots$
$x_1^{i_j},x_2^{i_j},\cdots\cdots x_{M_{i_j}}^{i_j}\}\subseteq X_i^{ada}$，$E_i$ 表示第

- （4）行字切分
  - 将大幅的图象先切割为行
  - 从图象行中分离出单个字符

- （5）特征提取
  - 整个环节中最重要的一环，提取的特征的稳定性及有效性，直接决定了识别的性能
  - 从单个字符图象上提取统计特征或结构特征
    - 包括细化(Thinning)，归一化(大小等)等步骤

流程简介

- （6）文字识别
  - 模式识别研究范畴
  - 从学习得到的特征库中找到与待识字符相似度最高的字符类
- （7）后处理
  - 利用词义、词频、语法规则或语料库
  - 上下文联想等语言先验知识进行校正

预处理方法——降噪

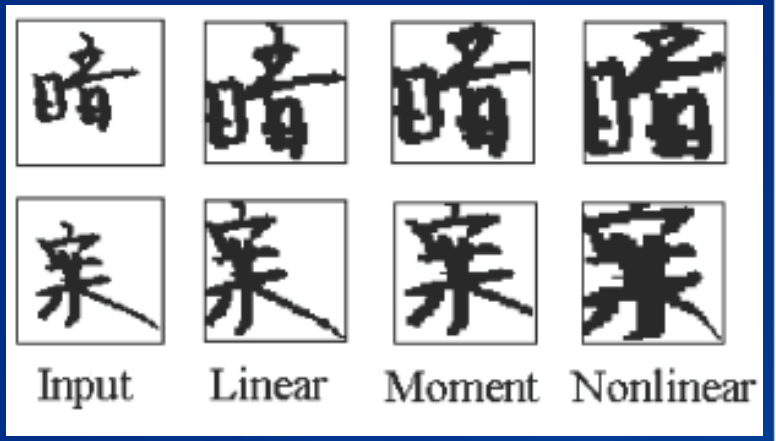
- 噪声来源
  - 书写时手的“飘忽”移动
  - 数字化的不准确性
- 降噪方法
  - 平滑，滤波，野点校正
  - (断)笔画连接

预处理方法——降维

- 降维
  - 轨迹线等距重采样（equidistance sampling）
    - 重采样以使得相邻点距离相等
    - 简单，但维度降低不大
  - 特征点检测（line approximation）
    - 检测：角点，笔划端点
- 角点检测典型方法
  - 曲率法
    - 估计每个点的曲率，保留那些高曲率点
  - 多项式近似
    - 递归的寻找具有最大“点弦距离”的顶点

预处理方法——归一化

- 将字符轨迹归一到一个标准尺寸
- 线性归一化
  - 方法1：平移和尺度变换
  - 方法2：距归一化
    - 一阶中心矩固定为标准框的中心
    - 二阶矩则固定为某标准值
- 非线性归一化
  - 根据线密度分布重新分配轨迹点坐标，以使得笔画空间分布均匀(equalizing the stroke spacing)



二、传统汉字识别技术

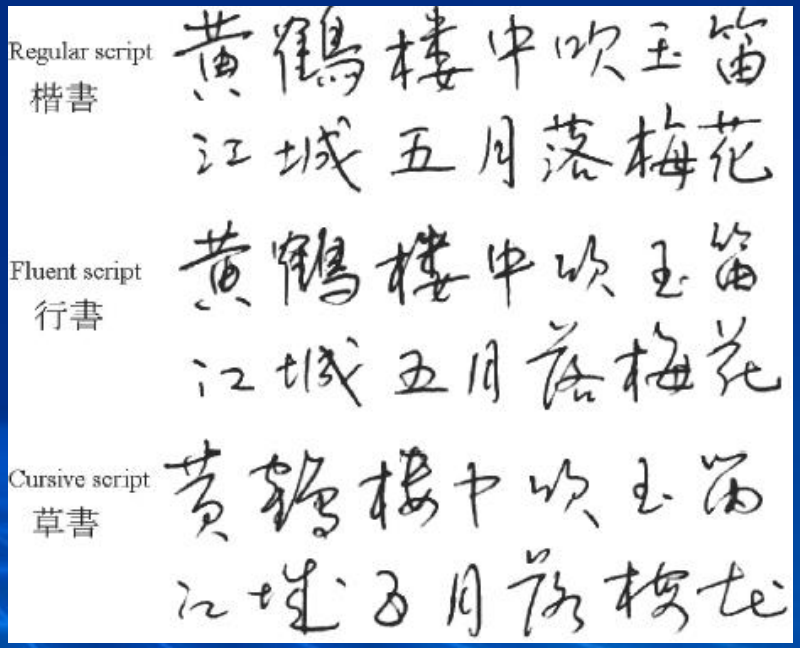
句法结构模式识别
统计特征模式识别
句法+统计混合方法

汉字字符集

- 简体中文（大陆，新加坡，香港）
  - GB2312-80
  - 一级汉字：3755个
  - 二级汉字：6763个（含一级）
- 台湾地区
  - 5401个繁体字
- 日本
  - JIS Level-1 2965个汉字
  - JIS Level-2 3390个汉字
  - 互不重叠

汉字特性

- 字形特征
  - 象形表意文字（ideograph）
  - 由直线、曲线等线条组成
  - 有子结构
    - 字根：字根编码可以减少模型数，加快识别速度
- 手写体类型
  - 楷书：Regular script
  - 行书：Fluent script
  - 草书：Cursive script



汉字识别的现状

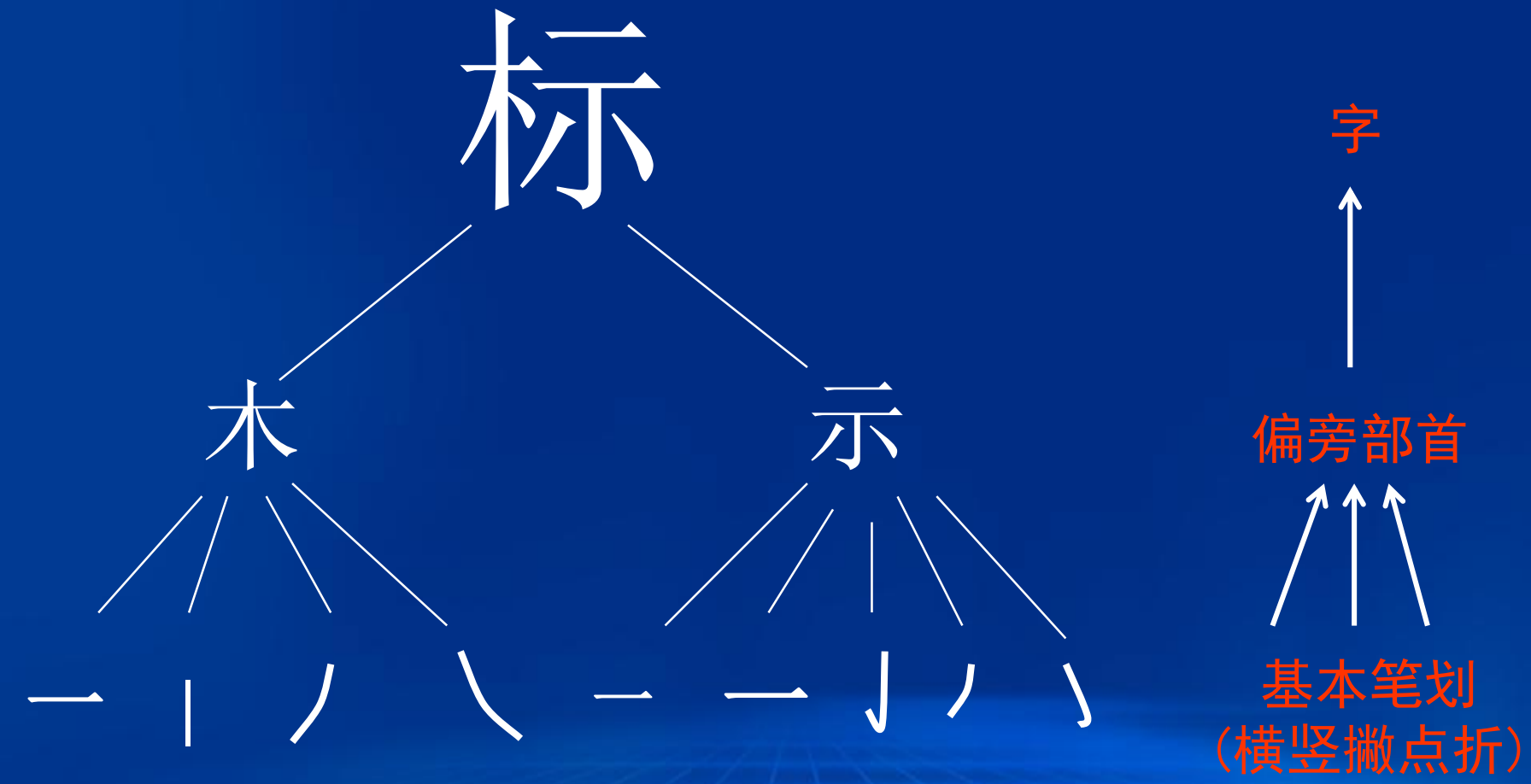
- 研究趋势
  - 放松强加给书写人的各种约束
    - 单字书写（不能连书）
    - 尽可能与标准楷体一致
  - 主要难题
    - 笔顺、笔划数的差异
  - 研究目标变迁
    - 从楷书到草书：笔顺、笔划数的更大变化
- 当前进展
  - 当前系统对楷体的识别率非常高：98%
  - 行书识别仍然没有完全解决
  - 草书的识别还有很长的路要走

汉字识别的方法

- 识别方法
  - 基于笔划句法的结构方法
    - 结构(structural)方法：stroke analysis
    - 笔顺无关的结构方法：stroke-order free
    - 笔顺依赖的结构方法：stroke-order dependent
  - 基于整体形状的统计方法
    - 统计方法：statistical methods
    - 自然是与笔顺无关的（stroke-order independence）
- 从应用角度
  - 书写人依赖系统：更容易一些
  - 书写人无关系统：更有挑战性

（一）基于句法的汉字识别

- 早期汉字识别研究的主要方法
- 其主要出发点是汉字的组成结构
  - 汉字图形结构复杂，但规律性强，含丰富的结构信息
    - 从汉字的构成上讲
    - 笔划(横竖撇点折)→偏旁部首→字
    - 由这些基元及其相互关系完全可以精确地对汉字加以描述
- 类比
  - 类比文章结构
    - 单字→词→短语→句子→篇章，按语法规律组成
  - 识别过程：编译理论中的句法分析



汉字的解析图表示示例

（一）基于句法的汉字识别

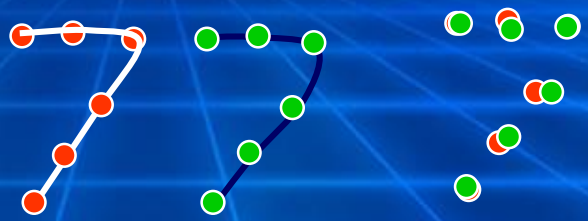
- 训练过程
  - 建立所有汉字的解析图描述
    - 基本单元
    - 基本单元之间的拓扑结构
- 识别过程
  - 图像获取、预处理、二值化
  - 基元提取
    - 基本笔画提取
    - 偏旁部首提取
    - 解析图表示
- 相似度计算
  - 句法分析过程
    - Top-down
  - 图匹配过程
    - 拓扑相似性
    - 节点相似性

1、汉字的结构表示

- 五个层次的结构
  - 上层由下层的“基元”构成
  - 特征点 == 线段
    - 两个特征点决定一个线段
  - 笔划编码 — 笔划模型
    - 笔划类型
    - HMM通常用来建模笔划或子笔划
  - 关系结构 — 汉字或字根模型
    - 建模笔划之间的结构关系
  - 层次结构
    - 模型数据库的层级结构：共享大量字根，建立所有字符的的表示模型
    - 输入模式的层级结构：以字根为基元的关系结构

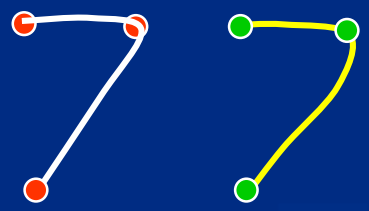
（1）采样点表示

- 直接采用“重采样的轨迹点序列”作为表示方法
  - 输入模式：重采样点序列
  - 字符原型：重采样点序列
- 比对方法
  - 对点序列进行对齐，然后通过点距离计算笔划之间的距离即可



（1）特征点/线段表示

- 特征点表示
  - 特征点的序列
  - 特征点 != 采样点
  - 可以与采用特征点序列、笔划编码、层级结构表示的字符原型作匹配



- 线段表示
  - 可以与采用线段或者更高层结构表示的字符原型作匹配
- 目前已被广泛采用的表示方法

（2）笔划编码表示

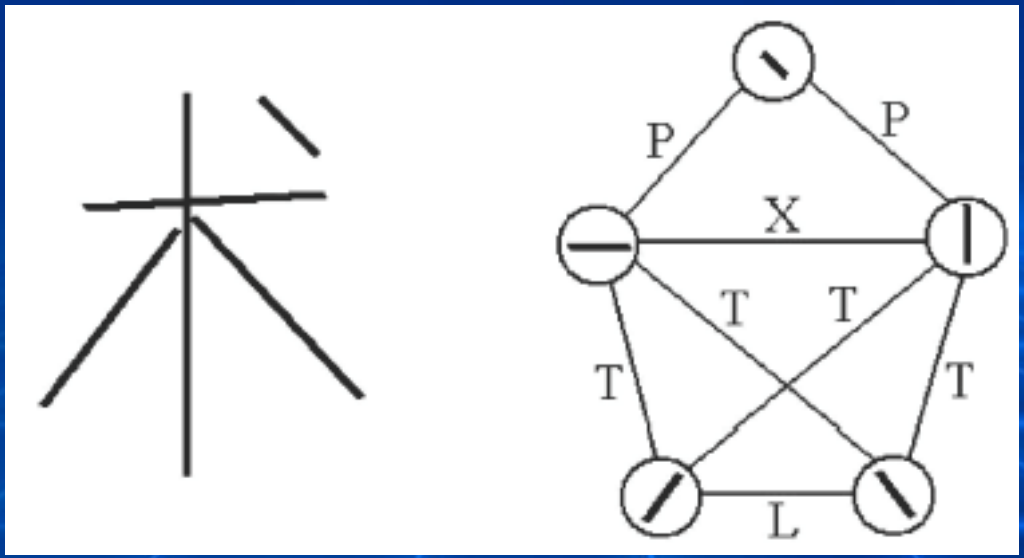
- 笔划编码
  - 笔划可以归类为若干“类别”
  - 每个类别分配一个码字/Index
    - 横-1，竖-2，撇-3，折-4，点-5
  - 每个笔划编码对应一个参考模型/原型或者一组规则（描述该笔划）
- 字符模型
  - 笔划编码的序列，e.g. 木：1235
  - or 以笔划编码为基元的关系结构
- 识别
  - 通过有限自动机等方法在输入模式中进行笔划模型匹配来检测笔划

（3）关系表示

- 关系结构
  - 将字符的构成基元（笔划或线段）及其关系进行建模
- 属性关系图(Attributed Relational Graph, ARG)
  - 节点表示基元
  - 弧（连接）表示基元之间的关系

（3）关系表示

- ARG示例
  - X——交叉（intersection）
  - T——端线T连接（end-to-line adjacency）
  - L——端端连接（end-to-end adjacency）
  - P——位置关系positional relation



（4）层次结构化表示

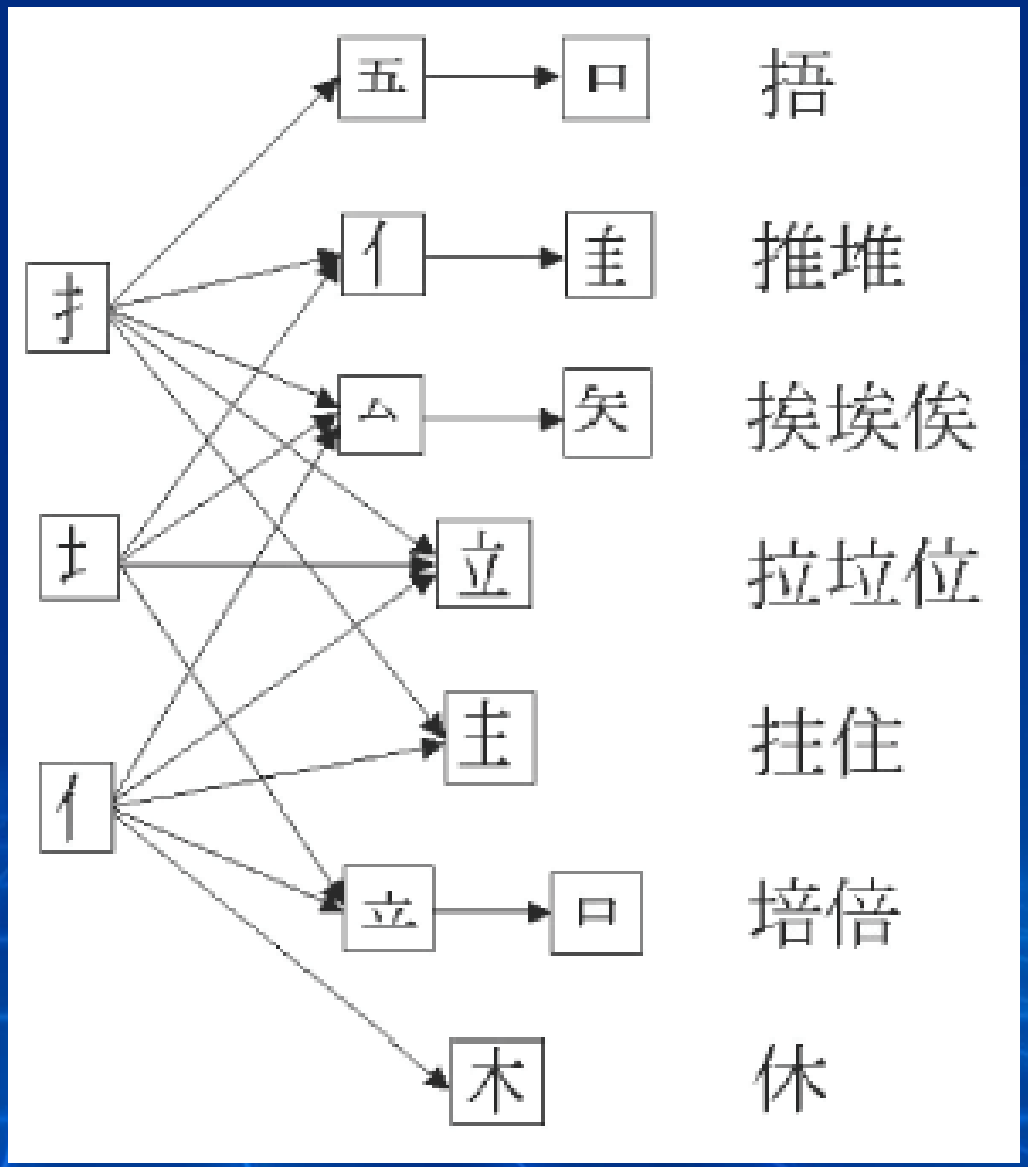
- 汉字天然的“层级结构”
  - 汉字—字根—笔划—子笔划—采样点
- 结构化表示的主要特点
  - 共享字根/笔划模型
  - 通过动态重组字根/笔划来构造汉字
  - 优点
    - 大量节省了模型的存储空间
      - 数百个字根→数千个汉字

（4）层次结构化表示

- 结构化表示的模型数据库组织方法
  - 查找表(lookup table)
  - 树结构
  - 网络结构
- 识别策略
  - 从输入模式中提取字根/笔划，然后遍历整个汉字结构模型数据库
- 不同变种
  - 使用线段、小笔划来代替字根或笔划

（4）层次结构化示例

- 网络层次结构化
  - 网络中每条路径对应一个汉字



2、汉字的结构匹配

- 基本过程
  - 输入模式与候选汉字的结构模型逐一匹配
  - 输出具有最小匹配距离的汉字类别即可
- 两类策略

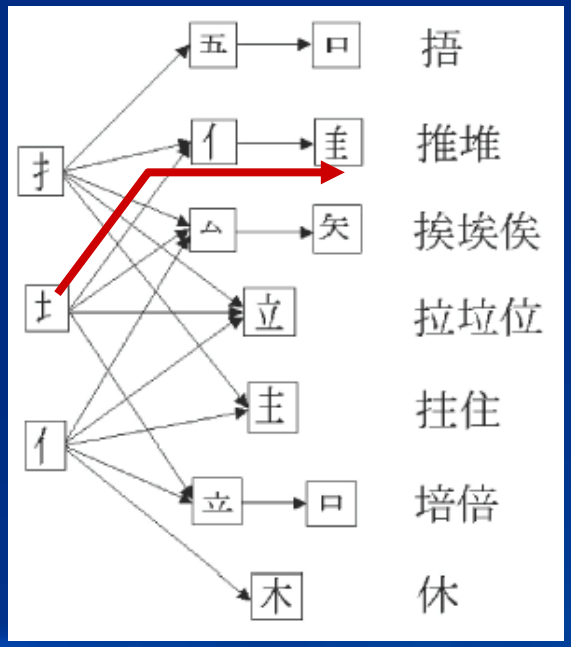
  分级匹配　　形变方法
- 四种方法

  动态规划　　笔划对应

  关系匹配　　基于知识的匹配

（1）结构匹配策略

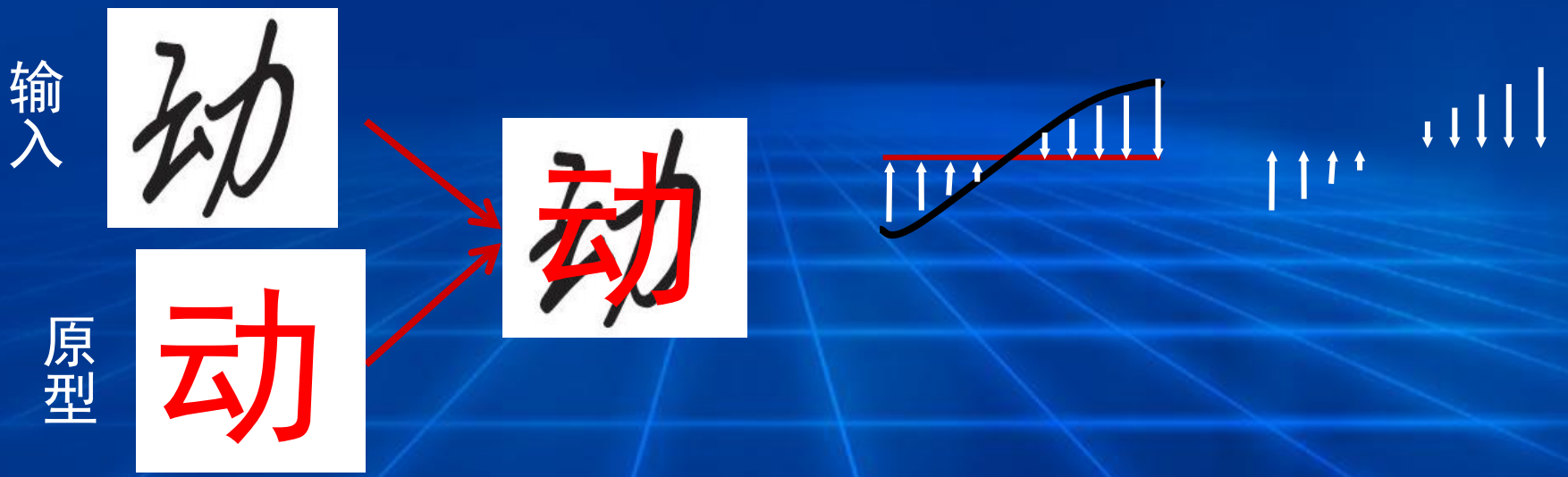
- 分级匹配
  - 优点——可以进一步提高速度
    - 笔划编码/字根模型共享，可使用决策树加速识别过程
    - 字根/笔划一旦检测出来，汉字识别就变成了遍历一棵树的某条路径
  - 缺点
    - 输入模式笔划/字根的检测并非易事
    - 解决：确定性遍历→度量路径的似然性



堆

（1）结构匹配策略

- 形变策略
  - 基于笔划对应关系，计算“输入模式和原型”之间的“形变向量场（DVF）”
    - 汉字原型通过局部仿射变换(Local Affine Transform)来适应输入模式
  - 形变程度决定相似度



（2）结构匹配方法

- 四类方法
  - 动态规划
    - 动态规划针对有序的序列进行，依赖于笔顺
  - 笔划对应
    - 笔划 vs.笔划，不考虑笔划间的关系
  - 关系匹配
    - 考虑笔划间的关系
  - 知识匹配

a、动态规划匹配方法

- 基本思路
  - 通过最小化编辑 (Levinstein)距离来寻求两个符号串(基元序列)中的符号(基元)的对应
    - 基元：采样点，小笔段，笔划，字根
  - 最小化的编辑距离即可作为二个串的距离/匹配度
    - 计算从原串(s)转换到目标串(t)所需要的最少的插入、删除和替换的数目

示例

汉字原型
笔划序列

输入模式
笔划序列

输入模式
笔划序列

输入模式
笔划序列

b. 笔划对应

- 任务
  - 输入模式与字符原型匹配距离计算
- 方法
  - 首先做笔划对应→笔划距离之和
  - 笔划对应也可以采用DP完成

汉字原型笔划序列

输入模式笔划序列

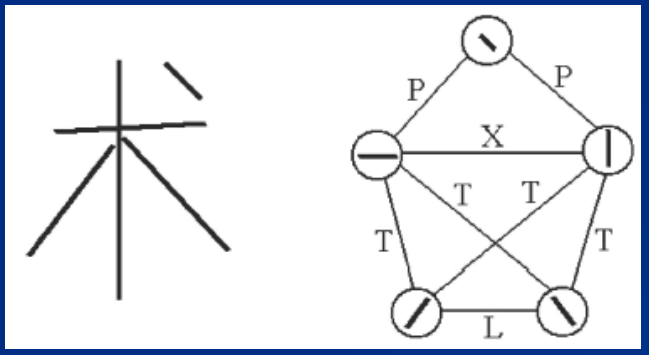
$$D = d1 + d2 + d3 + d4 + d5 + d6 + d7 + d8 + d9$$

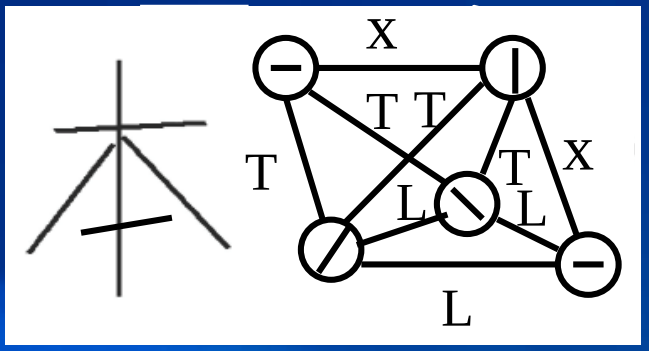
c. 关系匹配

- 任务
  - 关系表示的匹配 → 图匹配问题
- 形式化
  - 在关系（relationship）约束下搜索两个集合中的元素之间的对应关系
- 问题求解
  - 可以被形式化为“一致标定(consistent labeling)”问题，通过人工智能中的启发式搜索(heuristic search)或者松弛标定等方法解决
    - 松弛标定(Relaxation Labeling)计算效率高
    - 而启发式搜索则更灵活，可以很容易的结合不同的知识源和约束条件

C、关系匹配

- 属性关系图ARG的匹配
  - A*搜索算法
  - 松弛标定
- 应用于字符识别或者字根检测
- 优点
  - 笔划顺序无关——DP是笔划相关的
  - 关系约束提高了匹配精度——笔划对应不考虑笔划之间的关系
- 缺点
  - 计算效率低于DP和笔划对应等





d. 基于知识的匹配

- 含义
  - 利用汉字结构和书写方式的先验知识
  - 作为启发信息或者作为约束，用以减少搜索
- 知识
  - 结构知识：上下，左右，内外
  - 预先指定（字根/单字）笔顺、笔划数的允许变化范围
  - 笔划顺序的统计
    - 比如某个笔划之后很少出现另外一个笔划之类
- 优点
  - 知识规则的有效利用可以减少搜索
  - 有利于区分相似的字符，从而提高精度
- 缺点
  - 知识库的建立和组织并不容易而且费时

（一）基于句法的汉字识别

- 优点
  - 理论上是比较恰当的，对字体变化的适应性强，区分相似字能力强
- 缺点
  - 描述复杂，匹配过程复杂度也高
  - 抗干扰能力差，结构基元提取困难，导致推理过程难以进行
    - 实用中文本图象中存在着各种干扰，如倾斜，扭曲，断裂，粘连，纸张上的污点，对比度差等等
  - 纯结构模式识别方法已经逐渐衰落

（二）基于统计的汉字识别方法

- 基本思路
  - 将字符点阵看作一个整体，其所用的特征是从这个整体上经过大量的统计而得到的
- 缺点
  - 细分能力较弱，区分相似字的能力差一些
- 优点
  - 抗干扰性强，尤其适用于有污染的数据
  - 匹配与分类的算法简单，易于实现

1、汉字的统计特征

- 直接图像特征
- 变换特征
- 投影直方图
- 矩特征
- 几何描绘子
- 笔划密度特征
- 外围特征
- 微结构特征
- 特征点特征
- 粗网格特征
- 小笔段特征

（1）直接图像特征

- 实际上并不需要特征提取过程，字符的图象直接作为特征与字典中的模板相比，相似度最高的模板类即为识别结果
- 优点
  - 简单易行，可以并行处理
- 缺点
  - 不同大小、不同字体需要大量模板
  - 对于倾斜、笔划变粗变细均无良好的适应能力

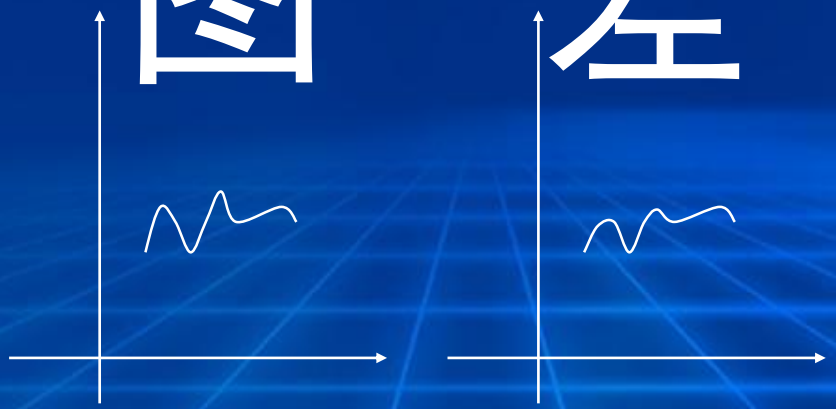
（2）变换特征

- 字符图象进行某种数学变换
  - 二值类变换，如Walsh, Hardama变换
  - 更复杂的变换，如KL, Fourier变换, 余弦变换(DCT)
- 优点
  - 变换后的特征的维数通常会降低，更紧凑，利于分类
- 缺点
  - 多数变换不是旋转不变的，因此对于倾斜变形字符的识别会有较大的偏差
  - 有些最优变换特征的运算复杂度较高，如：K-L变换在最小均方误差意义下是最优的变换，但是运算量大

（3）投影直方图

- 利用字符图象在特定方向的投影作为特征
  - 通常使用水平及垂直方向
- 该方法对倾斜旋转非常敏感，细分能力差

图　差



（4-5）矩特征、几何描绘子

- 矩特征
  - 各种几何矩均在线性变换下保持不变
  - 但往往很难保证线性变换这一前提条件

- 几何描绘子
  - Spline样条曲线近似
    - 在轮廓上找到曲率大的折点，利用Spline曲线来近似相邻折点之间的轮廓线，并用Spline曲线参数作为特征。
  - 傅立叶描绘子
    - 利用傅立叶函数模拟封闭的轮廓线，将傅立叶函数的各个系数作为特征。
    - 对于轮廓线不封闭的字符图象不适用，难用于笔划断裂的字

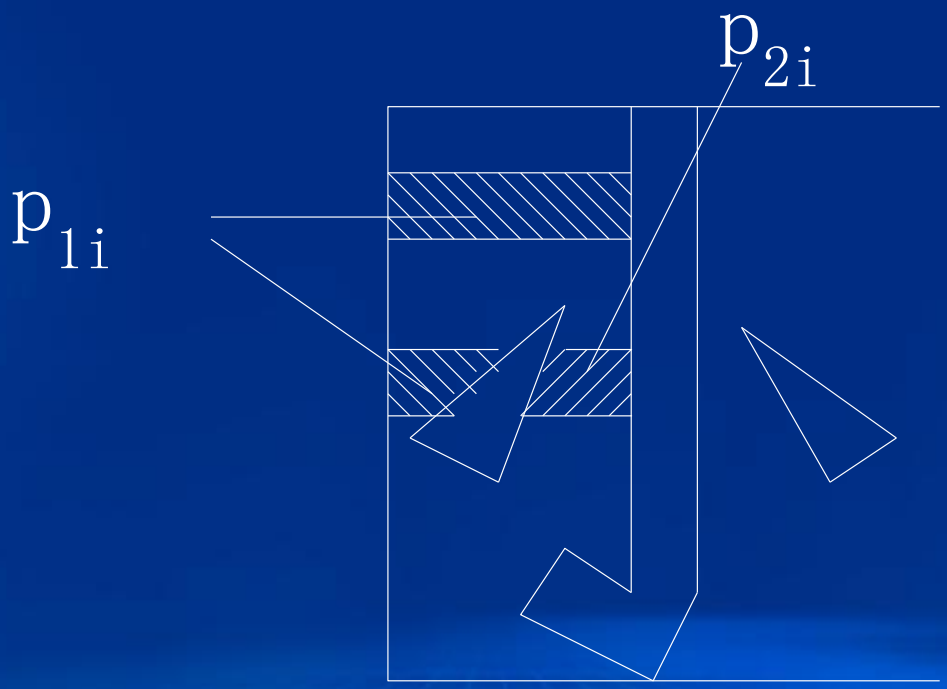
（6）笔划密度特征

- 笔划密度特征提取
  - 从不同方向扫描文字，计算和笔划相交 的次数，形成笔划密度特征。通常取水 平、垂直和两对角线四个扫描方向，每 个方向取若干个区域得到若干特征。
- 优点
  - 这种特征描述了汉字的各部分笔划的疏密程度
  - 在图象质量可以保证的情况下，这种特征相当稳定。
- 缺点
  - 在字符内部笔划粘连时误差较大

（7）外围特征

- 汉字的轮廓包含了丰富的特征
- 提取方式：
  - 从文字四边各向对边扫描，计算最初与文字相碰的非文字部分面积与全部面积之比作为一次粗外围特征
  - 再将第二次与文字 线相碰的非文字部分面积与全部面积之比作为二次粗外围特征，这样形成8n维特征向量
  - 一次粗外围反映了文字的外轮廓，二次粗外围在某种程度上反映了文字的内部结构
- 缺点
  - 不具备细分能力
  - 非常适合于作为粗分类的特征

例子



（8）特征点特征

- 1957年，Solatron Electronics Group公司发布了第一个利用窥视孔(peephole)方法的OCR系统
- 主要思想
  - 利用字符点阵中一些有代表性的黑点(笔划)，白点(背景)作为特征来区分不同的字符，运用到汉字识别中，对其中的黑点又增加了属性的描述，如端点、折点、歧点、交叉点等
    - 端点和折点决定了一个汉字的笔划位置和形状；
    - 歧点和交点决定了不同笔划间的连接关系；
    - 关键背景点弥补了区别相似笔划特征点汉字的不足
- 优点
  - 对于内部笔划粘连的字符的识别的适应性较强，直观性好
  - 匹配难度大，不适合作为粗分类的特征

（9）粗网格特征

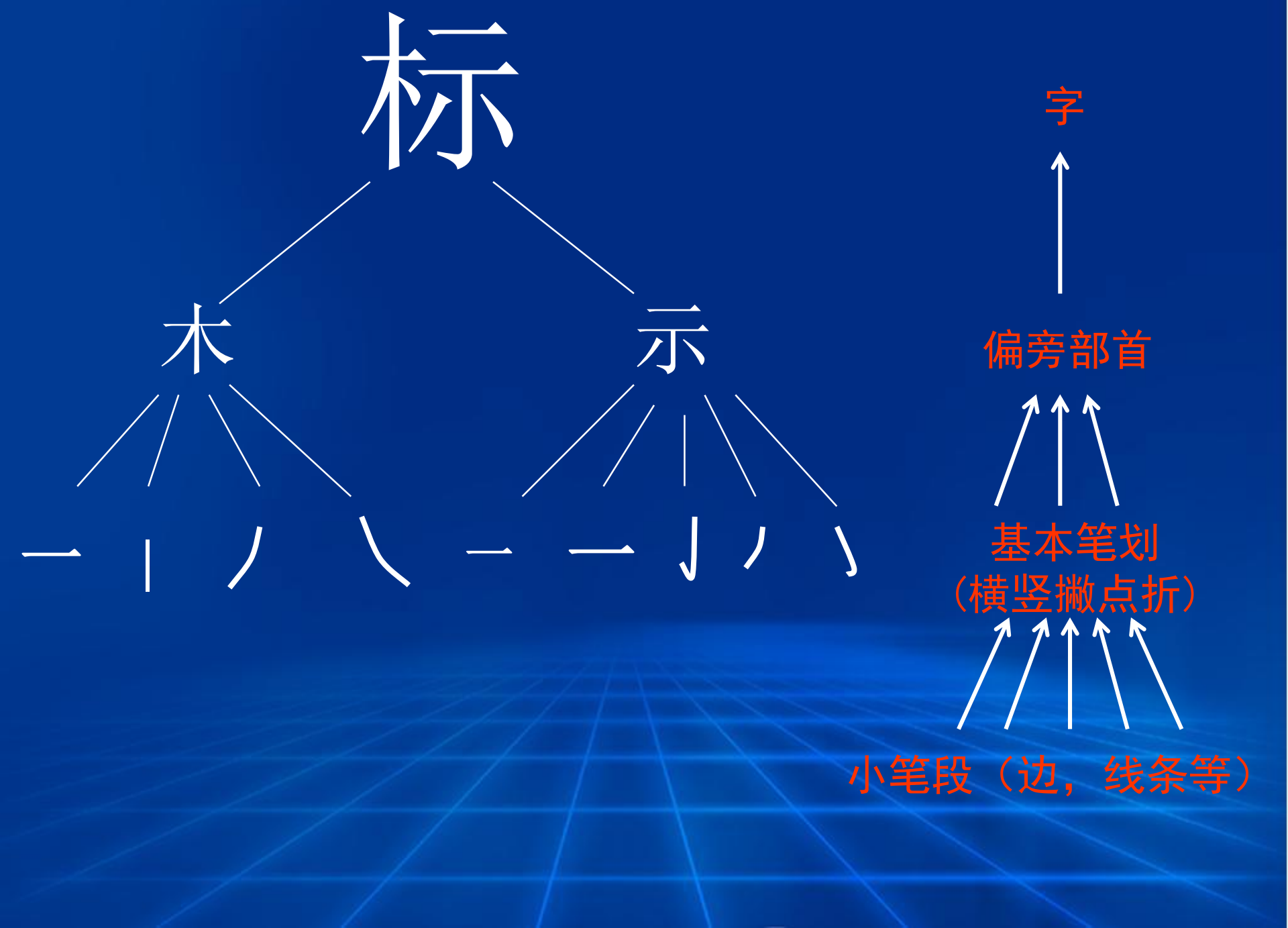
- 把文字分为n*n份，n通常为8
- 取每份的黑点数对整个文字黑点数的比例
- 形成n*n维特征向量

（10）基于微结构特征的方法

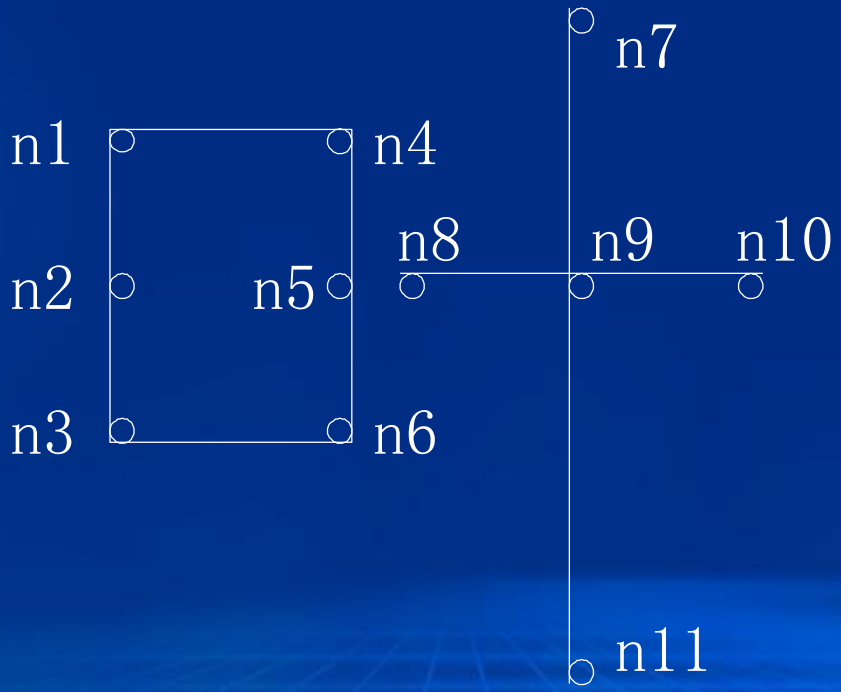
- 思路
  - 汉字是由笔划组成的，而笔划是由一些更小的结构组成的，称为微结构
  - 利用微结构及微结构之间的关系组成的特征对汉字进行识别
  - 微结构是比基本笔划更小的基本单元
- 不足之处
  - 在内部笔划粘连时，微结构的提取会遇到困难

（11）基于小笔段特征层次结构

- 汉字的笔划特征受字体、字号影响较小，这是识别汉字的很好特征，但是它受噪音影响很大难以提取
- 用基于小笔段的层次结构，能较好地解决这一问题
- 若干小笔段首尾相连构成汉字笔划和部件
- 小笔段比实际的笔划易于提取，同时它的抗噪音和字体变化能力较强


标
木
示
字
偏旁部首
基本笔划
（横竖撇点折）
小笔段（边，线条等）

汉字“叶”的小笔段表示



2、统计特征匹配方法

（1）概率匹配

- 简单属性→属性概率模型(容忍基元/关系差异)
- 距离/相似度计算
  - 笔划原型建模为(高斯)概率密度函数
  - 形成笔划类型概率表
- 输入模式 – 字符模型 之间的相似度计算
  - 首先通过属性串匹配或者启发式搜索将输入模式的特征点或线段组织为“笔划”，并给出相似概率
  - 计算字符模型中所需的全部“笔划”的联合概率

2、统计特征匹配方法

（2）统计分类法

- 子空间法
  - 每个汉字一个子空间
- 多相似度量，二次判别函数

- 缺点
  - 计算和存储需求大
- 替代算法
  - 直接计算特征向量之间的欧式距离
  - 每个汉字多个原型，可通过LVQ等聚类算法得到

（二）基于统计的汉字识别方法

- 统计模型
  - 输入模式→特征向量
  - 数据库中包含分类参数，可以通过标准的统计技术进行估计
- 优点
  - 特征向量与笔顺/笔划数无关，可以容忍笔顺/笔画数的较大差异
- 缺点
  - 丢失了笔顺/笔划信息，可能导致精度下降
  - 写完后才能识别，难以实现边写边提示

（三）结构+统计混合方法

- 结构表示的概率化
  - 原理上，任何结构表示模型都可以通过使用概率密度函数(PDFs)取代基元(节点)和关系(连接)的属性来获得其统计-结构表示
- 概率模型，例如：
  - 笔划和关系属性的均值和方差
  - 特征点/笔划属性分布的高斯函数模型
- HMM: Hidden Markov Models
  - 隐马尔科夫模型

HMM

- HMM
  - 有向图：通过概率描述节点及其节点跳转
  - 观测序列（特征点或线段）解码为最可能的状态序列（HMM：有向图）
  - 状态 == 笔划 或 字根 或 字根间连接
- 方法
  - Viterbi译码（一种动态规划算法）

HMM

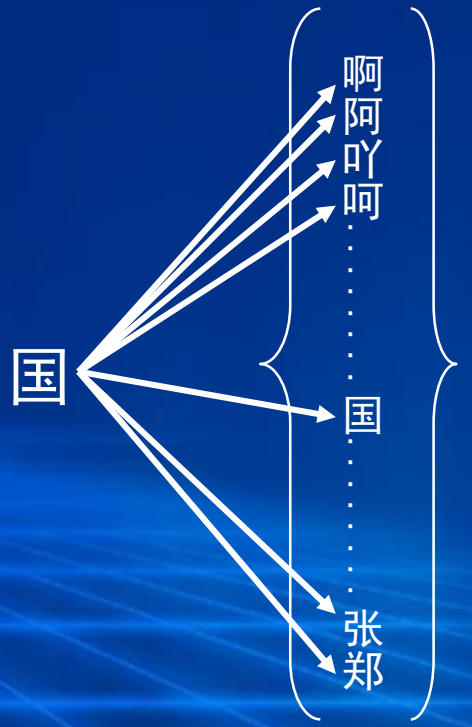
- HMM可以用来
  - 为子笔划、笔划、字根或整个字符，建立他们的点或线段的序列模型
- 基于字符的HMM(每个字符一个HMM)
  - 问题：笔划顺序依赖的模型
  - 解决：多个模型来覆盖不同的笔顺/形状变异
- 基于笔划/子笔划的HMM
  - 可以层级的建立字符模型
  - 笔顺差异可以使用一个差异网络来表示

粗分类与细分类

- 由于汉字数目庞大，在识别中采用在一个线性表中逐一匹配的方法会使识别速度特别慢
- 为了提高识别速度，实际系统中常采用树分类器
  - 层次化的分类结构
    - 树根,树枝,树叶
- 一般把最后一级分类称为细分类，而前面的分类称为粗分类

粗分类与细分类

- 快速分类的基本思路
  - 快速分类器
  - 快速排除大量不易混淆的类别
  - 快速选择出少量可能的候选
- 解决方法
  - 粗分类 → 细分类



由粗到细

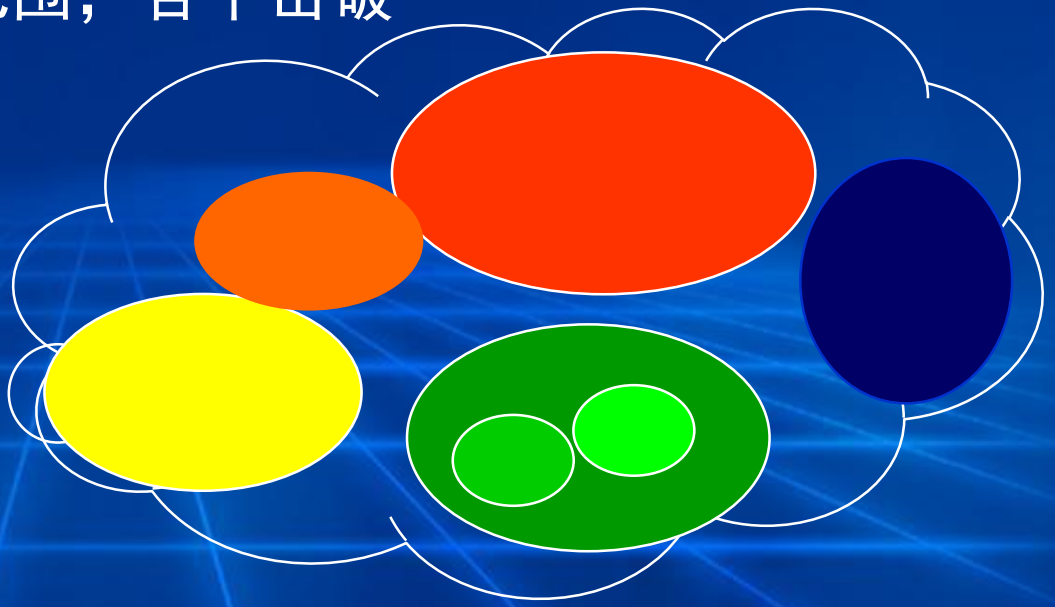
Coarse to Fine

粗分类方法之一

- 分组法
  - 类集合划分法Class set partitioning
  - 所有汉字划分为若干分组
  - 方法论上体现了一个重要理念
    - 军事上：分割包围，各个击破

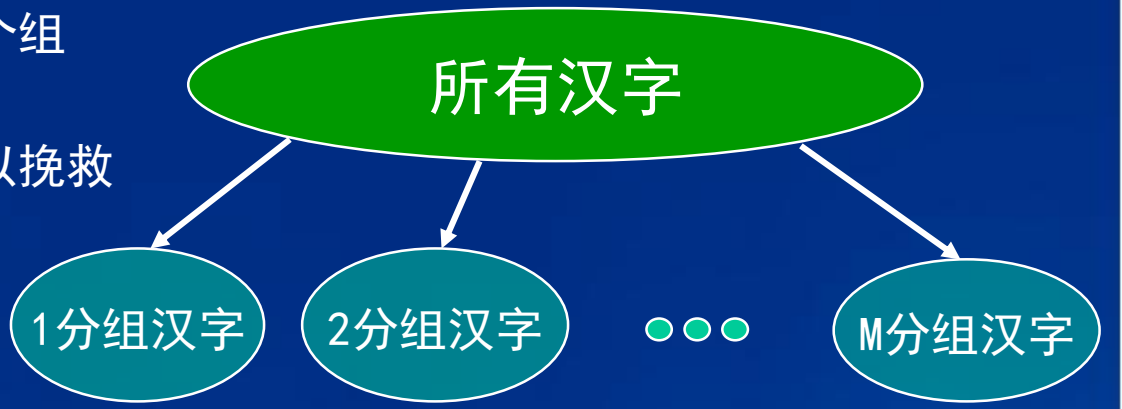
分而治之

Divide and Conquer

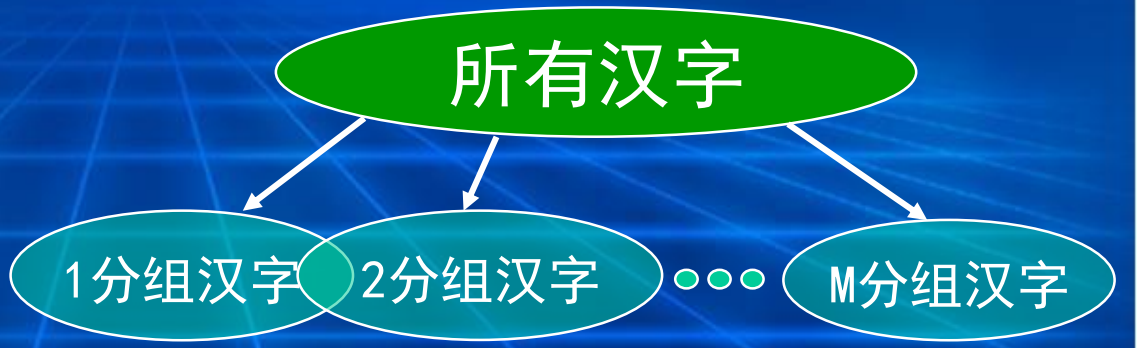


分组模式

- 无重叠
  - 严格划分
    - 每个字只能被划分到一个组
  - 缺点
    - 决策过程一旦出错，难以挽救



- 有重叠
  - 软划分
    - 每个字可被划分到多个组
  - 优点
    - 降低风险



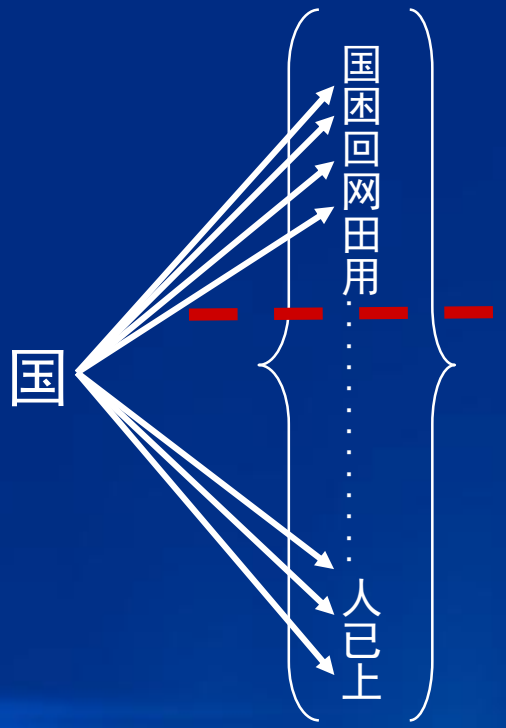
如何分组？

- 问题
  - 在什么时候进行分组？
  - 分类器设计阶段 or 识别阶段？
- 在分类器设计阶段进行汉字分组
- 分组方法
  - 聚类或者根据先验知识进行划分
- 分组依据
  - 总体字符结构
  - 基本笔划子结构
  - 笔划序列
  - 统计特征
  - 神经网络分类器

粗分类方法之二

- 候选汉字动态选择法
  - Dynamic candidate selection
- 基本思路
  - 快速选择出最相似的

    or
  - 快速排除不可能的候选
- 相当于动态分组
  - 在识别阶段分成两组
    - 候选组 + 不可能组



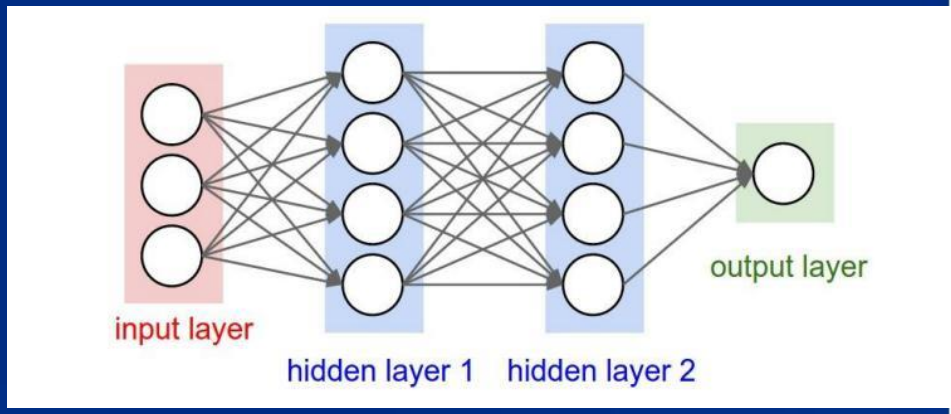
传统模式识别方法比较

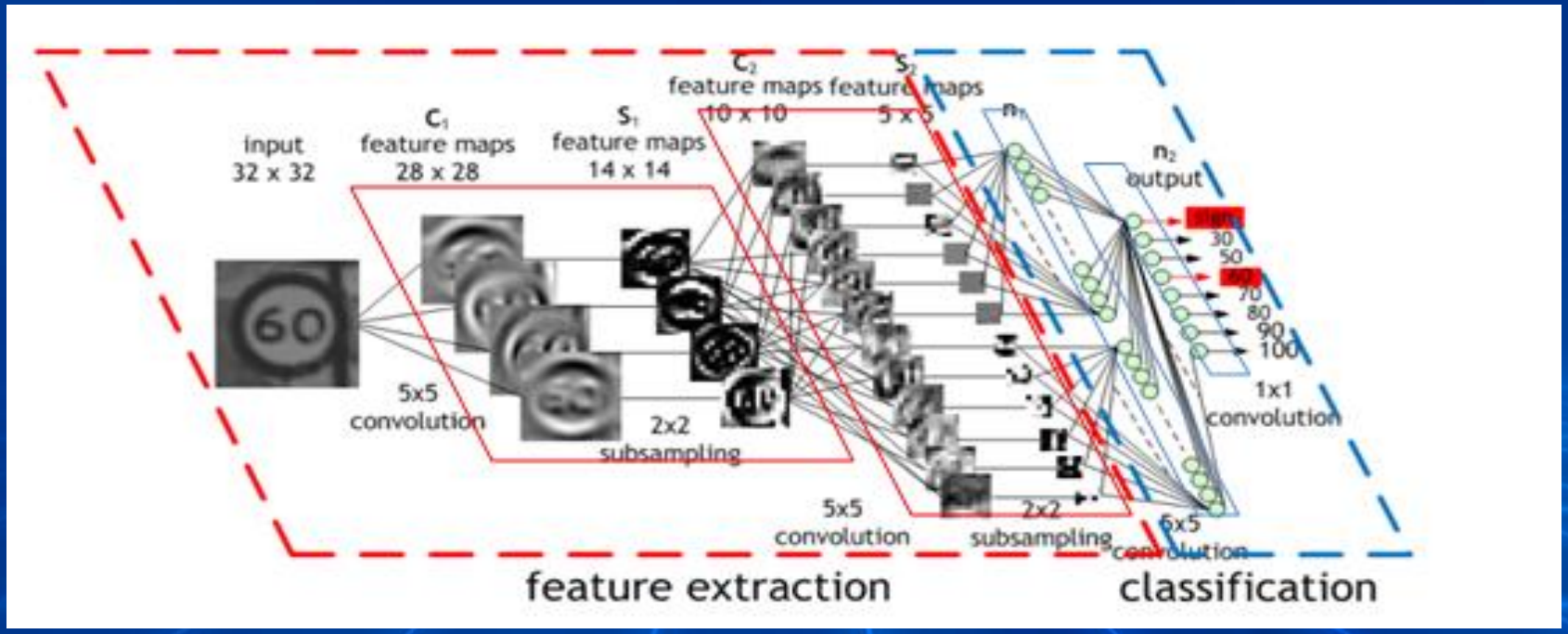
- 结构方法 PK 统计方法
  - 各有胜负，争论不休
  - 统计方法
    - 精度可以很高，但模型数据库很大
      - Velek等人的系统，模型数据库达30MB
  - 结构方法
    - 识别性能可以差不多高，但模型数据小得多
      - Akiyama等人系统，模型数据库仅166KB
  - 混合方法

三、深度学习汉字识别

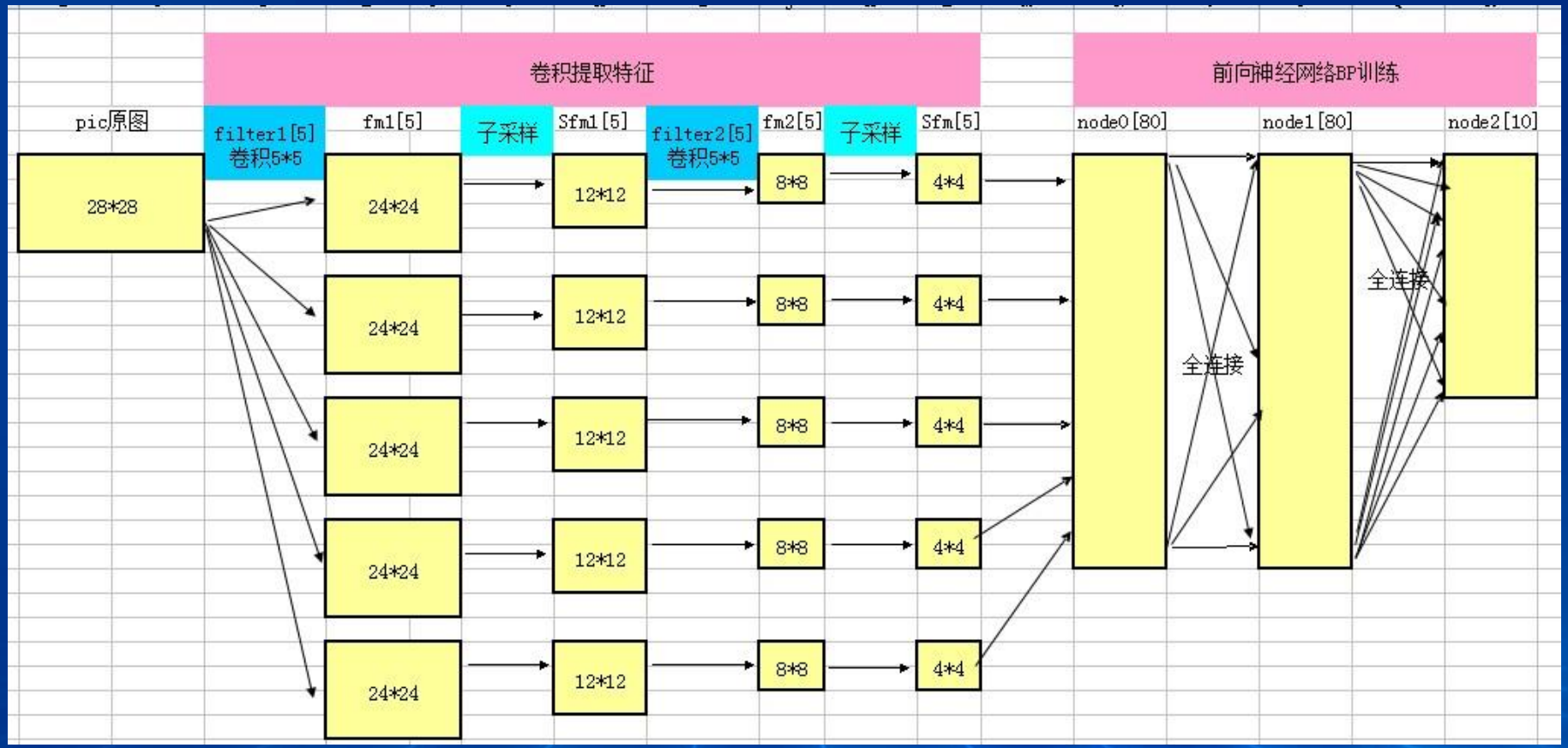
1、卷积神经网络

- 卷积其实是一个图像处理核
- 卷积用于增强图像的某种特征

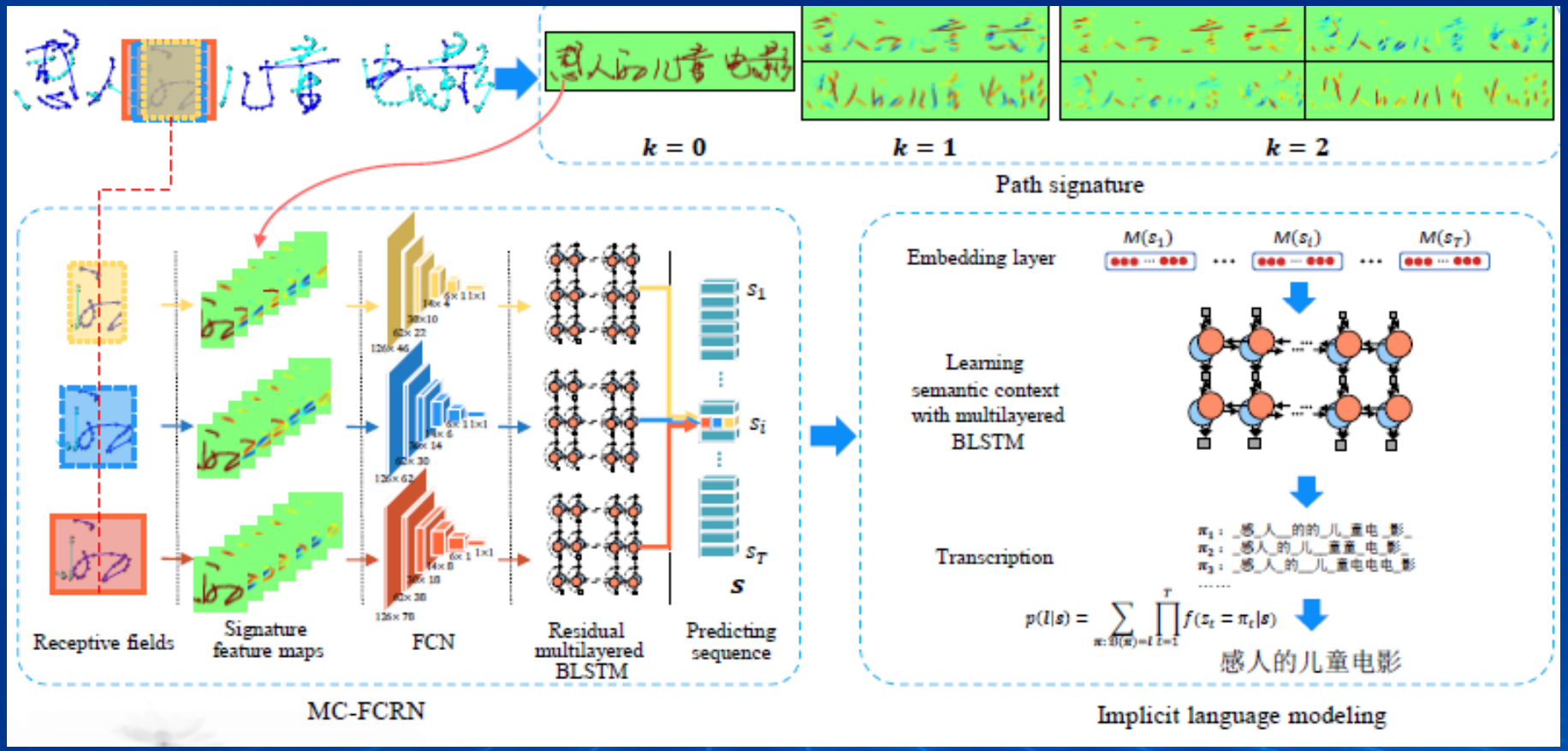




2、基于卷积神经网络的字符识别

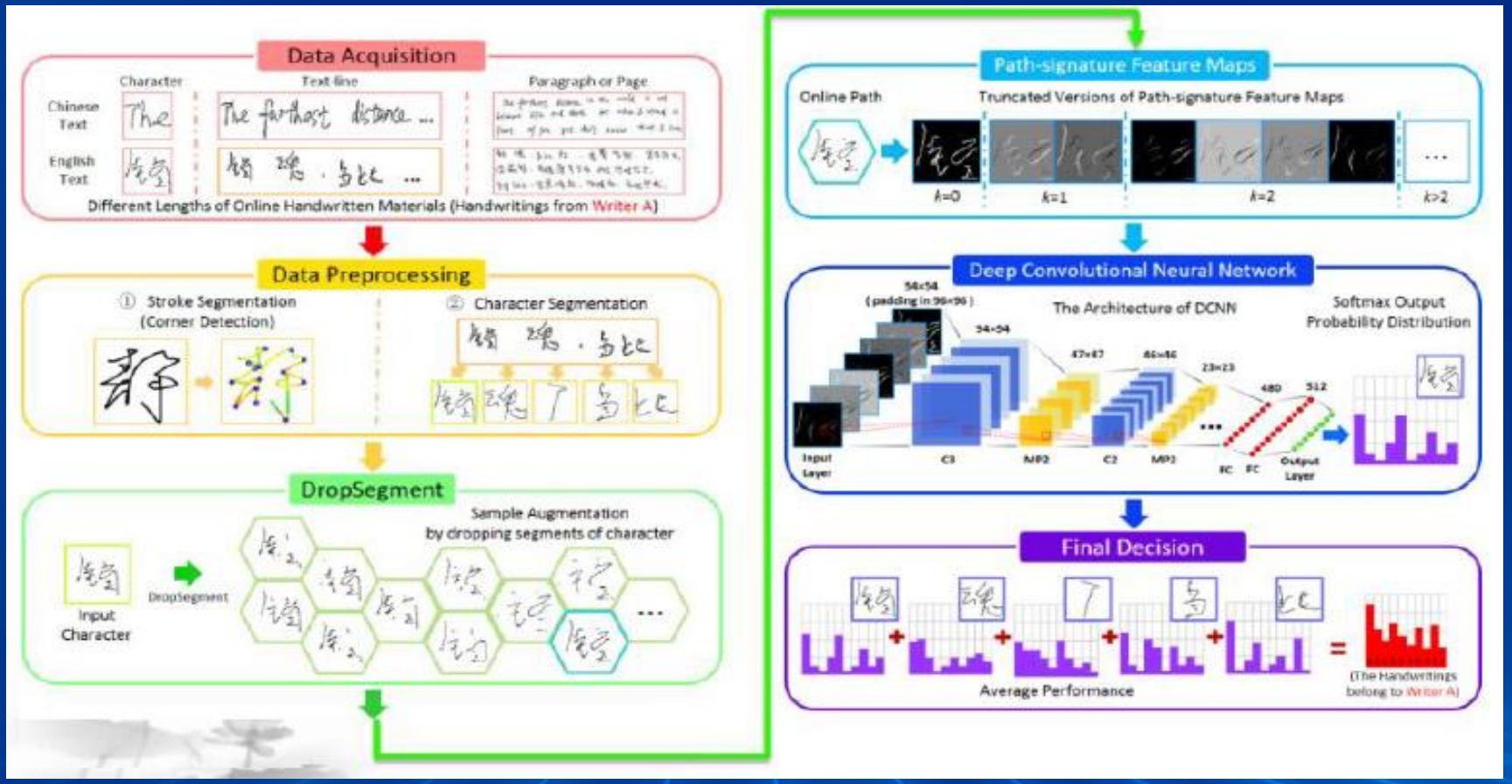


基于卷积回归神经网络的联机手写识别



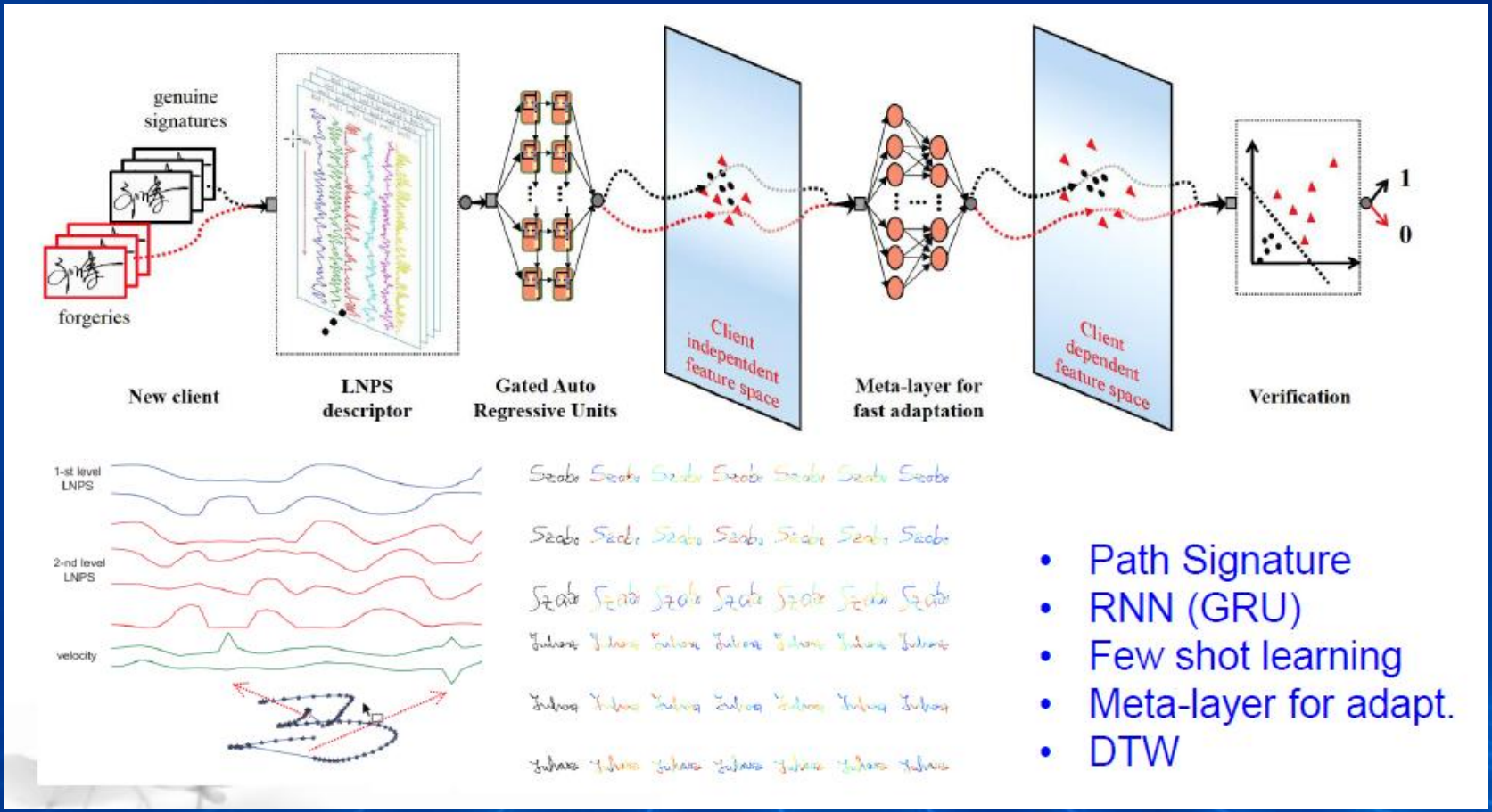
ZC Xie, ZH Shun, LW Jin, et al, Learning Spatial-Semantic Context with Fully Convolutional Recurrent Network for Online Handwritten Chinese Text Recognition, IEEE TPAMI 2018.

端到端的联机手写签名识别



W.X. Yang, L.W. Jin, et al. “DeepWriterID: An End-to-End Online Text-Independent Writer Identification System.”IEEE Intelligent Systems, 2016

基于自适应网络的手写签名识别



SXLai,LWJin. “Recurrent Adaptation Networks for Online Signature Verification.” IEEETIFS, 2019

提　纲

- 笔式交互的简介
- 笔式交互的技术
  - 笔式交互预处理
  - 传统汉字识别技术
  - 深度学习汉字识别
- 笔式交互的评测

一、笔式交互的优化与评测

1、上下文处理

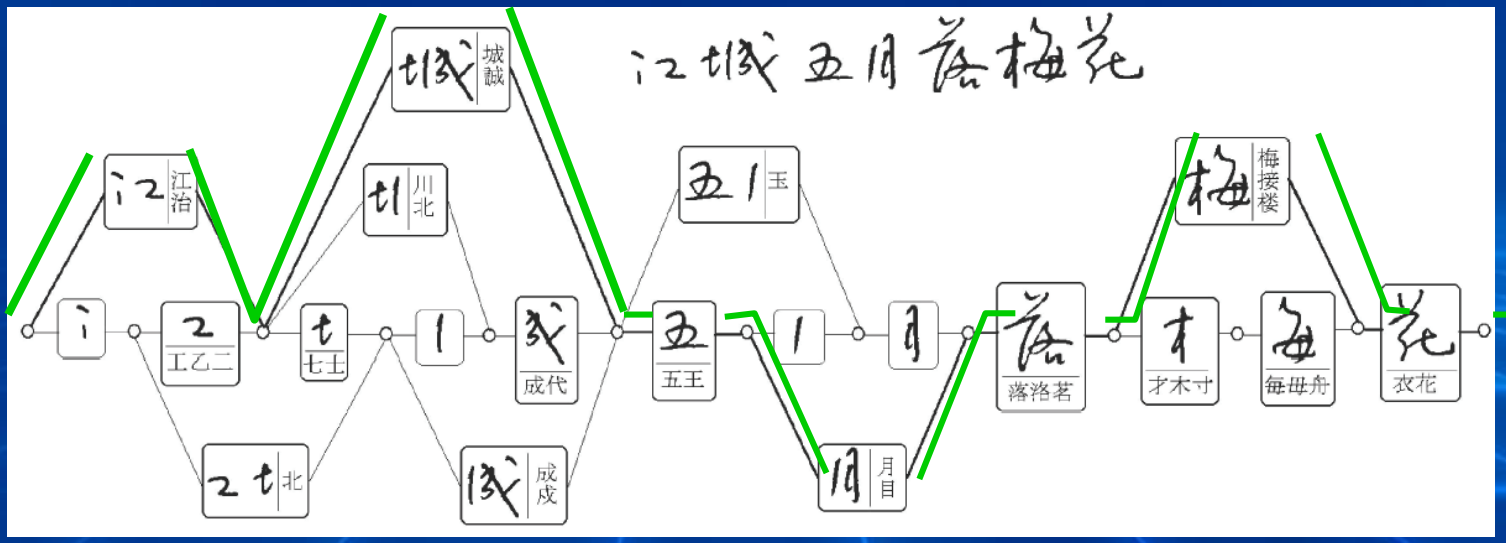
- 必要性和重要性
  - 语言上下文为识别提供了更多的约束
    - 例如：处埋 → 处理
  - 字符的几何特征对分割有帮助
    - 位置，大小，长宽比提供了丰富的信息
  - 特别：分割与识别的互动
    - 难题：连写导致很难得到完全无歧义的分割

1、上下文处理

- 利用语言学知识的上下文处理
  - 后处理措施：在识别完成后进行修正
- 语言模型
  - 基于字符的2-grams
    - e.g.“你好”，“你妙”
  - 基于词的2-grams
    - e.g.“语言模型”，语言模范
- 使用方法
  - 增加额外的候选字/词
  - 作为权重与识别输出信度(score)融合后重新排序
  - 书写人相关的语言模型——语言模型的修改和更新

1、上下文处理

- 分割与识别的联动
- 分割歧义消除
  - 几何特征：位置，间距
  - 识别结果：信度高低
  - 语言学知识：出现概率

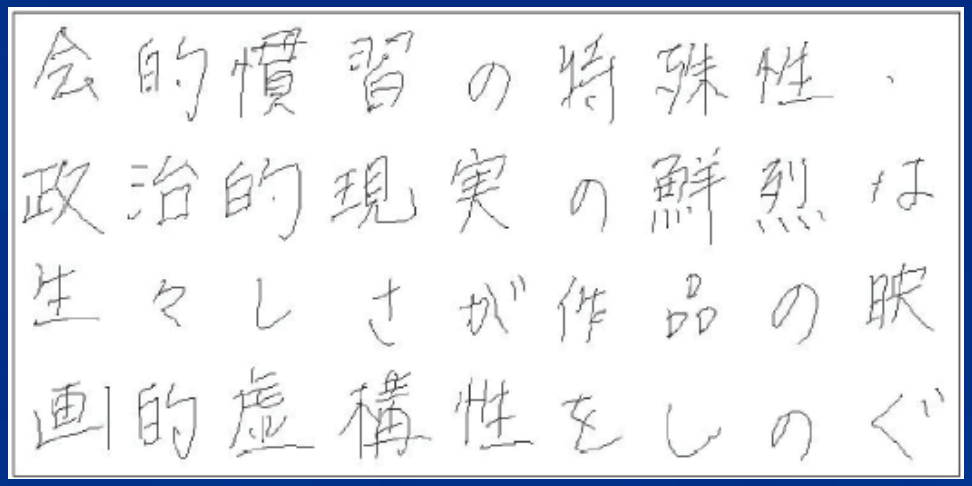


2、笔式交互的评测

- 评测数据库TUAT
  - 书写内容：日文报纸
  - 书写方式：相对随意，无特殊要求
  - Nakayosi：分类器设计
    - 手写板坐标，60*60dots
  - Kuchibue：评测用
    - 显示屏坐标，60*60dots

2、笔式交互的评测

- 实例



Database	#category	#writer	#samples per writer
Kuchibue	3,356	120	11,962
Nakayosi	4,438	163	10,403

2、笔式交互的评测

- 评测技术指标
  - 识别精度
  - 存储需求
  - 识别速度
- 技术指标概况
  - 统计方法
    - 高速，存储需求大
    - 特征向量维数高，尤其对类别子空间法和二次分类器
  - (统计-)结构方法
    - 低速，存储需求小
    - 低速原因：结构匹配是个组合问题

2、笔式交互的评测

- 训练/测试集合大小
  - #Samples per Category [下页表格：#PC]
  - #Samples per Writer [下页表格：#PW]
  - 测试集中待识别汉字个数(类别数)
- 测试集合数据类型
  - 楷体Regular
  - 行书Fluent
  - 草书Cursive

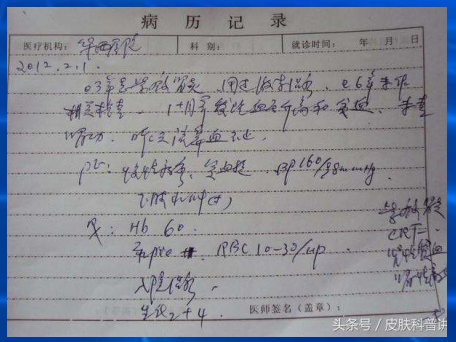
二、笔式交互的发展趋势

发展趋势

- 识别的智能性
  - 融合策略
    - 并行融合
    - 瀑布级联方法
    - 上下文处理
  - 深度学习
    - 自动建模学习
    - 各种GAN、CNN
    - 书写风格适应

发展趋势

- 识别的广泛性
  - 更多种类型
    - 行书、草书
    - 数学公式识别
    - 文艺书法识别
  - 更广泛场景
    - 视频图像场景中文字
    - 金融票据
    - 医学病历
    - 古籍文档
    - 教育笔记



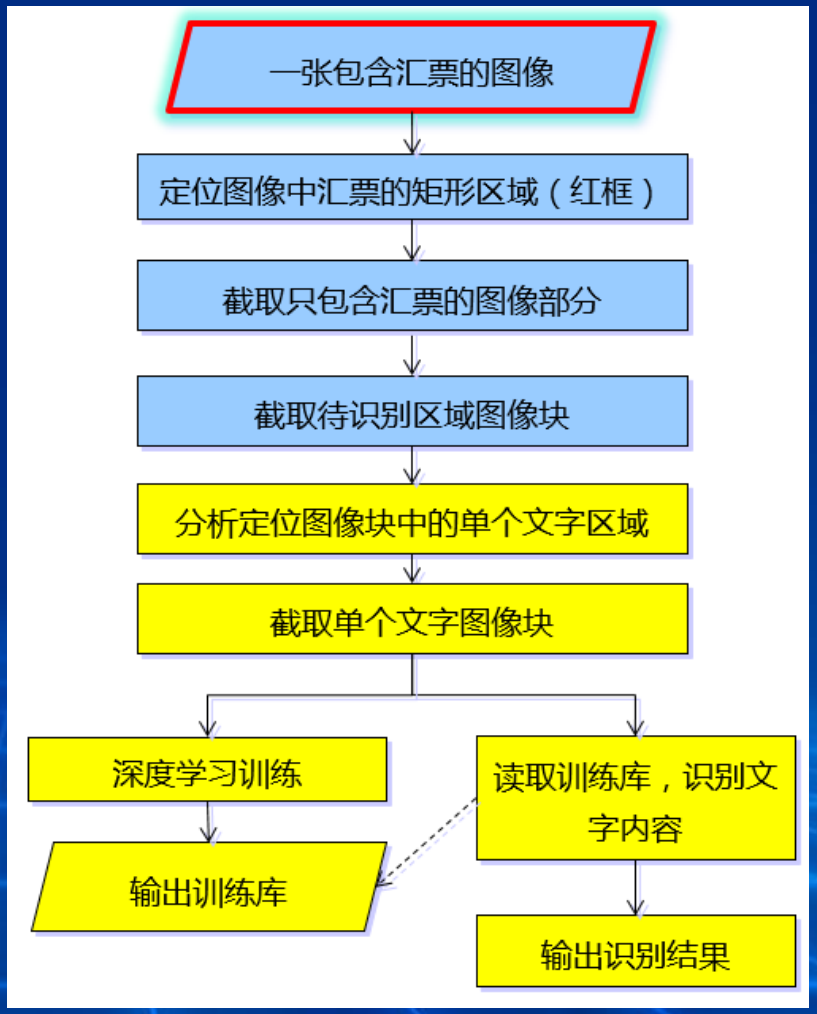
常用的OCR软件

- 英文OCR：
  - OmniPage： 世界最强的英文OCR
- 中文OCR：
  - 清华文通OCR
  - 汉王OCR
  - 中晶尚书OCR
  - 丹青OCR等
- 开源OCR引擎
  - 谷歌开源OCR引擎Tesseract
  - 百度的OCR开放平台AP

案例介绍

- 应用代码举例

票据的识别



- 思考题
  - 联机手写识别与脱机手写识别的区别？
  - 目前笔式交互还有哪些可以改进的方向？

练习1：

实现一个简单的文字识别算法，从黑白图像中识别出文字，就是背景是白色，文字是黑色。

要求：

（1）至少能识别英文26个大小写字母、0-9数字。

（2）有能力的同学可以识别汉字，或从场景中识别字符。

（3）实现语言不限，要求有测试识别率结果分析。

End