「WiseFace——人脸识别门禁系统」

算法描述文档

|  |  |
| --- | --- |
| 队 名： | 四个大聪明 |
| 团队成员： | 郝晓宇 王杰永 王天乐 赵贤贤 |
| 指导教师： | 王荣存 |
| 学 校： | 中国矿业大学 |

**目 录**

[1 概述 1](#_Toc111230440)

[1.1 算法流程 1](#_Toc111230441)

[1.2 主要创新点 1](#_Toc111230442)

[2 人脸位置检测 2](#_Toc111230443)

[2.1 MTCNN 2](#_Toc111230444)

[2.2 RetinaFace 4](#_Toc111230445)

[2.3 SCRFD 5](#_Toc111230446)

[2.4 实验结果 5](#_Toc111230447)

[3 人脸对齐 5](#_Toc111230448)

[4 人脸特征提取 6](#_Toc111230449)

[4.1 算法原理 6](#_Toc111230450)

[4.2 训练集测试 7](#_Toc111230451)

[5 人脸比对 7](#_Toc111230452)

[5.1树 8](#_Toc111230453)

[5.2-++聚类 9](#_Toc111230454)

[6 用户密码保护 10](#_Toc111230455)

[7参考文献 11](#_Toc111230456)

* 1. 概述
     1. 算法流程

人脸识别的流程主要分为四个部分[1]：首先，通过人脸位置检测模型对输入的对象进行处理，以得到图像中的人脸位置信息；其次，通过人脸对齐模型对定位到的人脸图像进行对齐；然后，将对齐后的人脸图像输入人脸特征提取网络，以获得表征人脸信息的、固定维度的向量，即人脸特征向量；最后，与人脸数据库中存储的人脸特征向量进行逐一比对，以获得当前输入图像的人脸信息。人脸识别的一般步骤如图1所示。

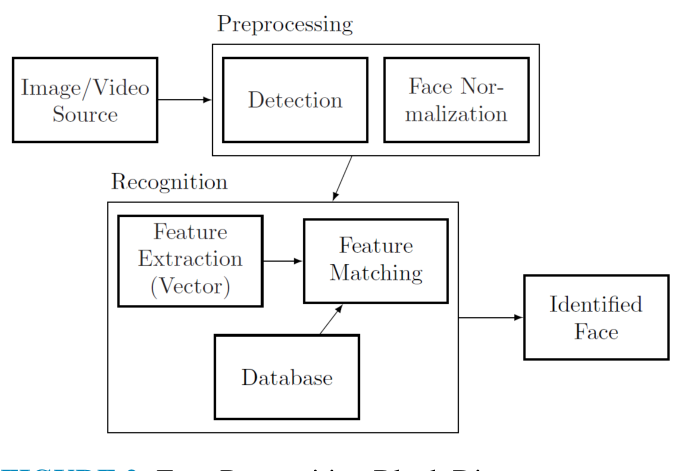


图1 人脸识别流程

* + 1. 主要创新点

对于人脸位置与关键点检测，我们团队在SylixOS系统中分别部署了三种模型，并进行了实验对比分析。综合考虑了SylixOS系统的特点以及三种模型的检测精度、检测速度，最终挑选了精度最高、检测速度最快的以mobilenet作为backbone的RetinaFace模型用于WiseFace的人脸检测。

对于人脸特征提取模型，我们同样选择以mobilenet为backbone的mobilefacenet网络。最终在四个开源数据集上的精确度均超过96%，实际精度均满足并超过主办方的精度要求。

在人脸数据库的比对阶段，考虑到系统的可扩展性，我们不再采用传统的、效率较低的线性比对策略。在WiseFace中，我们建立了以kd-tree作为存储结构的数据组织形式，使得在小规模人脸数据库中的查询时间复杂度从降低至；实现了kmeans++聚类算法，最终在较大规模的人脸数据库中的查询速度相比于线性查找提升10倍左右。

在整个系统的架构上，我们采用了C/S架构——服务器负责重要数据的备份、历史数据的条件查询以及管理员权限的获取验证；客户端主要负责人脸识别。服务器与客户端基于TCP协议通信。

由于人脸的录入、删除等众多操作不应该对普通用户开放，因此我们的系统做了权限管理。在客户端，输入管理员账户和密码以获得管理员权限，从而拥有更大的操作权限。特别的，密码我们采用MD5算法进行加密，保护用户的安全隐私。

最终，我们的WiseFace创新点如下：

1. 支持口罩识别、面部部分遮挡识别。
2. 全部模型文件大小不超过4.8M，检测速度快，准确度均达到甚至超过主办方的要求。
3. 使用聚类算法，大大地提升了人脸数据库的比对效率。
4. 以SQLite3作为用户人脸特征存储的数据库，保证了用户信息安全性。(在软件设计文档中体现)
5. 以多线程的方式优化系统运行速度。(在软件设计文档中体现)
6. 使用MD5算法对管理员密码加密，保证用户的安全隐私。
7. 系统采用C/S架构，实现多种必要的拓展功能，以支持系统部署到真实应用场景中。(在软件设计文档中体现)
   1. 人脸位置检测

人脸位置检测是人脸识别的第一步，对于输入图像，通过人脸检测模型，得到输入图像中的人脸边界框和特征点。我们团队在SylixOS系统中分别部署了三种用于人脸检测的模型——MTCNN、RetinaFace和Scrfd。下面介绍三种模型的算法原理及模型结构，并展示三种模型的训练过程。

* + 1. MTCNN

MTCNN[2]是基于cascade级联框架的，由三个子模型构成：Proposal Network(P-Net)、Refine Network(R-Net)和Output Network(O-Net)。P-Net用于提取面部候选区域； R-Net用于对P-Net产生的可能包含人脸的边界框过滤，回归得到确定包含人脸的边界框； O-Net用于人脸检测以及关键点提取。MTCNN模型的框架如图2所示。

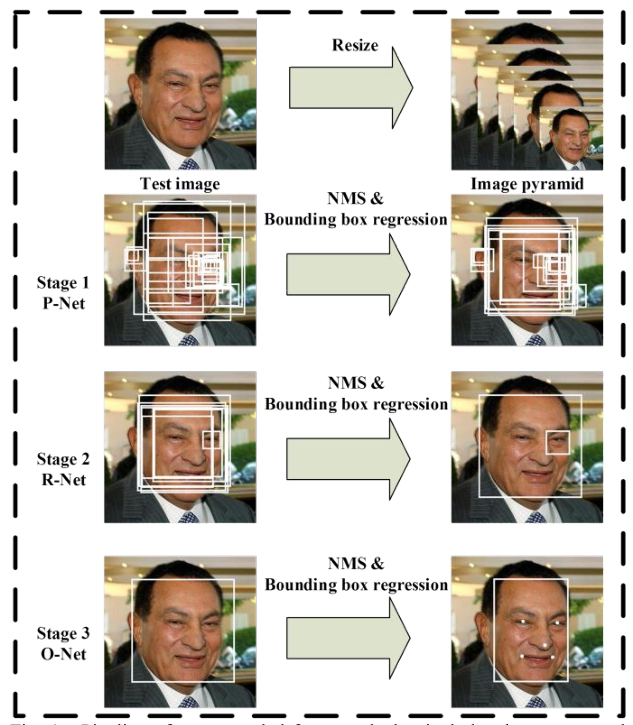


图2 MTCNN baseline

P-Net是一个全卷积神经网络，由卷积层与最大池化下采样层组成，可以接受任意尺寸的输入图像。将输入图像重新缩放为不同尺寸的图像金字塔，连同预先产生的若干anchor，输入到P-Net中，产生可能包含人脸的候选区域，并回归出人脸边界框。最后通过NMS[3](非极大值抑制)去除重叠区域较大的候选区域。图3展示了P-Net的网络结构。

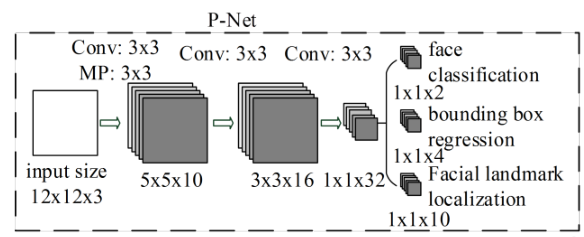


图3 P-Net网络结构

R-Net则是普通的卷积神经网络。将P-Net的输出经过双线性插值得到24×24的输入图像，输入R-Net网络中。R-Net同样通过边界框回归来对P-Net网络的结构进行修正，通过NMS滤去假正例区域。R-Net的网络结构与P-Net相比，在网络最后拉平并连接一个全连接层，因此对假正例的抑制有更好的效果。R-Net网络结构图4所示。

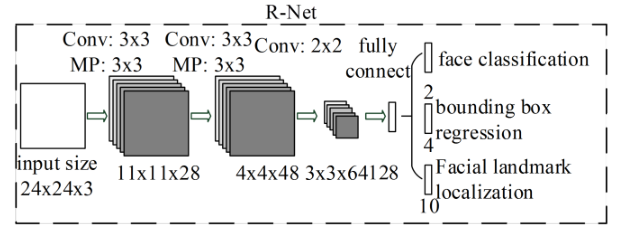


图4 R-Net网络结构

与R-Net相比，O-Net多加入了一层卷积以提取更深、更精细的特征图。将R-Net的输出通过双线性插值重新缩放至48×48，作为O-Net网络的输入；该层对人脸区域进行了更多的监督，同时还会输出人脸的5个特征点。O-Net的网络结构如图5所示。

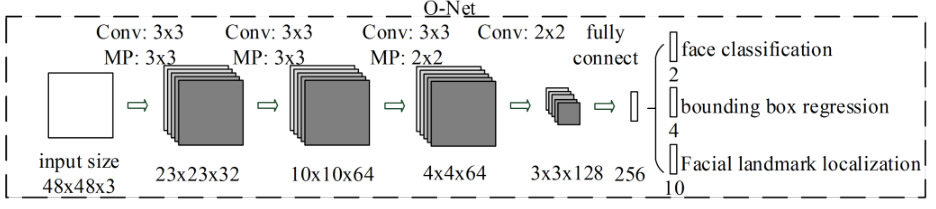


图5 O-Net网络结构

MTCNN中的三个网络的损失函数均是多任务损失函数。其中，对于人脸分类采用交叉熵损失函数；对于边界框预测与关键点检测，采用损失。

损失中的超参数取值如表1所示。

表1 超参数的取值

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| P-Net | 1 | 0.5 | 0.5 |
| R-Net | 1 | 0.5 | 0.5 |
| O-Net | 1 | 0.5 | 1 |

* + 1. RetinaFace

RetinaFace[4]是一个强大的one-stage人脸检测器，通过联合外监督以及自监督的多任务学习，对各种尺寸的人脸进行像素级别的定位与检测。RetinaFace的网络结构如6所示。

将输入图像送入主干网络，获得多张特征图，经过特征金字塔加强特征图的语义信息。最后，每一张特征图通过预测头获得结果。

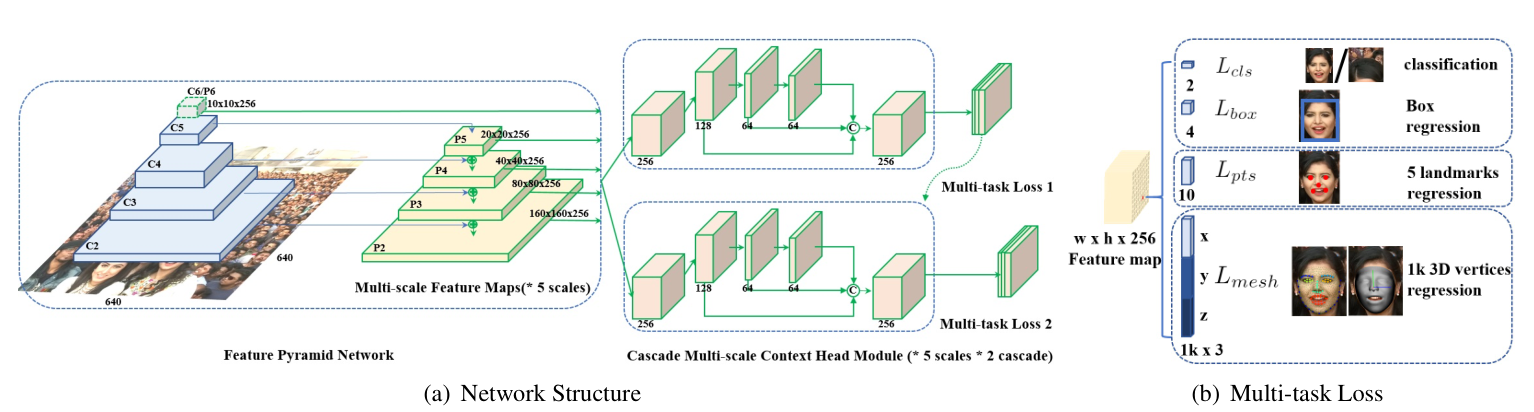


图6 RetinaFace网络结构

RetinaFace的损失函数同样是多任务损失。对于人脸分类任务，采用softmax损失；而对于预测边界框、关键点定位以及3D人脸重建，采用Smooth-L1损失。各个损失之间使用不同的权重保持平衡。

* + 1. SCRFD

SCRFD[5]使用两种方法对网络加以优化：

1. 样本再分配(Sample Redistribution, SR)，基于基准数据集的统计，在最需要的阶段增加训练样本。
2. 计算再分配(computing Redistribution, CR)，它基于一种精心定义的搜索方法，在模型的脊柱、颈部和头部之间重新分配计算。

经实验表明，SCRFD可以在提高准确率的同时，能有效地降低网络计算量。

* + 1. 实验结果

我们将训练好的三种pytorch模型转换成ncnn模型，部署在SylixOS操作系统中。对于同一张分辨率为640×180的包含一张人脸的图像，三种模型用时如表2所示。

表2 三种人脸检测模型在SylixOS上的运行速度对比

|  |  |
| --- | --- |
| **模型** | **时间()** |
| MTCNN | 2500 |
| RetinaFace | 600 |
| SCRFD-0.5GF | 1000 |

观察表2可以看出，以mobileface为backbone的RetinaFace模型检测速度最快，因此我们选择RetinaFace作为WiseFace的人脸检测与关键点定位模型。

* 1. 人脸对齐

人脸对齐是人脸识别系统中必须完成的一步。通过关键点检测进而进行人脸对齐，可以使要是别的人脸进行空间归一化（这一句话不通顺，请修改）——使后续特征提取模型可以提取到与位置无关、只与人脸纹理等相关的特征，从而提升识别的准确率。

人脸对齐是通过仿射变换实现的。仿射变换是一种二维坐标到二维坐标之间的线性变换，允许图像任意倾斜、在两个方向上任意伸缩，也就是说，仿射变换允许图像平移、缩放、剪切、旋转。同时平移和旋转的仿射变换矩阵如下：

|  |  |
| --- | --- |
|  | () |

通过人脸五个关键点以及目标点坐标，构建关于上述仿射变换矩阵参数的线性方程，通过最小二乘法[6]估计参数，可以计算出仿射变换矩阵。

对于原图像中的每一个像素点，利用仿射变换矩阵计算该像素点新的坐标位置，从而完成对原图像中人脸的对齐变换。

* 1. 人脸特征提取
     1. 算法原理

通过人脸位置检测、关键点检测以及人脸对齐，我们得到了对齐后的标准大小的人脸图像。通过人脸特征提取网络，提取表征人脸的特征，进而完成人脸识别。

对人脸特征提取网络有一个基本的要求——同一人的人脸特征向量距离近，不同人的人脸特征向量距离远。最终，选择了MobileFaceNet[7]网络，其网络结构如图7所示。

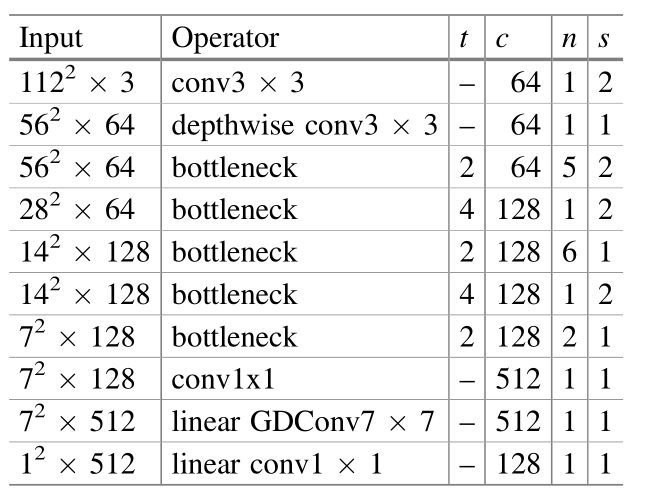


图7 MobileFaceNet网络结构

主要的构建模块采用了MobileNetV2中提出的residual bottlenecks，但与其不同的是，采用了更小的扩张因子，同时使用了在人脸识别中表现较优的Prelu非线性激活函数。最后，采用Arcface损失函数训练：

Arcface通过对特征向量和权重归一化，对加入了更直接影响角度的角度间隔，从而产生更好的性能表现。

* + 1. 训练集测试

在四个开源的数据集上对MobileFaceNet进行了测试，识别准确率均超过96%，均超过主办方对人脸识别准确度的要求。其在四个数据集上的具体性能如表3所示。

表3 MobileFaceNet在开源数据上的测试

|  |  |  |
| --- | --- | --- |
| **数据集** | **出题方建议准确率** | **模型实际准确率** |
| LFW | ≥99% | 99.52% |
| CFP-FP | ≥96% | 97.29% |
| CFP-FF |  | 99.71% |
| AgeDB-30 | ≥92% | 96.40% |

* 1. 人脸比对

在特征提取后，我们得到了表征人脸信息的特征向量。由于不同人脸的特征向量距离远、同一人脸的特征向量距离近，因此我们可以通过特征向量完成人脸识别。

人脸比对的算法框架如图8所示。



图8 人脸比对流程

在对数据库中人脸向量和要识别的人脸向量进行比对时，传统的方法是直接进行线性扫描，此时的时间复杂度是。若数据库中存储了海量的人脸特征向量，传统的线性扫描方法的时间开销较高。尤其是，在实时的人脸检测场景下，该方法是不可行的。为此，我们进一步优化了人脸数据库在内存中的存储结构，使之从线性存储改为树形存储，将时间复杂度从。

* + 1. 树

树[8]是一种对维空间中的实例点进行存储以便对其进行快速检索的树形数据结构。树的构造相当于不断地用垂直于坐标轴的超平面将维空间划分，构成一系列的维超矩形区域。树构造过程对应的流程图如图9所示。

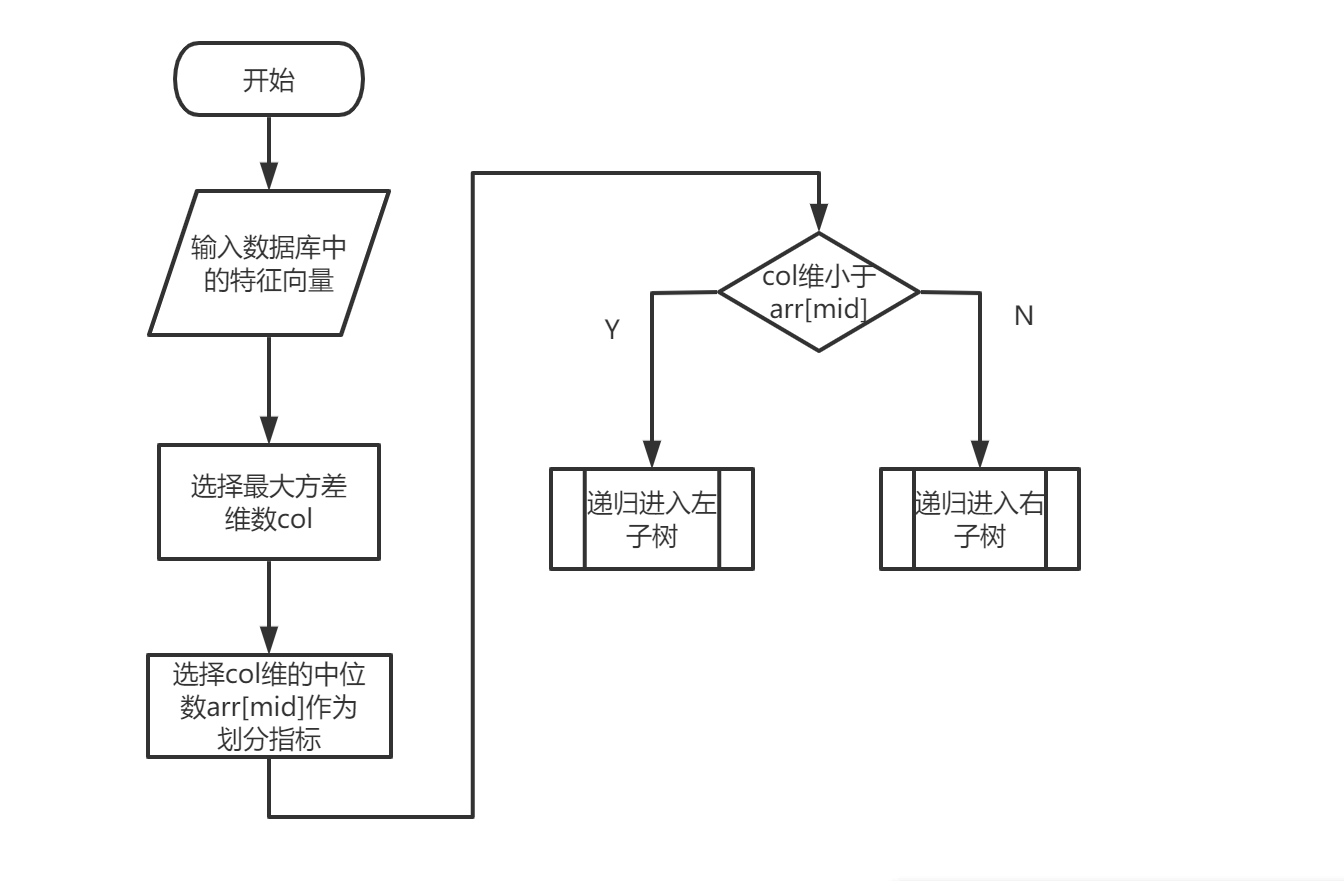


图9 -构造流程图

在实例点随机分布的情况下，树能够实现平均计算复杂度为的搜索。在人脸识别系统中，我们将树用于最近邻搜索，得到人脸识别结果。树的最近邻搜索算法如下：

|  |
| --- |
| 算法：树的最近邻搜索 |
| 输入：已构造的树；目标点； |
| 输出：的最近邻. |
| 1)在树中找出包含目标点的叶结点；从根节点出发，递归地向下访问树。若目标点当前维的坐标小于切分点的坐标，则移动到左子结点，否则移动到右子结点，直到子结点为叶结点为止. |
| 2)以此叶结点为“当前最近点”. |
| 3)递归地向上回退，在每个结点进行以下操作：  (a)如果该结点保存的实例点比当前最近点距离目标点更近，则以该实例点为“当前最近点”.  (b)当前最近点一定存在于该结点一个子结点对应的区域.检查该子结点的父结点的另一子结点对应的区域是否有更近的点.具体地，检查另一子结点对应的区域是否与以目标点为球心，以目标点与“当前最近点”间的距离为半径的超球体相交. 若相交，可能在另一个子结点对应的区域内存在距目标点更近的点，移动到另一个子结点. 接着，递归地进行最近邻搜索；若不相交，向上回退. |
| 4)当回退到根节点时，搜索结束. 最后的“当前最近点”即为所求点. |

经过实际测试发现，在数据库存储张人脸时，线性扫描要优于树搜索。主要原因是，暴力计算的复杂度不受数据结构的影响，而基于树结构的算法对于稀疏数据来说有较大的提升，对于类型的数据则性能较差，人脸向量数据恰好是类型的，即在整个参数空间里几乎没有的存在。在数据库中存储的人脸数量较少时，基于树的最近邻搜索依旧拥有较好的表现。为此，在程序之中设置了的变量，以对是否使用KD树进行搜索的简单控制，实现人脸识别系统在不同应用场景下搜索方式的灵活切换。

* + 1. -++聚类

对于大规模向量查询，我们采用聚类算法降低查询复杂度。考虑到初始聚类中心对最终结果有较大影响，我们选择了比-算法具有更好表现的-++算法。

其初始化聚类中心的算法如下：

|  |
| --- |
| 算法：-++算法个聚类中心的初始化 |
| 1)从样本中随机选择个样本作为初始聚类中心； |
| 2)对于任意一个非聚类中心x，计算与现有最近聚类中心聚类； |
| 3)基于距离计算概率，来选择下一个聚类中心，选择距离当前聚类中心最远的点作为聚类中心； |
| 4)重复步骤2与3，直到选择出来个聚类中心为止. |

常规-聚类流程图如下：

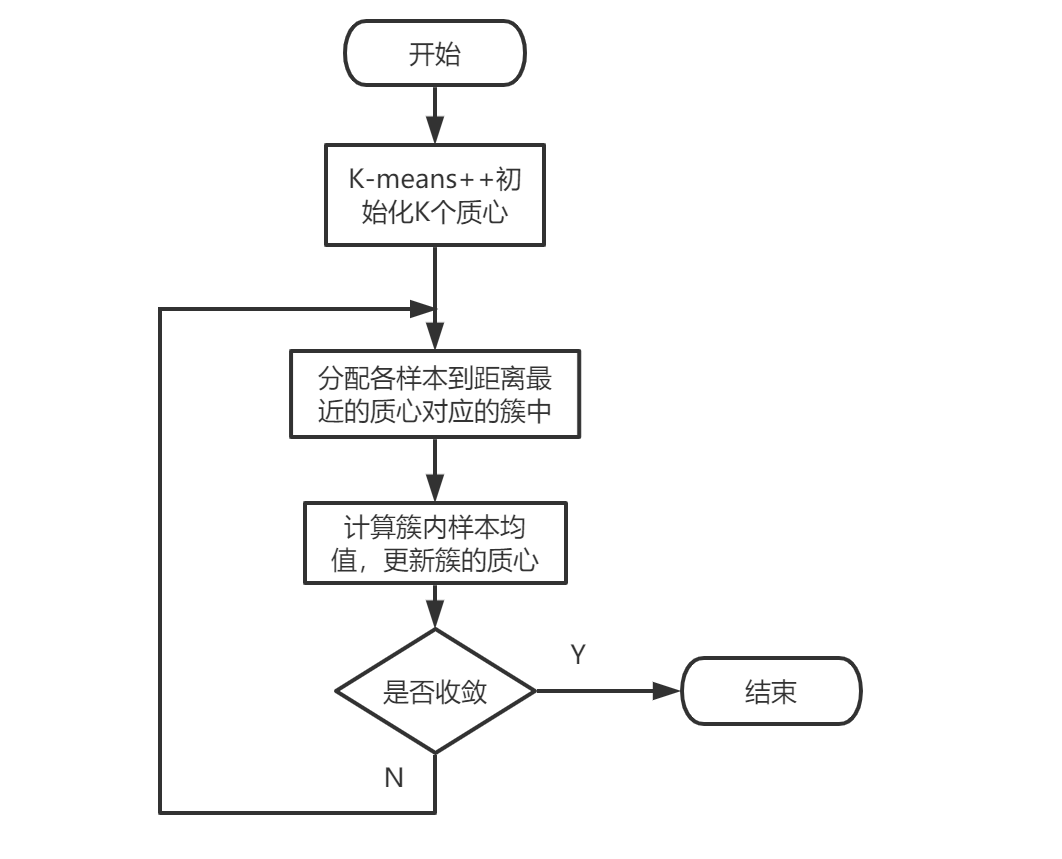


图10 -聚类算法流程图

* 1. 用户密码保护

为了保护用户的隐私，增强系统的安全性，我们使用MD5算法对用户密码进行加密。MD5算法是计算机安全领域广泛使用的一种散列函数，用于提供消息的完整性，是计算机广泛使用的哈希算法之一。在保存用户密码时，MD5算法不记录密码本身，只记录密码的MD5结果，即使数据库被盗也无法反推出明文。此外，MD5算法还具有长度固定、高度的离散性和抗碰撞性等特点。

MD5 算法将输入的信息进行分组，每组512位，顺序处理完所有分组后输出128位结果。在每一组消息的处理中，都要进行4轮、每轮16步、总计64步的处理。其中每步计算中含一次左循环移位，每一步结束时将计算结果进行一次右循环移位。算法过程如下：

|  |
| --- |
| 算法：MD5加密 |
| 1) 要加密的数据进行填充和整理，将要加密的二进制数据对取模，得到的结果如果不够位，则进行补足，补足的方式是第1位填充，后面全部填充； |
| 2) 经过第一步整理完成后的数据的位数可以表示为，再向其后追加位用来存储数据的长度； |
| 3) 在循环处理开始之前，拿个标准数作为输入，分别是： ; |
| 4) 进行轮循环处理，将最后的结果输出。  每一轮处理要循环次，这次循环被分为组，每次循环为一组，每组循环使用不同的逻辑处理函数，处理完成后，将输出作为输入进入下一轮循环。  通过标准 输入，参与每组 计算，得到一个新的值，接着参与下一轮循环运算，最终得到一个值；  具体运算：  这里用到 个逻辑函数，分别对应轮运算，它们将参与运算。（4轮16步）  a)第一轮逻辑函数： 参与第一轮的 步运算  b)第二轮逻辑函数： 参与第二轮的 步运算  c)第三轮逻辑函数： 参与第三轮的16 步运算  d)第四轮逻辑函数： 参与第四轮的16 步运算  再引入一个移位函数，它将整型变量 左循环移 位，如变量 为 位，则。 |

7参考文献

1. Fuad M , Fime A A , Sikder D, et al. Recent Advances in Deep Learning Techniques for Face Recognition[J]. 2021.
2. Zhang K , Zhang Z , Li Z , et al. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks[J]. IEEE Signal Processing Letters, 2016, 23(10):1499-1503.
3. Bodla N , Singh B , Chellappa R , et al. Soft-NMS -- Improving Object Detection With One Line of Code[J]. 2017.
4. Deng J , Guo J , Ververas E , et al. RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild[C]// 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.
5. Guo J , Deng J , Lattas A , et al. Sample and Computation Redistribution for Efficient Face Detection[J]. 2021.
6. Umeyama S . Least-squares estimation of transformation parameters between two point patterns[J]. IEEE Trans.patt.anal.mach.intell, 1991, 13(4):376-380.
7. Sheng C , Yang L , Xiang G , et al. MobileFaceNets: Efficient CNNs for Accurate Real-time Face Verification on Mobile Devices[J]. 2018.
8. Bentley J L . Multidimensional binary search trees used for associative searching[J]. Communications of the ACM, 1975, 18(9):509-517.