

110 人脸检测类

第二步，定义用于人脸检测的类。

110.1 人脸检测类的声明

C:\Users\Minwei\Projects\Qt\Monitor\face.h:

```
1  #ifndef FACE_H
2  #define FACE_H
3
4  #include <QObject>
5
6  class Face : public QObject
7  {
8      Q_OBJECT
9
10 public:
11     explicit Face(QObject* parent = nullptr);
12
13 public slots:
14     void recognize(QImage const& frame);
15 };
16
17 #endif // FACE_H
```

110.2 人脸检测类的实现

C:\Users\Minwei\Projects\Qt\Monitor\face.cpp:

```
1  #include <vector>
2  using namespace std;
3
4  #include <QImage>
5  #include <QDebug>
6
7  #include <opencv2/opencv.hpp>
8  using namespace cv;
9
10 #include "face.h"
11
12 Face::Face(QObject *parent) : QObject(parent)
13 {
14 }
15
16 void Face::recognize(QImage const& frame)
17 {
18     QImage rgb888 = frame.convertToFormat(QImage::Format_RGB888);
19     Mat image = Mat(rgb888.height(), rgb888.width(), CV_8UC3,
20         rgb888.bits(), rgb888.bytesPerLine());
21
22     CascadeClassifier faceClassifier;
23     faceClassifier.load("haarcascade_frontalface_default.xml");
```

```
24
25     Mat gray;
26     cvtColor(image, gray, COLOR_RGB2GRAY);
27     equalizeHist(gray, gray);
28
29     vector<Rect> faces;
30     faceClassifier.detectMultiScale(gray, faces);
31
32     if (faces.empty())
33         qDebug() << "没有人进入监控区";
34     else
35         qDebug() << QString("%1个人进入监控区").arg(faces.size());
36 }
```