

108 人脸识别

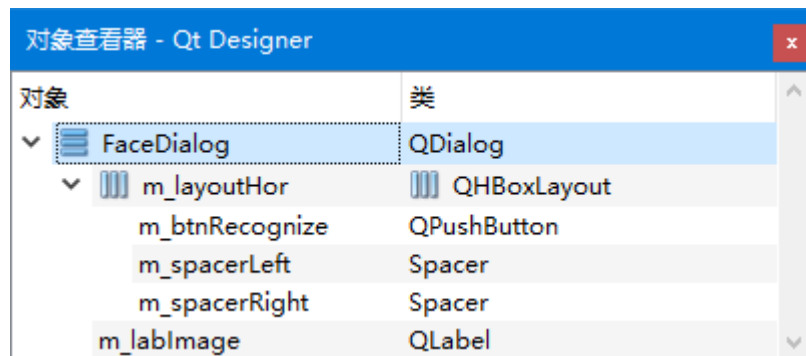
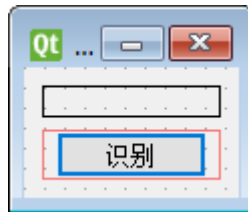
108.1 创建项目

通过QtCreator，在C:\Users\Minwei\Projects\Qt路径下，创建名为Face的项目。

C:\Users\Minwei\Projects\Qt\Face\Face.pro:

```
1  ...
2  INCLUDEPATH += C:\opencv-3.4.3\build\install\include
3  ...
4  LIBS += C:\opencv-3.4.3\build\install\x64\mingw\lib\libopencv_*.a
5  ...
```

108.2 设计界面



C:\Users\Minwei\Projects\Qt\Face\facdialog.ui:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ui version="4.0">
3  <class>FaceDialog</class>
4  <widget class="QDialog" name="FaceDialog">
5  <property name="geometry">
6  <rect>
7  <x>0</x>
8  <y>0</y>
9  <width>107</width>
10 <height>65</height>
11 </rect>
12 </property>
13 <property name="windowTitle">
14 <string>人脸识别</string>
15 </property>
16 <layout class="QVBoxLayout" name="m_layoutVer">
17 <item>
18 <widget class="QLabel" name="m_labImage">
19 <property name="sizePolicy">
```

```
20     <sizepolicy hsiptype="Preferred" vsiptye="Expanding">
21         <horstretch>0</horstretch>
22         <verstretch>0</verstretch>
23     </sizepolicy>
24 </property>
25 <property name="frameShape">
26     <enum>QFrame::Box</enum>
27 </property>
28 </widget>
29 </item>
30 <item>
31     <layout class="QHBoxLayout" name="m_layoutHor">
32         <item>
33             <spacer name="m_spacerLeft">
34                 <property name="orientation">
35                     <enum>Qt::Horizontal</enum>
36                 </property>
37                 <property name="sizeHint" stdset="0">
38                     <size>
39                         <width>40</width>
40                         <height>20</height>
41                     </size>
42                 </property>
43             </spacer>
44         </item>
45         <item>
46             <widget class="QPushButton" name="m_btnRecognize">
47                 <property name="text">
48                     <string>识别</string>
49                 </property>
50                 <property name="default">
51                     <bool>true</bool>
52                 </property>
53             </widget>
54         </item>
55         <item>
56             <spacer name="m_spacerRight">
57                 <property name="orientation">
58                     <enum>Qt::Horizontal</enum>
59                 </property>
60                 <property name="sizeHint" stdset="0">
61                     <size>
62                         <width>40</width>
63                         <height>20</height>
64                     </size>
65                 </property>
66             </spacer>
67         </item>
68     </layout>
69 </item>
70 </layout>
71 </widget>
72 <resources/>
73 <connections/>
74 </ui>
```

108.3 实现功能

将C:\opencv-3.4.3\build\install\etc\haarcascades目录下的haarcascade_frontalface_alt.xml和haarcascade_eye_tree_eyeglasses.xml文件复制到项目目录下。

C:\Users\Minwei\Projects\Qt\Face\facdialog.h:

```
1  #ifndef FACEDIALOG_H
2  #define FACEDIALOG_H
3
4  #include <QDialog>
5
6  #include <opencv2/opencv.hpp>
7  using namespace cv;
8
9  QT_BEGIN_NAMESPACE
10 namespace Ui { class FaceDialog; }
11 QT_END_NAMESPACE
12
13 class FaceDialog : public QDialog
14 {
15     Q_OBJECT
16
17 public:
18     FaceDialog(QWidget *parent = nullptr);
19     ~FaceDialog();
20
21 private slots:
22     void on_m_btnRecognize_clicked();
23
24 private:
25     Ui::FaceDialog *ui;
26     Mat m_image;
27 };
28
29 #endif // FACEDIALOG_H
```

C:\Users\Minwei\Projects\Qt\Face\facdialog.cpp:

```
1  #include <vector>
2  using namespace std;
3
4  #include "facdialog.h"
5  #include "ui_facdialog.h"
6
7  FaceDialog::FaceDialog(QWidget *parent)
8      : QDialog(parent)
9      , ui(new Ui::FaceDialog)
10 {
11     ui->setupUi(this);
12
13     cvtColor(imread(QString("tahiti.jpg")).toLatin1().data(), m_image,
14             COLOR_BGR2RGB);
15
16     ui->m_labImage->setPixmap(QPixmap::fromImage(QImage(
```

```

16         m_image.data, m_image.cols, m_image.rows, QImage::Format_RGB888).
17         scaled(QSize(m_image.cols / 2, m_image.rows / 2),
Qt::KeepAspectRatio));
18
19     setWindowFlag(Qt::MSWindowsFixedSizeDialogHint);
20 }
21
22 FaceDialog::~FaceDialog()
23 {
24     delete ui;
25 }
26
27 void FaceDialog::on_m_btnRecognize_clicked()
28 {
29     ui->m_labImage->setPixmap(QPixmap::fromImage(QImage(
30         m_image.data, m_image.cols, m_image.rows, QImage::Format_RGB888).
31         scaled(QSize(m_image.cols / 2, m_image.rows / 2),
Qt::KeepAspectRatio));
32     ui->m_labImage->repaint();
33
34     CascadeClassifier faceClassifier;
35     faceClassifier.load("haarcascade_frontalface_alt.xml");
36     CascadeClassifier eyesClassifier;
37     eyesClassifier.load("haarcascade_eye_tree_eyeglasses.xml");
38
39     Mat gray;
40     cvtColor(m_image, gray, COLOR_RGB2GRAY);
41     equalizeHist(gray, gray);
42
43     Mat canvas = m_image.clone();
44
45     vector<Rect> faces;
46     faceClassifier.detectMultiScale(gray, faces);
47     for (Rect const& face : faces)
48     {
49         ellipse(canvas,
50             Point(face.x + face.width / 2, face.y + face.height / 2),
51             Size(face.width / 2, face.height / 2), 0, 0, 360,
52             Scalar(0, 255, 0), 6, 8, 0);
53
54         vector<Rect> eyes;
55         eyesClassifier.detectMultiScale(gray(face), eyes);
56         for (Rect const& eye : eyes)
57             ellipse(canvas,
58                 Point(face.x + eye.x + eye.width / 2, face.y + eye.y +
eye.height / 2),
59                 Size(eye.width / 2, eye.height / 2), 0, 0, 360,
60                 Scalar(0, 255, 0), 6, 8, 0);
61     }
62
63     ui->m_labImage->setPixmap(QPixmap::fromImage(QImage(
64         canvas.data, canvas.cols, canvas.rows, QImage::Format_RGB888).
65         scaled(QSize(canvas.cols / 2, canvas.rows / 2),
Qt::KeepAspectRatio));
66 }

```

108.4 测试验证

运行效果如图所示：

