Face Detection Program

By: Shaza Ismail Kaoud

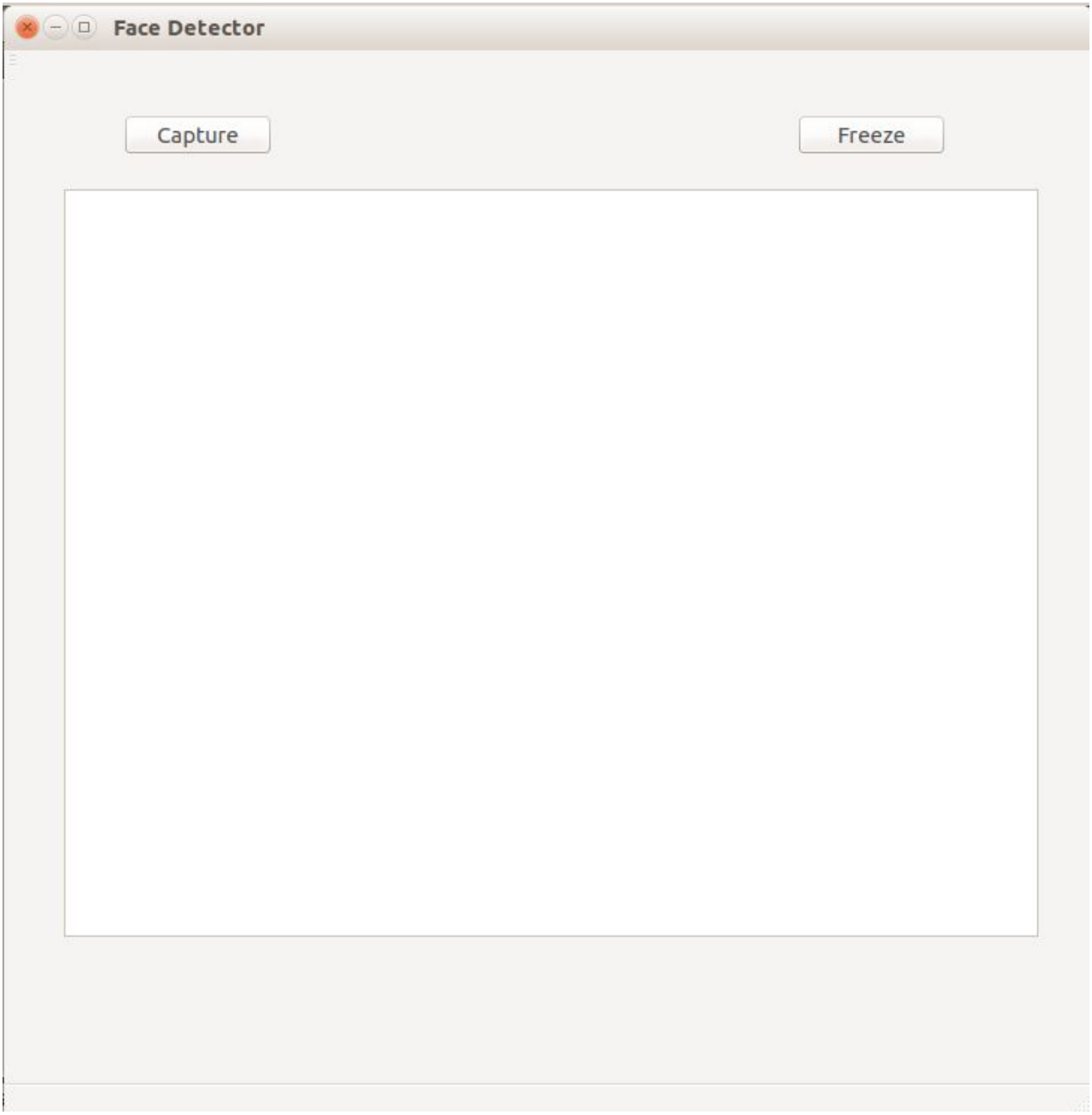
Table of Contents:

1. Program Description:

A simple GUI application for face detection, using the laptop's webcam and drawing rectangles around faces facing the camera. The application is developed with Qt 5.5 with C++ and opencv 3.

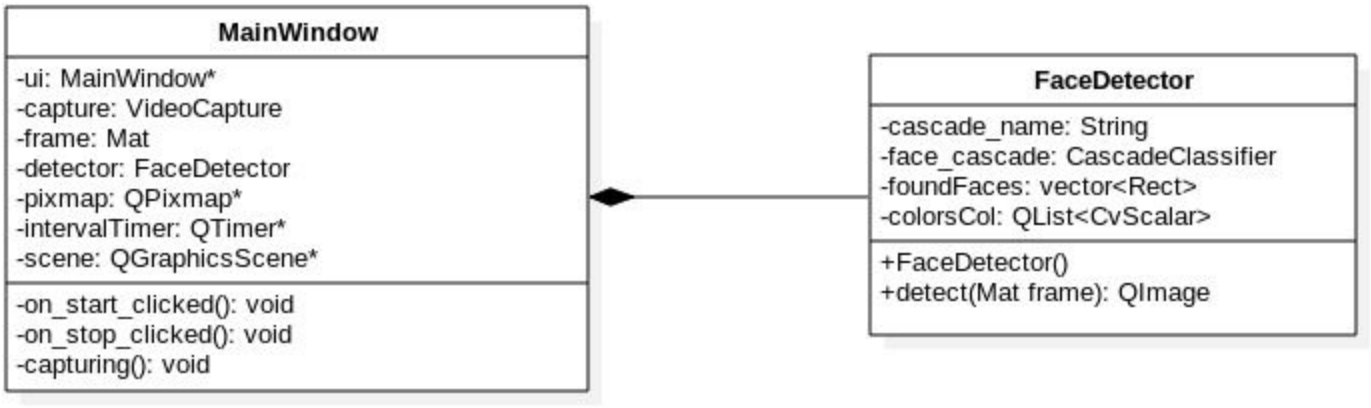
The interface:

A simple two buttons' interface to capture and freeze the current captured view that looks initially:



2. The Software Design:

2.1 The class diagram of the code:



2.2 The detection part of the code (the slot/method detect(Mat frame)):

```
QImage FaceDetector::detect(Mat frame)
{
    Mat frame_gray;
    cvtColor(frame, frame_gray, COLOR_BGR2GRAY);
    equalizeHist(frame_gray, frame_gray);

   //Detection
    face_cascade.detectMultiScale(frame_gray,
                                  foundFaces,
                                   1.1, 2, 0|CASCADE_SCALE_IMAGE, Size(100, 100));

    int n = foundFaces.size();
    //Draw rectangles around faces
    CvRect rect;
    for(int i = 0; i < n ; i++)
    {
       rect = foundFaces[i];
       rectangle(frame, cvPoint(rect.x, rect.y),
                 cvPoint((rect.x + rect.width), (rect.y + rect.height)),
                 colorsCol[i%8]);
    }
```

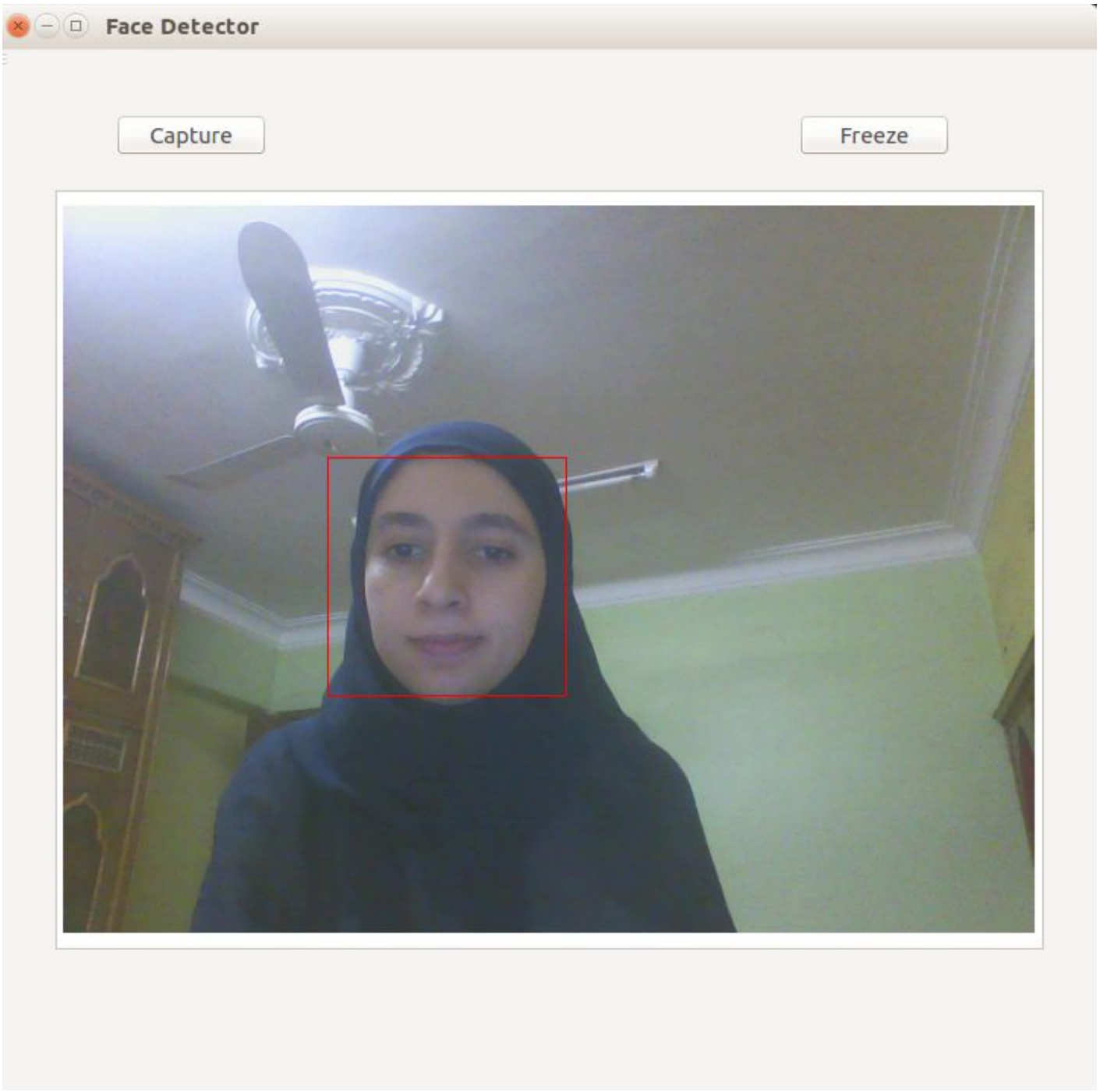
```
    // Convert Mat to QImage
    QImage result = QImage((const uchar *)frame.data,
                           frame.cols,
                           frame.rows,
                           frame.step,
                           QImage::Format_RGB888).rgbSwapped();
    return result;
}
```

2.3 Calling the detector from MainWindow using the timer connected to a slot:

```
  void MainWindow::on_start_clicked()
{
        capture.open(-1);
        connect(intervalTimer, SIGNAL(timeout()), this, SLOT(capturing()));
        intervalTimer->start(33);
}
void MainWindow::capturing()
{
        capture.read(frame);
        QImage result = detector.detect(frame);
        pixmap = new QPixmap(QPixmap::fromImage(result));
        scene = new QGraphicsScene(ui->graphicsView);
        scene->addPixmap(*pixmap);
        scene->setSceneRect(pixmap->rect());
        ui->graphicsView->setScene(scene);
}
```

3. The Program Running:

Test:



4. References:

The main references are the opencv3.0+ and Qt 5.5+ documentation listed and some sub-links of them

[1] Cascade Classification
http://docs.opencv.org/3.0-beta/modules/objdetect/doc/cascade_classification.html
[2] Face Detection using Haar Cascades
http://docs.opencv.org/master/d7/d8b/tutorial_py_face_detection.html#gsc.tab=0
[3] Qt documentation of QTimer
http://doc.qt.io/qt-5/qtimer.html
[4] Qt documentation of QImage
http://doc.qt.io/qt-5/qimage.html
[5] Qt documentation of QGraphicsView
http://doc.qt.io/qt-5/qgraphicsview.html