```cpp
#include "opencv2/objdetect/objdetect.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"

#include <iostream>
#include <stdio.h>

using namespace std;
using namespace cv;
void detectAndDisplay( Mat frame );

//把"haarcascade_frontalface_alt.xml"和"haarcascade_eye_tree_eyeglasses.xml"这两
String face_cascade_name = "/home/yinan/objects(opencv)/haarcascade_frontalface
String eyes_cascade_name = "/home/yinan/objects(opencv)/haarcascade_eye_tree_ey
CascadeClassifier face_cascade;
CascadeClassifier eyes_cascade;
string window_name = "Capture - Face detection";
RNG rng(12345);

static void ShowHelpText()
{
     cout<< "\n\n\t\t\t    当前使用的OpenCV版本为: " << CV_VERSION
        <<"\n\n   ----------------------------------------------------------------
}

int main( void )
{
  VideoCapture capture;
  Mat frame;
  //-- 1. 加载级联 (cascades)
  if( !face_cascade.load( face_cascade_name ) ){ printf("--(!)Error loading\n")
  if( !eyes_cascade.load( eyes_cascade_name ) ){ printf("--(!)Error loading\n")
  //-- 2. 读取视频
  capture.open(0);
  ShowHelpText();
  if( capture.isOpened() )
  {
    for(;;)
    {
      capture >> frame;
      //-- 3. 对当前帧使用分类器 (Apply the classifier to the frame)
      if( !frame.empty() )
       { detectAndDisplay( frame ); }
      else
       { printf(" --(!) No captured frame -- Break!"); break; }
      int c = waitKey(10);
      if( (char)c == 'c' ) { break; }
      //waitKey(0);会一帧一卡
    }
  }

  return 0;
}
```

```
void detectAndDisplay( Mat frame )
{
   std::vector<Rect> faces;
   Mat frame_gray;
   cvtColor( frame, frame_gray, COLOR_BGR2GRAY );
   equalizeHist( frame_gray, frame_gray );

   //人脸检测
   face_cascade.detectMultiScale( frame_gray, faces, 1.1, 2, 0|CASCADE_SCALE_IM
   for( size_t i = 0; i < faces.size(); i++ )
    {
      Point center( faces[i].x + faces[i].width/2, faces[i].y + faces[i].height
      ellipse( frame, center, Size( faces[i].width/2, faces[i].height/2), 0, 0
      Mat faceROI = frame_gray( faces[i] );
      std::vector<Rect> eyes;

      //-- 在脸中检测眼睛
      eyes_cascade.detectMultiScale( faceROI, eyes, 1.1, 2, 0|CASCADE_SCALE_IMA
      for( size_t j = 0; j < eyes.size(); j++ )
       {
         Point eye_center( faces[i].x + eyes[j].x + eyes[j].width/2, faces[i].y
         int radius = cvRound( (eyes[j].width + eyes[j].height)*0.25 );
         circle( frame, eye_center, radius, Scalar( 255, 0, 0 ), 3, 8, 0 );
       }
    }
   //-- 显示最终效果图
   imshow( window_name, frame );
}
```