facedetect.cpp

```cpp
// CPP program to detects face in a video

// Include required header files from OpenCV directory
#include "opencv2/objdetect.hpp"
#include "opencv2/highgui.hpp"
#include "opencv2/imgproc.hpp"
#include <iostream>

using namespace std;
using namespace cv;

// Function for Face Detection
void detectAndDraw( Mat& img, CascadeClassifier& cascade,
                CascadeClassifier& nestedCascade, double scale );
string cascadeName, nestedCascadeName;

int main( int argc, const char** argv )
{
    // VideoCapture class for playing video for which faces to be detected
    VideoCapture capture;
    Mat frame, image;

    // PreDefined trained XML classifiers with facial features
    CascadeClassifier cascade, nestedCascade;
    double scale=1;

    // Load classifiers from "opencv/data/haarcascades" directory
    nestedCascade.load( "./haarcascade_eye_tree_eyeglasses.xml" ) ;

    // Change path before execution
    cascade.load( "./haarcascade_frontalcatface.xml" ) ;

    // Start Video..1) 0 for WebCam 2) "Path to Video" for a Local Video
    capture.open(0);
    if( capture.isOpened() )
    {
        // Capture frames from video and detect faces
        cout << "Face Detection Started...." << endl;
        while(1)
        {
            capture >> frame;
            if( frame.empty() )
                break;
            Mat frame1 = frame.clone();
            detectAndDraw( frame1, cascade, nestedCascade, scale );
            char c = (char)waitKey(10);

            // Press q to exit from window
            if( c == 27 || c == 'q' || c == 'Q' )
                break;
        }
    }
    else
```

```
54          cout<<"Could not Open Camera";
55      return 0;
56  }
57
58  void detectAndDraw( Mat& img, CascadeClassifier& cascade,
59                      CascadeClassifier& nestedCascade,
60                      double scale)
61  {
62      vector<Rect> faces, faces2;
63      Mat gray, smallImg;
64
65      cvtColor( img, gray, COLOR_BGR2GRAY ); // Convert to Gray Scale
66      double fx = 1 / scale;
67
68      // Resize the Grayscale Image
69      resize( gray, smallImg, Size(), fx, fx, INTER_LINEAR );
70      equalizeHist( smallImg, smallImg );
71
72      // Detect faces of different sizes using cascade classifier
73      cascade.detectMultiScale( smallImg, faces, 1.1,  2, 0|CASCADE_SCALE_IMAGE,
    Size(30, 30) );
74
75      // Draw circles around the faces
76      for ( size_t i = 0; i < faces.size(); i++ )
77      {
78          Rect r = faces[i];
79          Mat smallImgROI;
80          vector<Rect> nestedObjects;
81          Point center;
82          Scalar color = Scalar(255, 0, 0); // Color for Drawing tool
83          int radius;
84
85          double aspect_ratio = (double)r.width/r.height;
86          if( 0.75 < aspect_ratio && aspect_ratio < 1.3 )
87          {
88              center.x = cvRound((r.x + r.width*0.5)*scale);
89              center.y = cvRound((r.y + r.height*0.5)*scale);
90              radius = cvRound((r.width + r.height)*0.25*scale);
91              circle( img, center, radius, color, 3, 8, 0 );
92          }
93          else
94              // rectangle( img, cvPoint(cvRound(r.x*scale), cvRound(r.y*scale)),
    cvPoint(cvRound((r.x + r.width-1)*scale), cvRound((r.y + r.height-1)*scale)), color,
    3, 8, 0);
95          if( nestedCascade.empty() )
96              continue;
97          smallImgROI = smallImg( r );
98
99          // Detection of eyes in the input image
100         nestedCascade.detectMultiScale( smallImgROI, nestedObjects, 1.1, 2,
    0|CASCADE_SCALE_IMAGE, Size(30, 30) );
101
102         // Draw circles around eyes
103         for ( size_t j = 0; j < nestedObjects.size(); j++ )
104         {
105             Rect nr = nestedObjects[j];
106             center.x = cvRound((r.x + nr.x + nr.width*0.5)*scale);
```

```
107             center.y = cvRound((r.y + nr.y + nr.height*0.5)*scale);
108             radius = cvRound((nr.width + nr.height)*0.25*scale);
109             circle( img, center, radius, color, 3, 8, 0 );
110         }
111     }
112
113     // Show Processed Image with detected faces
114     imshow( "Face Detection", img );
115 }
116
```