```
#include "highgui.h"
#include "cv.h"

CvHaarClassifierCascade* load_object_detector(const char* cascade_path)
     {
        return (CvHaarClassifierCascade*) cvLoad(cascade_path);
     }

void detect_and_draw_objects(IplImage* image,
                                   CvHaarClassifierCascade* cascade,
                                   int do_pyramids)
    {
       IplImage* sam_image = image;
       CvMemStorage* storage = cvCreateMemStorage(0);
       CvSeq* faces;
       int i, scale = 1;

            if (do_pyramids)
                {
                  sam_image = cvCreateImage(cvSize(image->width / 2, image->height / 2),
IPL_DEPTH_8U, 3);
                  cvPyrDown(image, sam_image, CV_GAUSSIAN_5x5);
                  scale = 2;
                }

           faces = cvHaarDetectObjects(sam_image, cascade, storage, 1.2, 2,
                                              CV_HAAR_DO_CANNY_PRUNING);

                for (i = 0; i < faces->total; i++)
                   {
                    CvRect face_rect = *(CvRect*) cvGetSeqElem(faces, i);
                    cvRectangle(image, cvPoint(face_rect.x*scale, face_rect.y*scale),
                                          cvPoint((face_rect.x + face_rect.width)*scale,
                                                  (face_rect.y + face_rect.height)*scale),
                                                        CV_RGB(0, 255, 0), 3);
                   }

                if (sam_image != image)
                      cvReleaseImage(&sam_image);
                      cvReleaseMemStorage(&storage);
       }

int main(int argc, char** argv)
  {
     IplImage* image;
     if (image = cvLoadImage("lena.jpg", 1))
     {
          CvHaarClassifierCascade* cascade
                          =load_object_detector("haarcascade_frontalface_alt.xml”);
          detect_and_draw_objects(image, cascade, 1);
          cvNamedWindow("test", 1);
          cvShowImage("test", image);
          cvWaitKey(0);
          cvReleaseHaarClassifierCascade(&cascade);
          cvReleaseImage(&image);
     }
     return 0;
}
```