OpenCV Tutorial 11 - Chapter 13

Author: Noah Kuntz (2009)
Contact: nk752@drexel.edu

Keywords: OpenCV, computer vision, machine learning, haar classifier, face detection

My Vision Tutorials Index

This tutorial assumes the reader:
(1) Has a basic knowledge of Visual C++
(2) Has some familiarity with computer vision concepts
(3) Has read the previous tutorials in this series

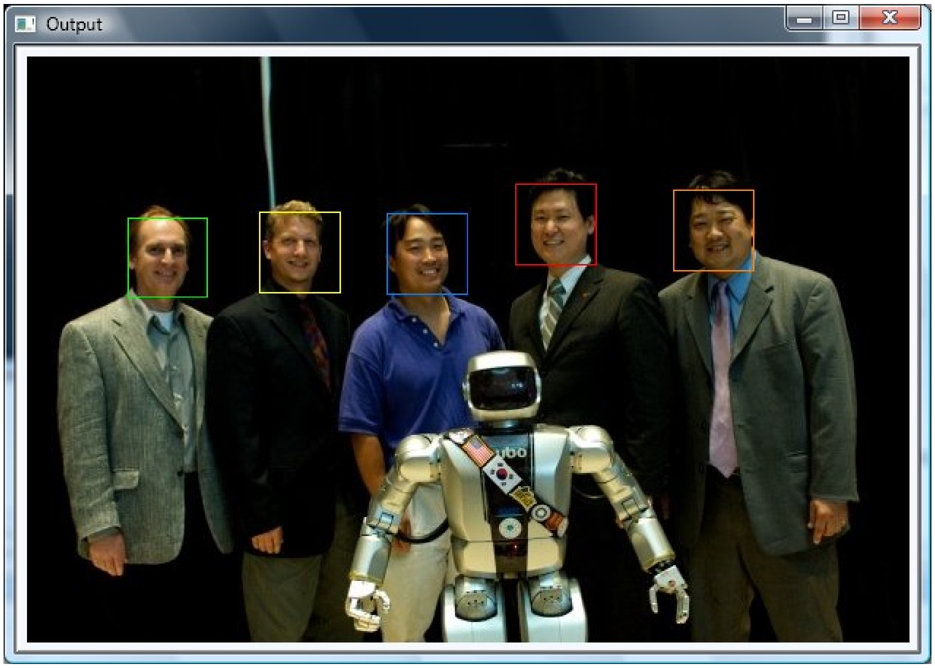
The rest of the tutorial is presented as follows:

- Step 1: Face Detection with the Haar Classifier
- Final Words

Important Note!

More information on the topics of these tutorials can be found in this book: Learning OpenCV: Computer Vision with the OpenCV Library

Step 1: Face Detection with the Haar Classifier



Face Detection

Machine learning is a powerful set of algorithms that can be used to improve computer vision algorithms by allowing the program to build on the knowledge of many example cases. For face detection, this example uses a Haar classifier, which is a statistically boosted classifier supported by training on many sample images of faces. Much more detail of machine learning theory can be found in the book. This code uses a *CvHaarClassifierCascade** matrix that should be copied from your OpenCV directory to your working directory (see the code comments), and loaded with *cvLoad*. Then *cvHaarDetectObjects* takes the input image, the loaded cascade, the empty memory storage and a few options and performs the face detection. The important option here is the minimum size defined in *cvSize*. Base this on the size of the faces in your image, in pixels. Then once the detected faces are stored in a sequence, a loop can be used to go through them and draw a *CvRect* object around each one. Here is the code:

```
int _tmain(int argc, _TCHAR* argv[])
```

```
{
        IplImage* img;
        img = cvLoadImage( "dasl_hubo.jpg" );
        CvMemStorage* storage = cvCreateMemStorage(0);
        // Note that you must copy C:\Program Files\OpenCV\data\haarcascades\haarcascade_frontalface_alt2.xml
        // to your working directory
        CvHaarClassifierCascade* cascade = (CvHaarClassifierCascade*)cvLoad( "haarcascade_frontalface_alt2.xml" );
        double scale = 1.3;

        static CvScalar colors[] = { {{0,0,255}}, {{0,128,255}}, {{0,255,255}},
        {{0,255,0}}, {{255,128,0}}, {{255,255,0}}, {{255,0,0}}, {{255,0,255}} };

        // Detect objects
        cvClearMemStorage( storage );
        CvSeq* objects = cvHaarDetectObjects( img, cascade, storage, 1.1, 4, 0, cvSize( 40, 50 ));

        CvRect* r;
        // Loop through objects and draw boxes
        for( int i = 0; i < (objects ? objects->total : 0 ); i++ ){
                r = ( CvRect* )cvGetSeqElem( objects, i );
                cvRectangle( img, cvPoint( r->x, r->y ), cvPoint( r->x + r->width, r->y + r->height ),
                        colors[i%8]);
        }

        cvNamedWindow( "Output" );
        cvShowImage( "Output", img );
        cvWaitKey();

        cvReleaseImage( &img );

        return 0;
}
```

Final Words

This tutorial's objective was to show how to use a basic machine learning classifier to detect faces.

Click here to email me.
Click here to return to my Tutorials page.