void detectAndDraw(Mat& img, CascadeClassifier & cascade, double scale)

{

int i = 0;

double t = 0;

vector<Rect> faces;

const static Scalar colors[] = {CV\_RGB(0,0,255),

CV\_RGB(0,128,255),

CV\_RGB(0,255,255),

CV\_RGB(0,255,0),

CV\_RGB(255,128,0),

CV\_RGB(255,255,0),

CV\_RGB(255,0,0),

CV\_RGB(255,0,255)};

Mat gray, smallImg (cvRound (img.rows/scale), cvRound (img.cols/scale), CV\_8UC1);

cvtColor(img, gray, CV\_BGR2GRAY);

resize(gray, smallImg, smallImg.size(), 0, 0, INTER\_LINEAR);

equalizeHist(smallImg, smallImg);

cascade.detectMultiScale(smallImg, faces, 1.1, 2, 0

|CV\_HAAR\_FIND\_BIGGEST\_OBJECT

|CV\_HAAR\_DO\_ROUGH\_SEARCH //Size(30,30);

|CV\_HAAR\_SCALE\_IMAGE

//Size(30,30);

for(vector<Rect>::const\_iterator r = faces.begin(); r != faces.end(); r++, i++)

{

Scalar color = colors[i%8];

rectangle(img, cvPoint(r->x\*scale,r->y\*scale), cvPoint((r->x+r->width)\*

scale,(r-> y+r->height)\*scale), color,3,8,0);

}

}