Shane Zabel

Detection Project Description

Detect Wink Training Method:

I used the OpenCV HAAR cascades:

string FACES_CASCADE_NAME = cascades + "haarcascade_frontalface_alt.xml";

string EYES_CASCADE_NAME = cascades + "haarcascade_eye.xml";

I histogram equalized and ran a 3 pixel median blur on the images:

equalizeHist(frame_gray, frame_gray); // input, outuput

medianBlur(frame_gray, frame_gray, 3); // input, output, neighborhood_size

For faces I used a scaling factor of 1.05, 2 neighbors and a min size of (40,40):

cascade_face.detectMultiScale(frame_gray, faces, 1.05, 2, 0|CASCADE_SCALE_IMAGE, Size(40,40));

For eyes I used a scaling factor of 1.025, 70 neighbors and a min size of (5,5):

cascade.detectMultiScale(ROI, eyes, 1.025, 70, 0, Size(5,5));Detect Fingers:

Detect Fingers Training Method:

The 3 positive training images are found at:
CreateSamples/fingers/pos/*.jpg

The 10 negative training images are found at:
CreateSamples/fingers/neg/*.png

For opencv_createsamples I created 100 positive images for each of the 3 positive images with parameters width,height = 30,30; max x and y angles of 0.1; and max z angle of 0.3.

opencv_createsamples -img fingers/pos/fin$a.png -bg fingers/neg.txt -info fingers/crop$a.txt -num 100 -w 30 -h 30 -maxxangle 0.1 -maxyangle 0.1 -maxzangle 0.3

opencv_createsamples -info fingers/crop.txt -bg fingers/neg.txt -vec fingers/cropped.vec -num 300 -w 30 -h 30

For opencv_traincascade I used the following parameters. Width,height=30,30; number of positive images=200; number of negative images=600; number of stages=10; minimum hit rate=0.995; and maximum false alrm rate=0.1

opencv_traincascade -data . -vec cropped.vec -bg neg.txt -w 30 -h 30 -numPos 200 -numNeg 600 -numStages 10 -precalcValBufSize 2048 -precalcIdxBufSize 2048 -minHitRate 0.995 -maxFalseAlarmRate 0.1 -featureType HAAR

In the main program I histogram equalized and ran a 5 pixel median blur

equalizeHist(frame_gray, frame_gray);

medianBlur(frame_gray, frame_gray, 5);

For the cascade detector I ran scaling of 1.025, 30 neighbors and a min size of 10x10 pixels

cascade.detectMultiScale(frame_gray, detections, 1.025, 30, 0|CASCADE_SCALE_IMAGE, Size(10,10));