

The process

Jan Podmajersky

Smile detection is a popular feature of today's photo cameras. It is not implemented in all cameras, as a popular face detection, because it is more complicated to implement. This project shows a basic algorithm in the topic. It may be used but few improvements are necessary. Sobel filter and thresholding are used. There is a mask which is compared to every filtered image from a webcam. If the images are more than 60% equal, smile is detected.

Used Functions detectMultiScale, Sobel, medianBlur, threshold, dilate, bitwise_and

1. convert image from camera to gray scale

```
1 cvtColor( frame, frame_gray, CV_BGR2GRAY );
```

2. face detection using Haar cascade

```
1 face_cascade.detectMultiScale( frame_gray, faces, 1.3, 4,  
CV_HAAR_DO_CANNY_PRUNING, Size(50, 50) );
```

3. adjust size of image just to the detected face

4. cut only one third of the face, where mouth are always located

```
1 face = frame_gray( cv::Rect(faces[i].x, faces[i].y + 2 *  
faces[i].height/3, faces[i].width, faces[i].height/3) );
```

5. horizontal sobel filter

```
1 Sobel( face, grad_y, ddepth, 0, 1, 7, scale, delta,  
BORDER_DEFAULT );  
2 addWeighted( abs_grad_y, 0.9, abs_grad_y, 0.9, 0, output );
```

6. Median blur

```
1 medianBlur(output, detected_edges, 5);
```

7. threshold the image

```
1 threshold(detected_edges, detected_edges, 220, 255,  
CV_THRESH_BINARY);
```

8. dilate small parts

```
1 dilate(detected_edges, detected_edges, element);
```

9. logical and the image and mask image

```
1 | bitwise_and(detected_edges,maskImage,result);
```

10. detect smile

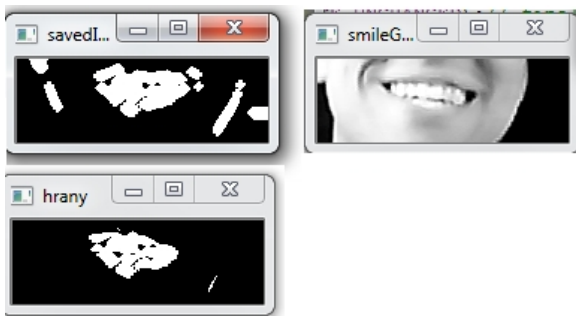
if the images are 60% equal there is a smile



input



horizontal Sobel filter



masked image



output

Viewed using [Just Read](#)