DSP Lab Project Report

Real-Time Eye Blink Detection Based on Eye Aspect Ratio and SVM

Heqing Chen(hc1988), Ziqi Peng(zp436), Zhaobin Li(zl1696)

Tandon School of Engineering

New York University

Problem Definition

Eye blink detection is a practical problem in our daily life and several applications based on the eye blink has been put forward such as making software to help people who aren't able to write or speck to better communicate with others, detecting driver drowsiness in the night to improving safety or to enhance the security of face recognition by anti-spoofing.

However, the professional way to detect eye blink need to use some special devices like of a pair of glasses with infrared cameras inside or an EGG(electroencephalograph) detector. And those ways for eye detection are complex and of high cost.

At the same time, several ways from the computer vision aspect were put forward such as using sliding window or SIFT descriptor to detect the eye area and adapt optical flow methods to examine the eyelid states. And though these methods are accurate, they are usually done with powerful GPUs, which has high computational cost.

Based on the paper [1], in our project, we first built a framework to detect the landmarks on the face in real-time video with the backend of dlib library. And then we extracted several needed landmarks to computer the Eye Aspect Ratio(EAR) to present the eye's states. To apply the EAR to examine the eye blink, we designed two experiments, one with a fixed threshold and another one with a pretrained SVM model.

Introduction of state-of-the art facial landmark detectors

The first and a very important step to detect eye blinks based on ear is to extract the facial landmarks in real time video streams. Those facial landmarks can be used to locate different parts on a face just like eyes or mouth. And the accuracy is essential to the eye blink project, too, since the further EAR computation is totally based on those landmark points.

Nowadays there are several robust facial landmark detectors such as Dlib or CLM-framework, and the error of the landmark localization is usually below five percent of the inter-ocular distance.

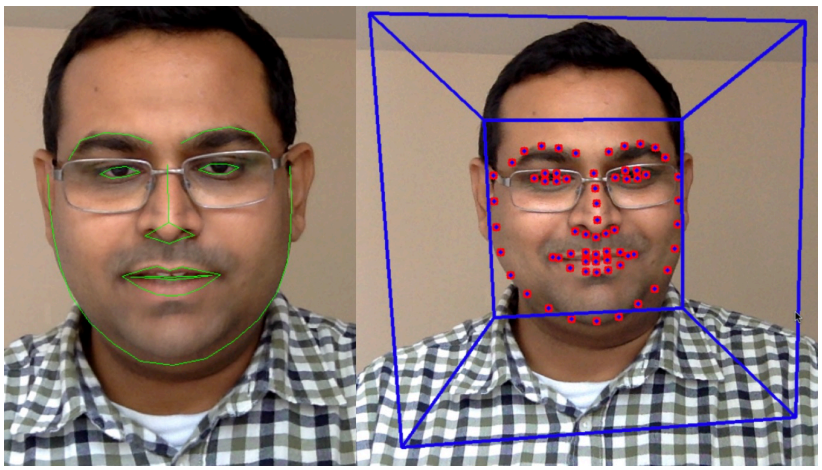


Figure 1 Landmarks on a face produced by Dlib(left) and CLM-framework(right). [2]

Dlib

Dlib is a very useful open source C++ toolkit, which contains a lot of machine learning and other algorithms. It is widely used in industry and academic fields, such as the auto-pilot system for the robots, the recognition system in the surveillance devices, as well as mobile applications. And except for C++, it also has a Python interface.

CLM-framework

CLM-framework is another famous library that provides with stable facial landmark detectors. The detector in the library not only could detect the face contour in real time, it also could provide the information of direction in which the tester is looking at, as showing in figure

In our project, we choose Dlib library as the backend since it is an open source library and has a very fast compute speed which is preferred in real-time use. Besides we adapted the pretrained facial landmark predictor that is available at Dlib's official website.

The concept of eye aspect ratio(EAR)

Using dlib library, we can extract 68 facial landmark points on a face, and among all those 68 points, 12 points are used to localize the left and the right eye. The eye aspect ratio is mainly computed based on the coordinates of those 12 eye points.

The eye aspect ratio is defined as

$$EAR = \frac{||p_2 - p_6|| + ||p_3 - p_5||}{||p_1 - p_4||}$$

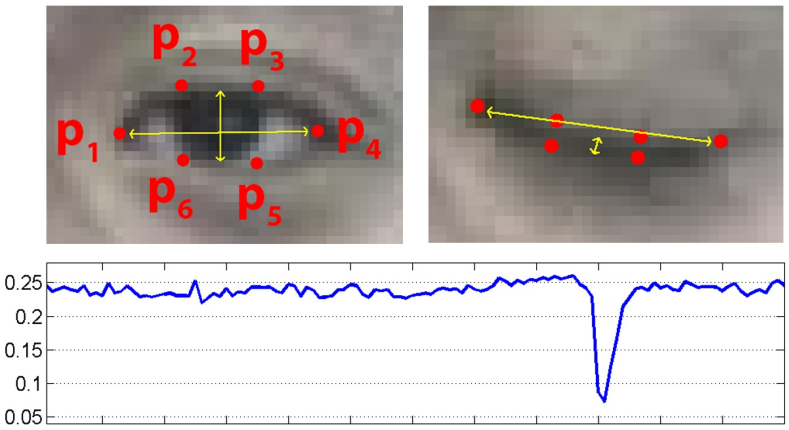


Figure 2 The detected eye landmarks in different eye-opening state and an EAR graph with a single blink. [1]

Where $p_1$to p_6 are coordinates of the eye landmarks detected by the detector and ||…|| means Euclidean distance.

First experiment

Since one regular blink process can have a time length of 100~400ms, in a video stream of 30fps, the total process of opened-closed-opened could be accomplished in 3~12 frames. Besides, according our observation, the EAR will approximately be bigger than 0.25 when the eyes keep an opened state. And when the eyes start to close, the EAR will decrease to a local minimum and then increase to the normal number after an opened-closed-opened process. So, in our first experiment we choose a threshold of 0.22 and decided the eye to be closed if the EAR was below 0.22 for a consecutive 3 frames.

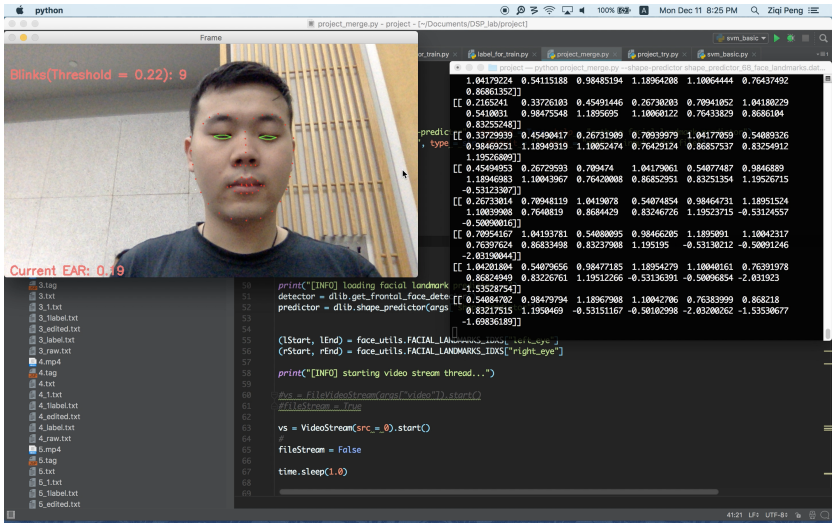


Figure 3 Screenshot of our first experiment

The consequence of the experiment is largely satisfied, but we also found several problems from the experiment.

Problems found in experiment1

In the couples of experiments we did, we set the threshold from 0.20 to 0.25 with the step 0.01 once a time and found that for different people who performed the test, the proper EAR threshold and the number of consecutive frames for the best eye blink detection results can be much different.

If we set a relatively large threshold, the classifier can be sensitive to the state changes of the eye. At the meantime, a large threshold also leads to some misclassification such as when the tester shook his head or made some facial expressions.

On the contrary, if we set the threshold too small, and the classifier seems to be more robust to the movement of the head. But sometimes it will ignore a quick unconscious eye blink of the tester.

Therefore, to get a better performance, we need to tune the EAR threshold several times for one tester.

Besides, as for the number of consecutive frames, it can also be regarded as a parameter to change the performance of the classifier. But unlike the threshold, it is a more universal to tune for all the people (mainly to adapt to different working environments). In our experiment, we set this parameter to 3, which means if the EARS from 3 consecutive frames are all below the threshold, we believe that there's an eye blink. And if we want to detect very fast eye blinks, we may need to change the consecutive frames to 2.

EAR SVM

To make the eye state classifier more robust and accurate, we used a Support Vector Machine to do the further classification job.

Super vector machine is a popular machine learning method used in medium size project. Compared with the simpler logistic regression or linear regression method, SVM is more robust to the abnormal data points and can have a better in-all performance. Besides, compared with nowadays very big scales neural networks like VGG-15, SVM can still get a good performance

when the features of the input are not too big and the training set is less than 100 thousand; further, the SVM just need a small computational cost and a fairly less time to train the model.

Since most video streams are captured with the speed of 30fps and one blink could last 100 to 400ms, in order to utilize characteristic of continuity in frames of the eye blink, we took the N-6 frames, Nth frame and N+6 frames to construct as our 13-dimension input features to predict the Nth frame's eye state.

To make the input data points separable, we introduced the Gaussian kernel.

The Gaussian kernel:

$$k(x,z) = \exp\left(-\frac{||x-z||^2}{2\sigma^2}\right)$$

which is used to map our input features to a high dimension where the features are linear separable.

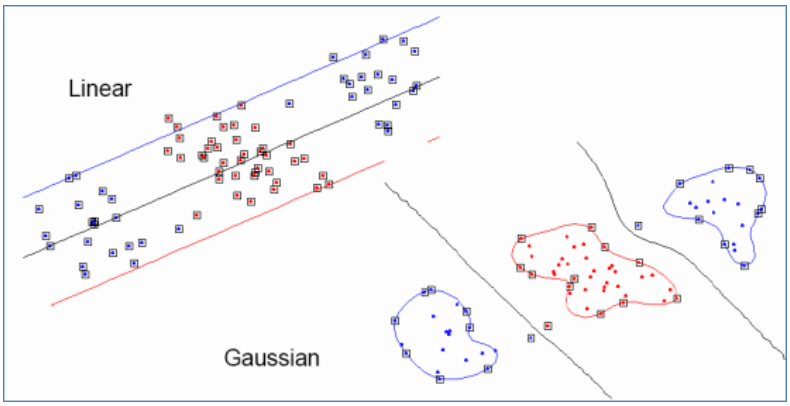


Figure 4 Data points that were linear inseparable(left) and became linear separable after using the Gaussian kernel(right). [3]

Dataset for training EAR SVM

There are many eye blink detections available, and mainly divided as picture datasets and video datasets. Since we are building a real-time eye blink detection system, we choose the video datasets. Two video eye blink datasets are ZJU [4] and Eyeblink8 [5], which created by Zhejiang University and Slovak University of Technology, respectively.

ZJU

ZJU dataset is a kind of normal eye-blink dataset because the people in the dataset videos keeps a fixed distance to the camera and make counted eye blinks. Besides, people in the videos don't have movement or make other expressions. So, we can look the ZJU as a standard eye blink dataset.

Eyeblink8

Eyeblink8 was an eye-blink dataset that comprise of 8 videos captured at 30fps with a resolution of 640*480. Every video has 4000 to 15000 frames and the frames have labels like "closed", "half" when there is a potential eye blink movement.

The person in the videos are mostly facing towards the camera and having some nature face movements as talking, yawning and frowning. Besides, person in one of the video wears glasses in order to make the dataset more robust.

In our project, to train the ear SVM, we choose the Eyeblink8 dataset since it is a dataset that contains more environments than ZJU, such as people's shaking head, making facial expression

or talking to others, which we think can help us build a robust SVM model to detect eye blinks in real life.

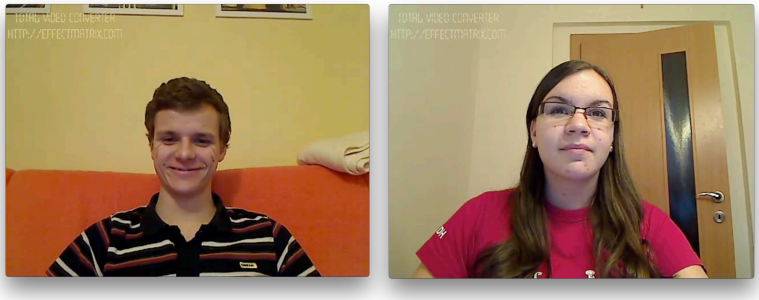


Figure 5 Two screenshots of the videos in the Eyeblink8

Preprocessing for the training dataset

We used the facial landmark descriptor pretrained with dlib library to detect the facial landmarks in every frame of the Eyeblink8, then computed the EAR of every frame of 8 videos in Eyeblink8. After we got the one dimension vector which saved the EAR of every frame in the Eyeblink8, we reshape the vector into a N*13 matrix using the N-6, N and N+6 frames method. Finally, we cascaded the 8 EAR matrix to make a 69649 * 13 matrix as the total dataset for SVM training and testing.

As well known, the normalization of the dataset is also important to the accuracy of the SVM. So, before training the SVM model, we shuffled the dataset and normalized the dataset to have a zero mean, unit variance in order to get a better training performance. After that, we separated the whole dataset into two parts, 80% of the dataset is used for training, and the remaining for testing.

Eventually we got the training dataset: shape of 62676×13 (has 3648 eye closed frames) and dataset for testing: shape of 6973×13 (has 383 eye closed frames).

Labels: 0 indicate the eye opened state, and 1 for eye closed.

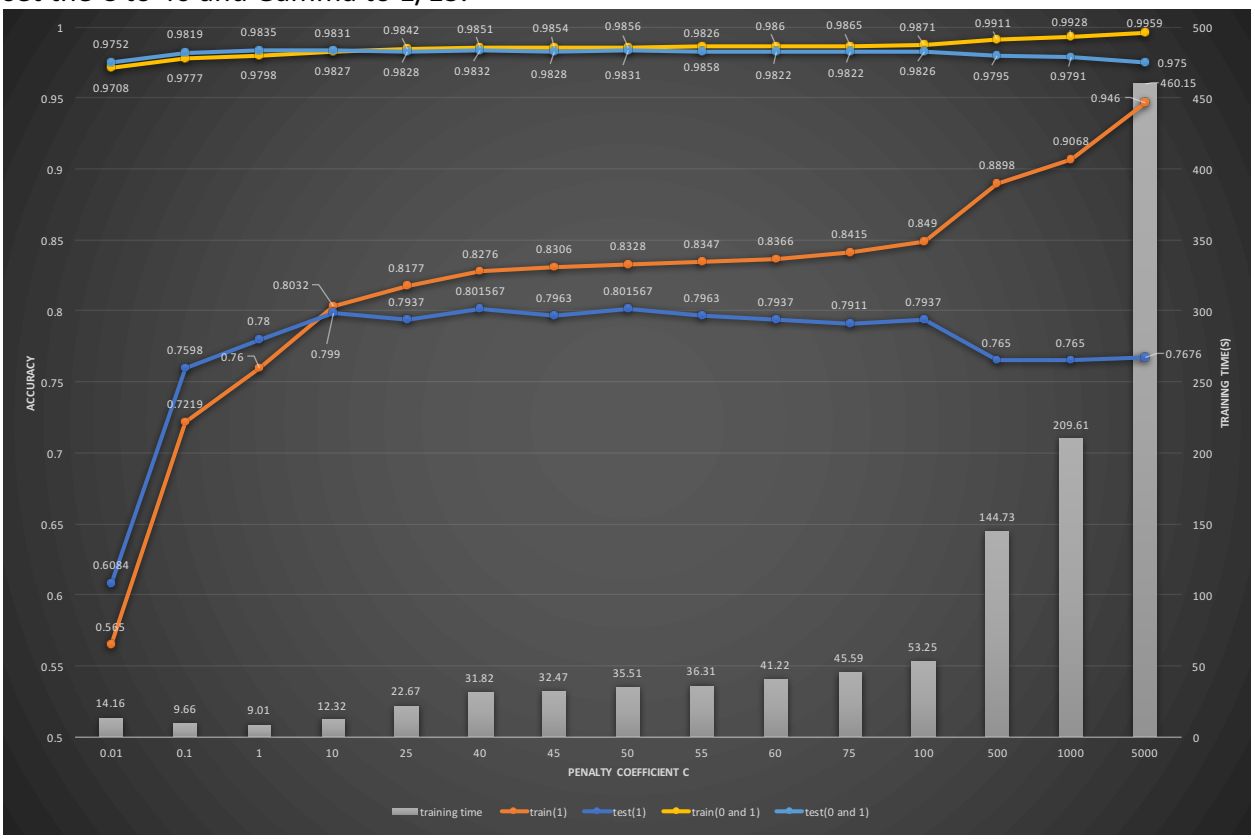
Tuning hyperparameters

Among all the hyper parameters of the SVM, C and Gamma is of the great importance. C is the penalty coefficient, that is, the tolerance to the fit error. The higher the C is, there is less tolerance to the error and harder to fit, and on the other side, if we set a smaller C, it is easier and quicker to train the set. What we cannot overlook is that setting a big C may lead to an excellent training accuracy, approximately 100 %, but a big C can also cause an overfit problem and will have a relatively poor performance in the testing dataset.

$Gamma$ implicitly determines the distribution of the data mapped to the new feature space. The greater the gamma is, the less the support vectors will be. And the number of support vectors affects the speed of training and prediction. In our experiment, we set G$amma$ as default equals to 1/dimension of the features =1/13.

From the chart 1, we can see that with an increasing C, the training time keeps increasing. The upper two lines indicate the training and testing accuracy for the dataset in the Eyeblink8. We need to notice that since usually people have an opened eye state, which labeled with 0. The whole training and testing dataset is a sparse matrix, so the total accuracy of training and

testing dataset is not so representative. For this reason, we introduce the 'accuracy for label 1', as orange and blue line showing in the graph, which represents the accuracy for predicting the label 1 in training and testing dataset, respectively. We can find that as the C increases, the training accuracy for 1 keeps growing, but the testing accuracy for 1 has its maximum when C is around 40 to 50, and after that the testing accuracy for 1 keeps decreasing which means the model already has the overfit problem. Therefore, after trying different hyper parameters, we set the C to 40 and Gamma to 1/13.



c	0.01	0.1	1	10	25	40	45	50
train(1)	0.565	0.7219	0.76	0.8032	0.8177	0.8276	0.8306	0.8328
test(1)	0.6084	0.7598	0.78	0.799	0.7937	0.801567	0.7963	0.801567
training time(s)	14.16	9.66	9.01	12.32	22.67	31.82	32.47	35.51
train accuracy for all(0 and 1)	0.9708	0.9777	0.9798	0.9827	0.9842	0.9851	0.9854	0.9856
test accuracy for all(0 and 1)	0.9752	0.9819	0.9835	0.9831	0.9828	0.9832	0.9828	0.9831
c	55	60	75	100	500	1000	5000	
train(1)	0.8347	0.8366	0.8415	0.849	0.8898	0.9068	0.946	
test(1)	0.7963	0.7937	0.7911	0.7937	0.765	0.765	0.7676	
training time(s)	36.31	41.22	45.59	53.25	144.73	209.61	460.15	
train accuracy for all(0 and 1)	0.9858	0.986	0.9865	0.9871	0.9911	0.9928	0.9959	
test accuracy for all(0 and 1)	0.9826	0.9822	0.9822	0.9826	0.9795	0.9791	0.975	

Chart 1 Performance of the EAR SVM

Result of experiment2

After combining the SVM with the original classifier, from the figure 6, we can see that the SVM predictor has an overall better performance than the predictor only with a fixed threshold. In the upper three graphs, we could conclude that in some scene, if we have a proper threshold, the both classifier could be able to detect the eye blinks well, in some scenes where people may have a quick unconscious eye blink, the classifier with a fixed threshold could have wrong classification, besides, in other scenes where people shake their head, make a face expression or talk to others, the EAR will have a fluctuation in a short time range, and the classifier with a fixed threshold could misclassify those small change in the EAR to an eye blink. As a comparison, the classifier with a SVM model always has a good performance and appears more robust to the normal tiny changes in the real-time EAR.

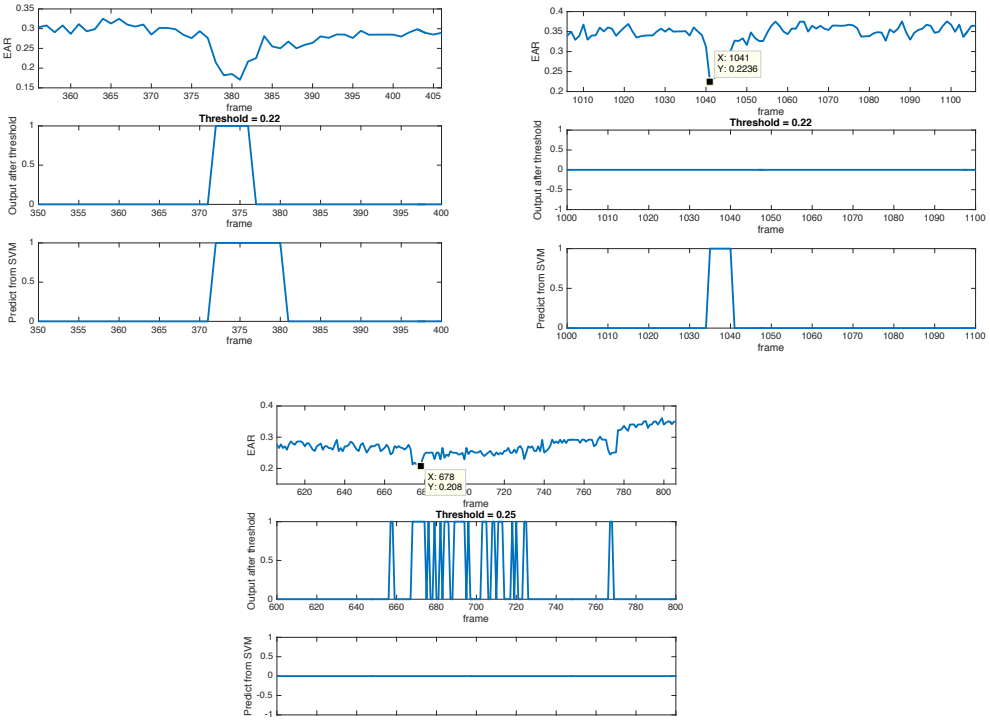


Figure 6 Comparison of classification results with/without SVM.

Conclusion

The method of predicting the eye blink using facial landmarks and SVM can have a good result to predict eye blink in real time with a generic camera, which has small computational costs and doesn't need special equipment.

To get an even better classification result, using a shallow Neural Network to substitute SVM is another choice since it can generate a more complex model to adapt to the 13-dimension features.

Besides, the inter-ocular distance can also have effect on the classification performance since if the distance is too far, the coordinates of the facial landmarks will get close to each other, which can be affect easily by regular movements of the tester's head.

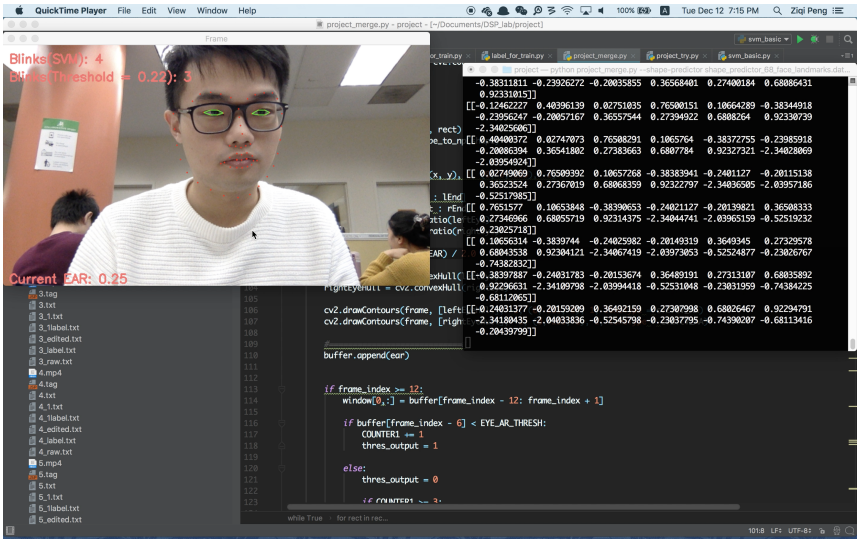


Figure 7 Screenshot of our experiment2

Reference

[1] Soukupova, T., & Cech, J. (2016, February). Real-time eye blink detection using facial landmarks. In 21st Computer Vision Winter Workshop (CVWW'2016) (pp. 1-8).

[2] Image adapted from https://www.learnopencv.com/facial-landmark-detection/

[3] Graph adapted from Eric Xing's slides.

[4] Pan, G., Sun, L., Wu, Z., & Lao, S. (2007). Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Webcamera. IEEE, International Conference on Computer Vision (pp.1-8). IEEE.

[5] Drutarovsky, T., & Fogelton, A. (2014, September). Eye Blink Detection Using Variance of Motion Vectors. In ECCV Workshops (3) (pp. 436-448).