EyeFace SDK v4.4.0

Technical Sheet

Contact:

Eyedea Recognition, s.r.o.

Office address:
Vysehradska 320/49
128 00, Prague 2
Czech Republic

web: http://www.eyedea.cz

email: info@eyedea.cz

1 About

This document aims to provide a thorough description of EyeFace SDK in a sense of the recommended setup and the quality of processing. The EyeFace SDK is based on the state-of-the-art machine learning and computer vision results. In the process of development, various decisions were made concerning processing speed versus quality of results. For EyeFace SDK, the goal was to achieve real-time performance with the best quality, however there exist scenarios where the primary goal is the best possible quality. We are always ready to present our best performing technologies to clients on request.

All effort has been put into making this document as correct and complete as possible.

1.1 Supported Platforms

EyeFace SDK is a standalone cross-platform x86/x64 C++ library with internal multi-threading. It currently provides the following to be integrated into the client's software solutions (the non-native APIs are bundled in a form of wrappers with source code included):

- C native API
- C# API
- Java JNI API
- Matlab/Octave MEX API

Eyedea Recognition, Ltd. focuses on cross-platform software development. The following operating systems and platforms are officially supported:

- Windows 7,8.1,10 32b/64b (Microsoft Visual Studio 2015)
- Ubuntu 16.04 64b
- CentOS 7 64b

Newer versions of the above systems and platforms should be compatible with EyeFace SDK. For information regarding additional interfaces, operating systems and/or platforms (e.g. ARM), please contact Eyedea Recognition, Ltd.

2 Licensing

The licensing is performed using gemalto's Sentinel LDK. The library is encrypted and has a real-time protection scheme. Every computer running a program which uses EyeFace SDK library must have a license. That means if a software integrator wants to develop software using EyeFace SDK, it is necessary to purchase a license for every computer where his program will be installed. See our store for complete list of prices.

The licensing schemes possible are

- Sentinel SL key
  - software license, delivered by email
  - machine tied with possibility to transfer to another computer
  - requires activation file to be transmitted to our company
  - licensing difficulties when changing hardware or reinstalling operating system
- Sentinel HL Time key
  - USB dongle key, delivered by post service
  - not machine tied

- can only be used on a machine which it is plugged into

- Sentinel HL NetTime key
  - USB dongle key, delivered by post service
  - not machine tied
  - creates a licensing server for all computer in the network with any number of licenses

Using these licensing scenarios, it is possible to either individually setup the license for every computer, or setup a licensing server with license pool which will be shared by multiple computers.

3 Recommended Hardware

The EyeFace SDK has moderate computational requirements on CPU resources when executed in typical setup on VGA resolution. It scans all possible position in the whole image on various scales to detect faces, describes the faces detected with multiple descriptors to obtain facial landmarks, age and gender and to verify identity. The requirements will vary with input image resolution, scene complexity, lighting complexity, number of people in the scene and many more.

Some scenarios might not require this amount of computational power and thus it might be sufficient to use low spec hardware like Intel Celeron or ARM. For example, if the detection and recognition only needs to be done on faces with known position and known image area on single image, not in video stream, EyeFace SDK might be configured to use this information to speed up the processing. For example, detecting a face in VGA resolution, when the face is known to has at least 240x240 pixels is much faster than searching for faces all the way from 24x24 pixels.

The clients should be advised that one of the bottlenecks is the size of the detection and recognition model files which need to be processed by CPU, in the order of hundreds of megabytes. This puts high stress mainly on a CPU with low amount of cache, like ARM devices. With 2014 ARM models, we were not able to achieve more than 1 FPS processing in standard setup.

3.1 Recommended Processing Speed

To obtain the best results possible, we recommend to setup the scenario and hardware in such a way, that at least 15 FPS processing is possible. We have seen good results also with processing at 10 FPS, but we consider this to be a hard limit and would not recommend using the software if the processing speed drops below 5 FPS.

3.2 Recommended CPUs

We have tested the following CPUs while processing VGA streams and for each of them we tagged whether it is recommended or not.

- Intel i5 3427U RECOMMENDED
- Intel i5 4570T RECOMMENDED
- Intel i7 4510U RECOMMENDED
- Intel i7 3770 RECOMMENDED
- Intel Celeron N2820 NOT RECOMMENDED

The benchmarks of the processing speed are shown in Table 1. All benchmarks were performed in the basic setup without any configuration to EyeFace SDK, the resolutions and number of cameras are specified.

	1 camera, 1 face	2 cameras, 1 face	1 camera, 4 faces	2 cameras, 4 faces
Intel i5 3427U	18	12	10	6
Intel i5 4570T	26	15	16	7
Intel i7 4510U	30	17	15	7
Intel i7 3770	30	25	22	20
Intel Celeron N2820	5	2	3	1

Table 1: Processing speed in frames per second on selected CPUs while processing VGA resolution video stream streaming at 30 frames per second, each containing the specified number of faces at once. Recommended setup labeled green, non-optimal setup labeled orange, not recommended labeled red.

4 Face Detector

Face detector is an algorithm which for an input image returns the positions where face candidates were localized, together with confidence of each candidate. EyeFace SDK is based on a state-of-the-art face detector which is a result of our more than ten years of experience in the field of detection. Our detectors beat the competitors not only by its detection performance, but also with processing speed. Our face detector detects both frontal and profile faces.

Basically, any detector, including the face detector, is bounded by two types of error. A type I error, called false negative, is a situation where the detector does not produce a face candidate for a face present in the image. A type II error, called false positive, is a situation where the detector does produce a face candidate in a location where no face is present in the image. A detector needs to make a compromise between these two contradictory errors. This is done by setting a threshold value on the candidate confidence. Then, face candidates with confidence above threshold are reported to the user.

The new face detector present in EyeFace SDK, when released in early 2015, placed 2nd in the challenging and well known FDDB benchmark, Illustration 1, and achieved the 1st place on MALF challenge, Illustration 2,which was part of FG 2015 conference in Ljubljana, Slovenia.

The detector performance may vary depending on the client's domain of use. To the best of our knowledge, the detector keeps high performance regardless of age, gender and ethnicity of person in the image, or whether the person wears (sun)glasses. The processing speed might be negatively influenced by difficult background, for example vegetation.

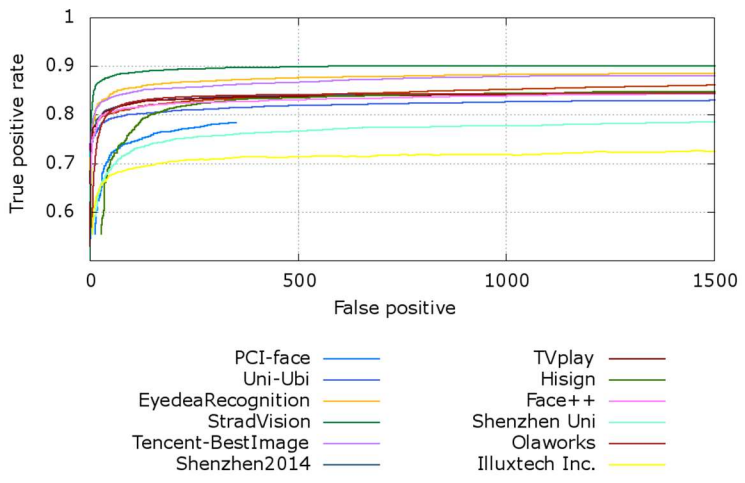

Illustration 1 : FDDB results, early 2015. [See http://vis-www.cs.umass.edu/fddb/results.html]

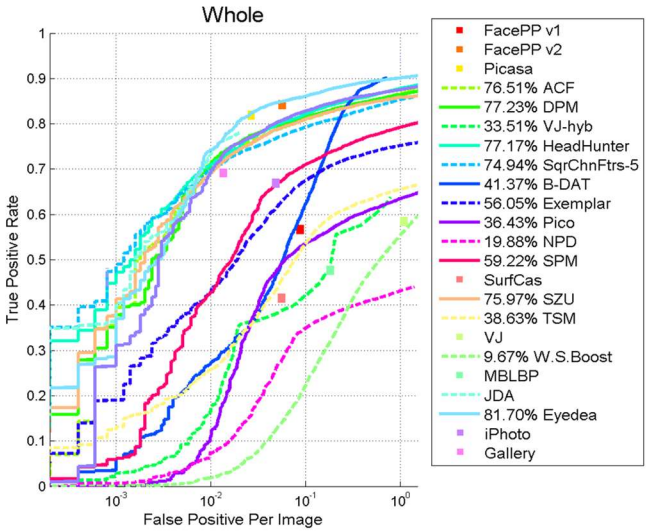

Illustration 2: MALF results, April 2015. [See http://www.cbsr.ia.ac.cn/faceevaluation/results.html]

5 Facial Landmarks

Facial landmarks, also called key points, are uniquely identifiable features of face, like mouths and eyes corners, nose and eyebrows. Facial landmarks are useful for precise localization of the face, necessary for augmented reality or recognition of age, gender and other facial statistics. The current version of our landmark detector can detect 20 facial landmarks. In Illustration 3 the precision of localization is given relative to the intra ocular distance, as measured on the famous face recognition LFW dataset (see http://vis-www.cs.umass.edu/lfw/). The facial landmarks are not enabled by default.

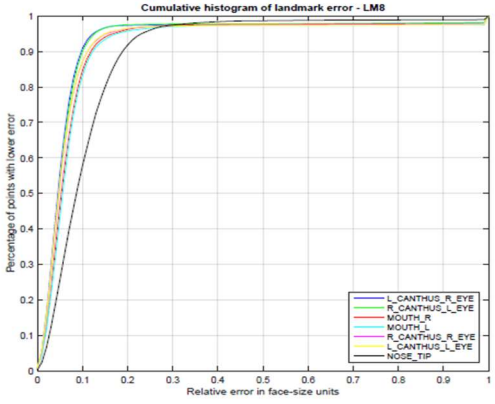

Illustration 3: Relative errors of localization of selected landmarks on LFW dataset.

6 Age and Gender Recognition

EyeFace SDK analyzes the input face images and recognizes age and gender statistics of the faces. The age and gender statistics are only available for frontal face images.

Because there currently exists no real-life age and gender recognition benchmark, we have prepared a real-life dataset containing millions of images of people from all age, gender and ethnicity groups. On this dataset, we have achieved 95% recognition performance in gender and mean absolute error of 5.24 years in age statistics, where mean absolute error is the average error of age measurement in absolute value. In our experiments, the mean absolute error is increasing when the audience is under 10 years or above 60 years old.

For the best accuracy when measuring age and gender, we recommend making the setup such that the intra ocular distance of the face is at least 60 pixels. Good performance is achieved when the intra ocular distance is higher than 30 pixels.

We have made all effort to make the age and gender recognition system invariant to audience statistics, however, the actual performance may vary when the statistics is biased, for example by strong correlation of age, gender or ethnicity.

7 Smart Tracking

The identity recognition subsystem in EyeFace SDK is used to join facial tracks in case of tracking failures. A facial track is a group of face images in consecutive frames, expected to be the images of the same person, which were joined by our tracker based on various statistics. When a facial track is lost, for example when the person disappears from the image for a while, we store a facial descriptor in the database, and wait for the person to return. If the person appears again in the input image, we merge the new facial track with the previous one of the same person using identity recognition. It is not possible to perform full scale face recognition using this library, please check our other products.

To make the identity recognition real-time, we do not use our state-of-the-art identity recognition techniques, which would consume more than a hundred of milliseconds CPU time per face image to produce descriptor, but a fast method with decreased precision. Performance wise, our face recognition method achieves 95% accuracy on the LFW dataset (see http://vis-www.cs.umass.edu/lfw/).