FACECHECK SDK

Developers Guide

Content description

This is a reference manual and configuration guide for the NeoCheck FaceCheck SDK product. It shows how to interact with the .Net library in order to manipulate supported capture devices, detect human faces, extract meaningful characteristics (QA) and match different samples in a fast and easy way.

Madrid, October 1st 2017

Disclaimer

The information contained within this document is and shall remain the property of NeoCheck. It must not be produced in whole or in part, or given or communicated to any third party without the prior consent of NeoCheck.

Whilst careful attention have been paid to ensure the appropriate referencing of information and sources, it is possible that a document of this nature might contain some errors. In such cases, upon notification NeoCheck will immediately analyse and make any required corrections or amendments.

Release Notes

Version	Date	Description
1.0	07/09/2017	Initial Version
1.1	01/10/2017	DirectShow camera optimization

Index

1 Introduction

This user manual describes the different functions and features provided by NeoCheck FaceCheck SDK. The purpose of this SDK is to provide functionality to manage photo cameras and perform facial recognition, matching and QA. The SDK is designed for simple integration of facial recognition into applications, which need biometric identity verification. The SDK has a high-level API that provides functionality for the facial recognition operations.

Features:

Camera Capture

Camera capture is a process when a camera connected to the computer is started and camera frames are extracted as images to be processed (facial recognition, matching and QA)

Face Detection

Face detection is a process when a subject can be detected by evaluating an image where the subject is present. The face detection process retrieves the location of the face with a high degree of confidence, even in cases when the face is rotated or the subject has a beard or is wearing glasses.

Face Matching

Face matching is a process when a subject can be identified by evaluating his biometric features and comparing them with another image to verify the individual is the person they claim to be. This process is called one-to-one matching because both images are from the same individual. Face matching is a fast way to compare images from the same subject, or a subject with several other subjects, which is called one-to-many matching.

Face QA

Face QA is the process to asses if a given subject image is compliant with the ICAO requirements checks. The different checks performed are:

- Good Vertical Face Position: Face has a good vertical face position
- Horizontally Center Face: Face is horizontally centered
- Good Exposure: Exposure of facial image is appropriate
- Eyes Not Red: Eyes are not red (no red eyes)
- Resolution: Image resolution is appropriate.
- Only One Face Visible: Only one face visible
- Is Frontal: Photo is frontal. Face image is frontally captured.
- Overall Photo Quality: This indicates the final score of the taken image if it is green the photo can be used for the face verification process meeting the min. quality requirements.

2 Installation

There are a set of prerequisites before start using the Face SDK:

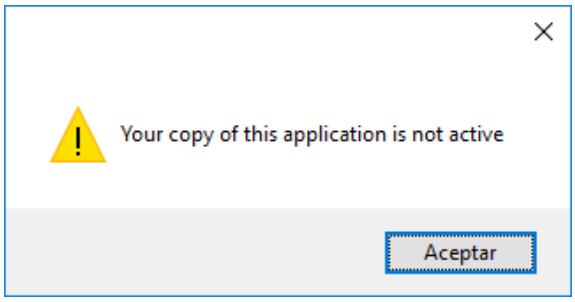
- Windows XP/7/10
- .NET framework 4.6

The Face SDK comes as a set of DLL libraries that are distributed together, along with a Demo application and C# sample code.

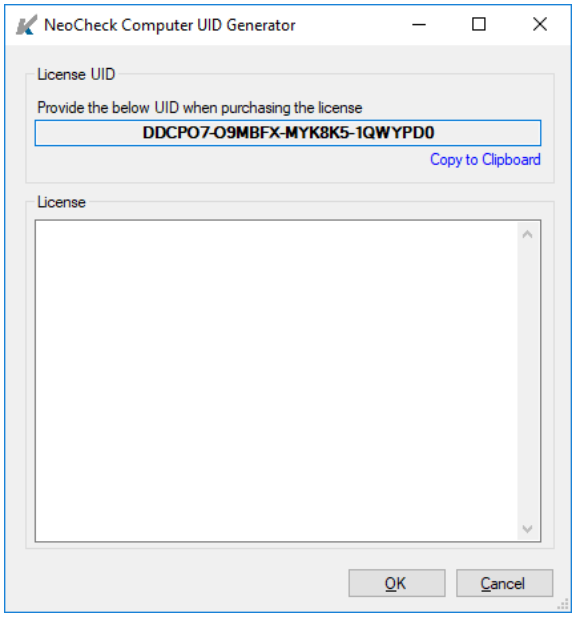
2.1 FaceCheck SDK Licensing

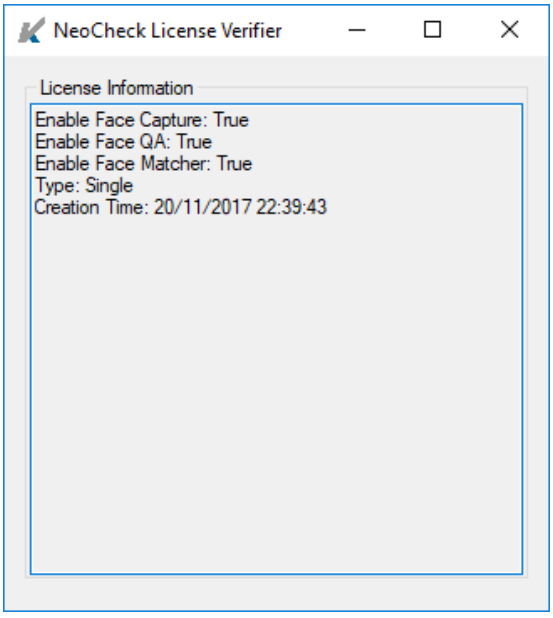
The Face SDK uses a third-party software provided to perform facial 1:1 matching and extended QA. The software requires to obtain a license on this component.

In order to activate the SDK, it is necessary to execute the NeoCheckLicenseVerifier tool. The following message will be displayed until a valid NeoCheckLicense.lic file is found:



A new window will be loaded showing your machine UID which should be sent to licenses@neocheck.com for generating the activation code. Once you paste the provided code in the lower textbox, a .lic file is going to be created in the execution route.





3 .NET Reference

NeoCheck Face SDK uses a set of components for working with cameras connected to the computer. These perform facial recognition, matching and QA.

3.1 Classes

NAME	DESCRIPTION
FaceSDK	The main point of entry of the Face SDK. Provides methods and events for working with cameras and performing facial recognition, matching and QA. See
GenericCamera	Represents the base class to operate with cameras connected to the computer, providing methods to select, start and stop a camera.
DirectShowCamera	Camera implementation, using the Microsoft DirectShow API.
NeuroTechnologyCamera	Camera implementation, using 3rd party module.
IFaceMatcher	Represents the base class to perform facial recognition, matching and QA.
NeuroTechnologyFaceMatcher	Face matcher implementation, using 3rd party module.

3.2 Events

NAME	DESCRIPTION
FrameReceived	Event raised by a camera implementation when a new frame is retrieved from the camera. It includes the frame bitmap object in the event argument.
LiveQAFrameReceived	Event raised by the face matcher implementation when performing a LiveQA operation. It includes a complex object in the event argument that includes the frame bitmap, the face rectangle, the most representative face points and the QA checks performed.

3.3 Enums

NAME	DESCRIPTION
CameraImplementation	The camera implementations available to interact with cameras connected to the computer. Values: Virtual DirectShow NeuroTechnology
FaceMatcherImplementation	The face matcher implementations available to perform facial recognition, matching and QA. Values: NeuroTechnology
FaceQACheckResult	The face matcher result for every QA check performed during a face QA operation. Values: NotSet, Ok Warning Error

FaceQAResultStatus	The face matcher global result after a face QA operation. Values: NotStarted Processing Finished FaceNotFound Canceled
IcaoWarning	The type of all the ICAO checks that are performed during a face QA operation. Values: FaceNotDetected RollLeft RollRight YawLeft YawRight PitchUp PitchDown TooNear TooFar TooNorth TooSouth TooEast TooWest Sharpness BackgroundUniformity GrayscaleDensity Saturation Expression DarkGlasses Blink MouthOpen LookingAway RedEye FaceDarkness UnnaturalSkinTone WashedOut Pixelation SkinReflection GlassesReflection

3.4 EventArgs

NAME	DESCRIPTION
FrameReceivedEventArgs	Event raised by the camera implementation when a new frame is available from the camera. It contains the frame Bitmap object.
LiveQAFrameReceivedEventArgs	Event raised by the face matcher implementation when performing a live QA operation. It contains a complex object with image captured and QA information. See FaceQAResult in Chapter 1.5.

3.5 Data Types

NAME	DESCRIPTION
FaceFeaturePoint	Contains information about a representative point extracted during facial recognition
FaceQAResult	Contains all the information about a face QA operation, including: - Face rectangle shape (Rectangle) - Representative face points (List<FaceFeaturePoint>) - ICAO Checks (Dictionary<IcaoWarning, FaceQACheckResult>) - Status (FaceQAResultStatus)

3.6 Methods

Following a list of all public methods from Face SDK, cameras and face matcher implementations

3.6.1 FaceSDK.GetInstance method

Returns a unique instance (Singleton) of the Face SDK.

PARAMETER	DESCRIPTION
CameraImplementation	The camera implementation used to interact with cameras connected to the computer.
FaceMatcherImplementation	The face matcher implementation used perform facial recognition, matching and QA.

3.6.2 FaceSDK.GetCameraDevices method

Returns a list of all camera devices connected to the computer.

3.6.3 FaceSDK.StartCamera method

Starts a camera connected to the computer. The camera can be selected by passing the camera name as parameter.

PARAMETER	DESCRIPTION
string cameraName	The name of the camera to start. To obtain the camera name use the method GetCameraDevices. If parameter is null, the first camera detected will be started.

3.6.4 FaceSDK.GetCameraObject method

Returns the camera COM object. The camera can be selected by passing the camera name as parameter.

PARAMETER	DESCRIPTION
string cameraName	The name of the camera to retrieve. To obtain the camera name use the method GetCameraDevices. If parameter is null, the first camera detected will be retrieved.

3.6.5 FaceSDK.FindFace method

Returns the coordinates of the face present in an image that is passed as parameter. If face is not detected, returns a null value.

PARAMETER	DESCRIPTION
Bitmap image	The image to perform the face detection.

3.6.6 FaceSDK.MatchFaces method

Returns a value indicating the similarity of the faces detected in the images passed as parameters. This value is a `float` between 0 and 1.

PARAMETER	DESCRIPTION
Bitmap image1	The first image to perform the face matching.
Bitmap image2	The second image to perform the face matching.

3.6.7 FaceSDK.FaceQA method

Returns a complex object with information about the quality of the face present in an image that is passed as parameter. This information includes:

- Coordinates of the face detected
- Representative points of the face detected (mouth, eyes, nose tip)
- List of ICAO standard checks performed over the face detected, and its result (Ok, Failed, Warning)
- Global result status (Process finished, Face not found, Process canceled)

PARAMETER	DESCRIPTION
Bitmap image	The image to perform the face QA.

3.6.8 FaceSDK.FaceLiveQA method

Asynchronous task that allows the client application to start a camera and perform a Live QA process over the live frame images captured by the camera. This task ends when a face is detected from the live frame captured by the camera and all ICAO checks are OK. During the process, client applications can be informed about the instant quality of the captured face by subscribing to the event `LiveQAFrameReceived`, described in Chapter 1.2.

PARAMETER	DESCRIPTION
string cameraName	The name of the camera to use for LiveQA operation. If parameter is null, the first camera available will be taken.

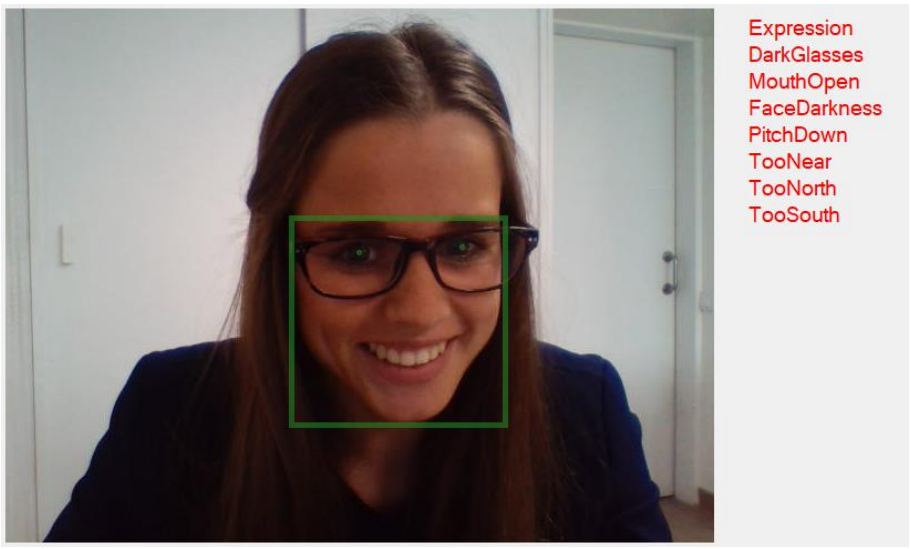


Figure 1 - Live failed checks detected

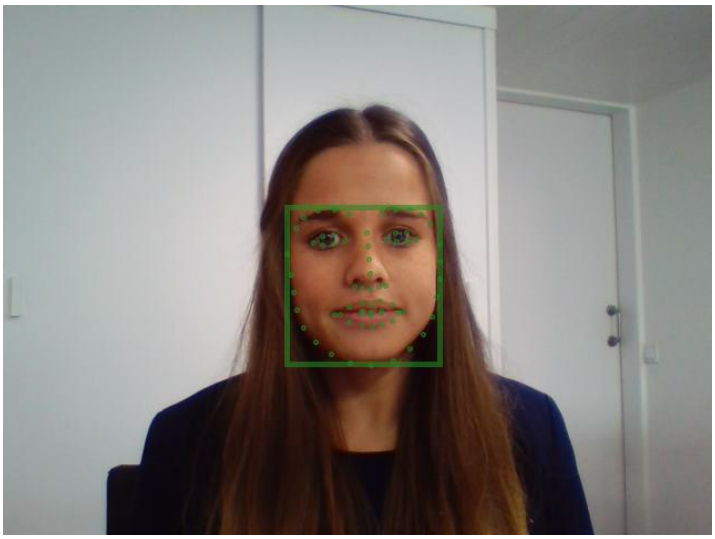


Figure 2 - Image acquired