

# Face Recognition System



[stendhal.syndrome.studio@gmail.com](mailto:stendhal.syndrome.studio@gmail.com)

## Contents

1. About.....	3
2. Quick start.....	4
3. API overview.....	4
4. Contact us.....	6

## 1. About

Face Recognition System is a plugin that provides high-quality face recognition offline.

### Features:

- **Face recognition** solution, including face detection (using 6 keypoints: left eye, right eye, nose, mouth, left ear, right ear), face alignment, face normalization, face feature extraction and face search;
- It recognizes faces quite accurately (up to **99.5%** on LFW dataset);
- It works **offline**, in **real time**;
- It includes a demo scene with an example of face search and recognition on a photo, a video, a video from a web-camera;
- Simple interface, low learning cost, can be applied to actual projects immediately, especially suitable for face attendance or face login authentication.

### Supported platforms:


- Windows 10, Windows 7 Service Pack 1;
- Android (armeabi-v7a or arm64-v8a).

### Links:

[Video instruction on plug-in usage](#)

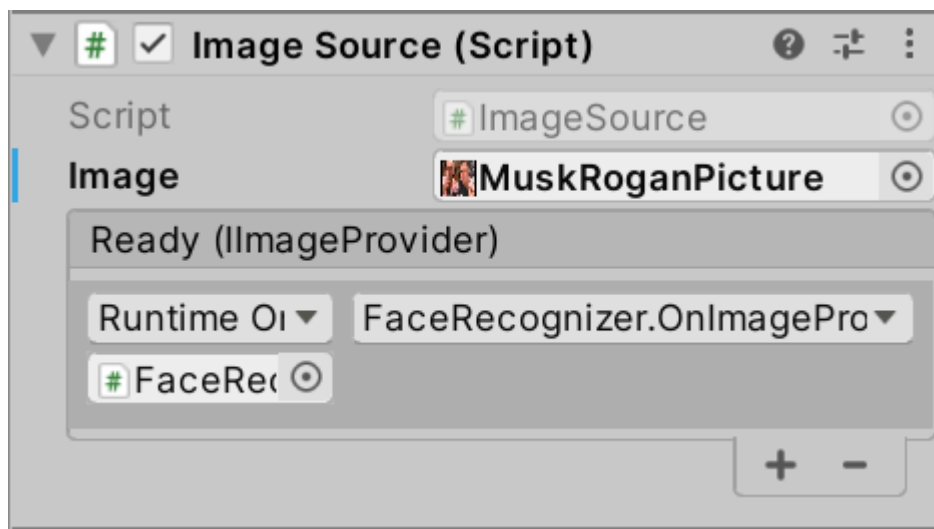
[Free version of plugin](#)

## 2. Quick start

- Import package **FaceRecognitionSystem.unitypackage**.
- Load the scene **SampleScene** from folder **Demo**;
- Activate object ImageSrs for search and recognition of faces on the photo, activate object VideoSrc for search and recognition of faces on the video, activate object WebCamSrc for search and recognition of faces on the video from a Web-camera;
- Press button  to load the scene.

## 3. API overview

### 3.1 ImageSource ( implements IImageProvider )



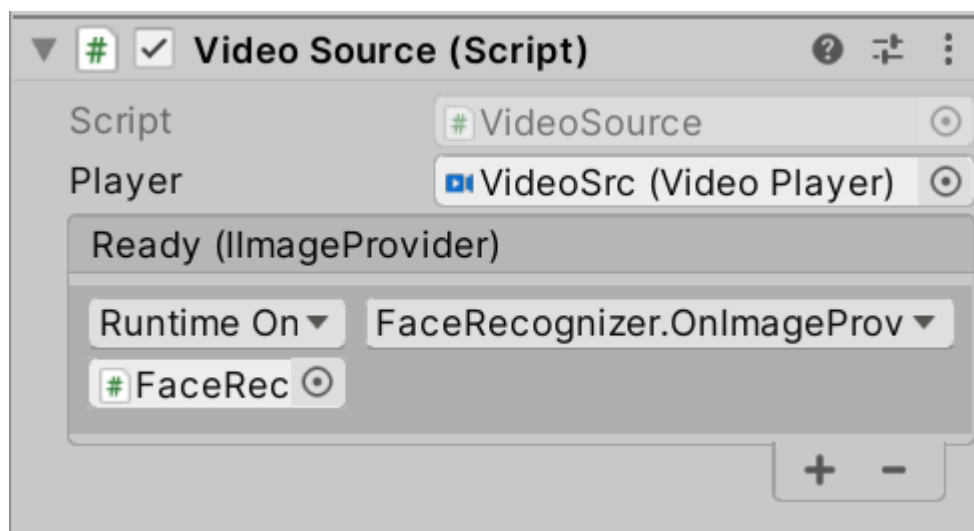
#### Fields of class:

- **Image** - image (Sprite (2D and UI)) on which it is required to find and recognize faces.

#### Events of class:

- **Ready (IImageProvider)** – event (UnityEvent<IImageProvider>) with results of initialization of the source of the image.

### 3.2 VideoSource ( implements IImageProvider )



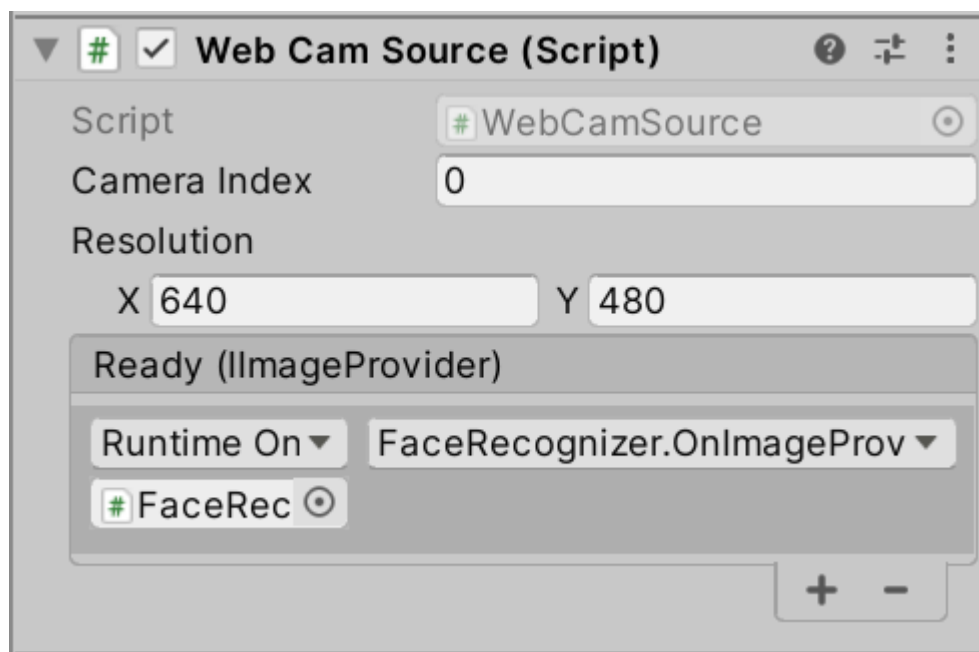
#### Fields of class:

- **Player** – link to the component Unity Video Player with the video on which it is required to find and recognize a face.

#### Events of class:

- **Ready (IImageProvider)** – event (UnityEvent<IImageProvider>) with results of initialization of the source of the image.

### 3.3 WebCamSource ( implements IImageProvider )



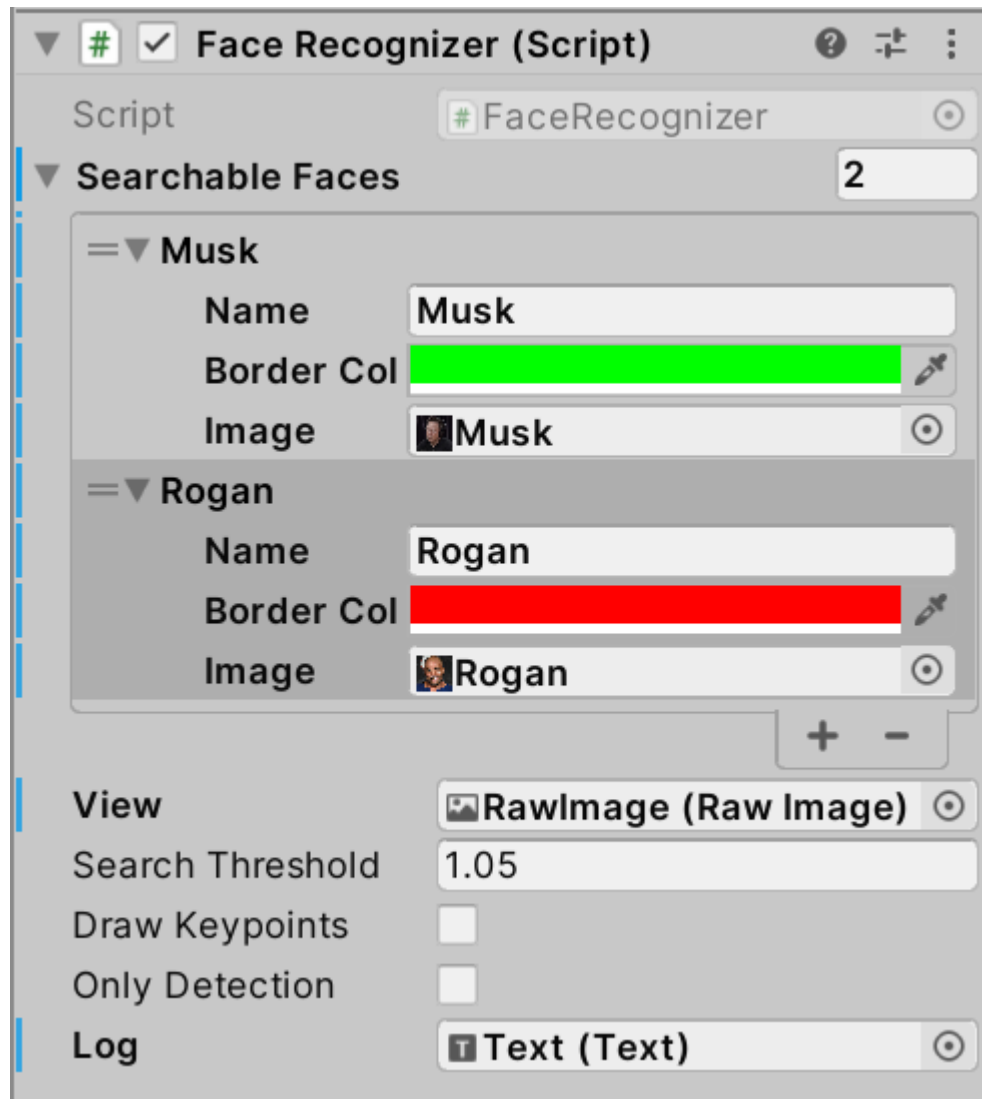
#### Fields of class:

- **CameraIndex** – index number of the web-camera in the operating system;
- **Resolution** – resolution of the web-camera.

#### Events of cladd:

- **Ready (IImageProvider)** – event (UnityEvent<IImageProvider>) with results of initialization of the source of the image.

### 3.4 FaceRecognizer



Fields of class:

- **SearchableFaces (List<Face>)** – list of faces that need to be found (identified) on the photo/video;
- **FoundFaces (List<FaceDescription>)** – list of the faces found on the photo/video;
- **View** – texture on which the result of recognition will be depicted;
- **SearchThreshold** – threshold defining the level of recognition of the face by the neural network. The less is the value, the less is the quantity of false-positive results and the more is the quantity of false-negative results;
- **DrawKeypoints** – defines the necessity of drawing the key points of the face (eyes, nose, mouth, ears) on View;

### Face Recognition System

- **OnlyDetection** – sets the mode in which search of faces without their recognition is performed. Setting of this mode increases the speed of plugin's work;
- **Log** – the object of `Unity.UI.Text` for depicting of debug messages.

#### Methods of class:

- **OnImageProviderReady( IImageProvider )** - event handler **Ready** of the component **ImageSource\VideoSource\WebCamSource** of the object **ImageSrc\VideoSrc\WebCamSrc**. The method receives the object implementing the interface **IImageProvider** and saves the link to it for the further access to the image (`Texture2D`).

### 3.5 FaceDescription

#### Fields of class:

- **Name** – face identifier (name of the person);
- **DescriptionVector** - face features;
- **RightEye** - coordinates of the right eye;
- **LeftEye** - coordinates of the left eye;
- **Nose** – coordinates of the nose;
- **Mouth** - coordinates of the mouth;
- **RightEar** - coordinates of the right ear;
- **LeftEar** – coordinates of the left ear;
- **Rect** – rectangular defining the borders of the face.

#### Methods of class:

- **ComputeDistance** – static function defining extent of similarity of two faces. Accepts two parameters **DescriptionVector**. If the result is less than **SearchThreshold**, then it returns the result that the two faces belong to one and the same person.



#### **4. Contact us**

Do you meet issues while using this plugin?

Do you have suggestions how to improve API?

Feel free to contact us: [stendhal.syndrome.studio@gmail.com](mailto:stendhal.syndrome.studio@gmail.com)