
AutoSDK

Software development kit for license plate recognition developed by VIT

We provide AutoSDK as a set of C/C++ runtime libraries, header files, debugging facilities and other utilities. Together they allow you to create an LPR module for your application without having to write your own recognition engine from scratch. By using AutoSDK, you get fast detection of license plates with Latin and Cyrillic characters and high processing efficiency.

Learn more about AutoSDK:

- [Applications](#)
- [Key features](#)
- [Operation modes](#)
- [Technical specifications](#)
- [Licensing parameters](#)
- [About developer](#)

Applications

Currently we use AutoSDK as a recognition core of our own traffic safety products: AutoCode series, Overseer series, EDGE¹. The engine is also used in several other software products that can't be cited due to nondisclosure agreements (NDAs).

A variety of third-party products that may need AutoSDK as a component:

- traffic surveillance systems;
- parking management systems;
- perimeter protection & access control systems;
- electronic toll collection systems;
- intelligent transportation systems;
- safe city solutions;
- embedded devices, speed radars, cameras;
- weighing systems.

¹ Check out the product specifications on <http://vit.community>.

Key features

License plate recognition

- Recognition of license plates of up to **50 countries**². For each country, we have developed a specific recognition algorithm (aka country module). From release to release, we never stop advancing them.
- Recognition of single- and double-line license plates.
- Recognition of up to 20 plates in the same frame, which allows **multi-lane surveillance**.
- Single frame processing takes 10-100 ms (depending on resolution), which facilitates **real-time traffic control**.
- Maximum vehicle speed for its license plate to be recognized is 250 km/h (under proper camera settings³).
- Determining the direction of vehicle movement.
- Flexible recognition fine tuning: you can select specific license plate templates, create multiple zones within the frame to detect license plates in, as well as control the results accuracy. Meanwhile, the license plate tracking algorithms will automatically select the most valid recognition results from a series of candidates and will filter out the duplicates.

Due to efficient inner architecture, AutoSDK has no scaling restrictions: you can split the processing of your video stream to multiple threads in accordance with your hardware concurrency. Distribution of image processing load between multiple CPU cores will allow you to reach high operational speed while analyzing high-definition video with a decent framerate.

Please note that AutoSDK is licensed by the number of recognition threads available for creation, not by the number of channels (video streams to be processed). See details in [Licensing parameters](#).

Portability

Both Windows and Unix OS families are supported (see the [Technical specifications](#) section).

Integration of new license plate templates

There is a possibility of on-demand addition of license plate templates (aka license plate layouts) that are not recognized by any of AutoSDK's country modules yet. For additional information, see [Quality dependencies](#).

Operation modes

To allow our customers purchase the functionality they will definitely utilize, we provide AutoSDK operating in either of two modes:

AutoSDK Lite ("Parking" mode). Processes up to 6 frames of the overall number of frames obtained each second. With the appropriate video quality, this processing speed is enough for identifying vehicles moving at ~20 km/h. Thus, it is reasonable to use AutoSDK Lite at access points, parking lots, residential & commercial areas, gas stations, repair shops, logistics centers, warehouses, airports, etc.

AutoSDK ("Freeflow" mode). With sufficient CPU power, AutoSDK processes the whole frameset obtained each second. This provides for identification of fast-moving vehicles when monitoring the highways.

² The actual list of countries supported by AutoSDK can be found on our wiki ("AutoSDK: supported countries" article).

³ We have gathered our experience in a manual which we strongly recommend our customers to get acquainted with, "License Plate Recognition Cameras Selection, Setup and Installation Guide". You can find it on our wiki.

Technical specifications

Quality dependencies

Accuracy of recognition results depends on:

- Quality of test materials (video records, license plate images) provided by the client⁴. We use those test materials for training of corresponding AutoSDK's neural networks. Higher quality and bigger amount of test materials contribute to higher recognition accuracy at final.
- Equally lit, contrasty, fully visible license plate images on the video supplied for recognition. During all the stages of video scene preparation, the main emphasis should be put on meeting the requirements from the "License plate recognition cameras selection, setup and installation" guide (available on our wiki).

System requirements

Programming languages: C, C++.

OS: Unix family (e.g. Mac OS X, Linux, FreeBSD) and Microsoft Windows family (e.g. 2000, 2003, XP, Vista, Seven, 8/8.1, 10).

CPU architectures: x86 (x86-32, x86-64), ARMv7, ARMv8 AArch32.

CPU model: selected individually (according to target number of traffic lanes under monitoring, video framerate and resolution). When adding one or more recognition threads (1 monitored lane = 1 thread, see [Key features](#)), CPU load increases linearly.

The table below contains number of frames per second that can be processed by AutoSDK under specific computer configuration, processor load and image resolution. This data is a result of performance testing under the following conditions:

- the video stream is transferred with minimum level of compression (Xvid or MJPEG formats are used) meaning that compression workload is not included into calculations;
- the CPU is loaded with recognition only.

Default configurations with CPUs of various price categories and performance rates (from lowest to highest) were selected for testing.

In real-world conditions, a portion of server resources may be used for additional tasks, such as video archive recording, interaction with the database, anti-virus protection, etc. With this in mind, it is recommended to pick a CPU that, with AutoSDK actively running, would have about 60% workload.

Comparing the operating rate of AutoSDK under 60% CPU workload (fps)

Computer configuration	Image resolution			
	720 × 576	1360 × 512	1360 × 896	2336 × 1753
Intel Core i5-2500 3.3Ghz 2Gb RAM	96	51	28	7
Intel Xeon DP Quad-Core E5620 8Gb RAM	135	70	40	11
Intel Core i7-2600 3.4Ghz 8Gb RAM	165	76	49	15
Intel Core i7-3930K 3.2Ghz 16Gb RAM	277	144	82	24
2* Intel Xeon X5650 2.66Ghz 12Gb RAM	426	224	135	40

⁴ We have gathered our requirements for test video in "Video material requirements for the addition of new license plate types" document. It is available on our wiki.

RAM: 2Gb and more (processing data of 1 thread (1 traffic lane) uses approximately up to 250 Mb).

Minimal height of the symbols on the license plate: 14-20 px for cameras with no hardware compression (analog, machine vision cameras); 20-30 px for cameras with hardware compression (IP-cameras).

Optimal recognition camera angles: vertical — 18-20 (maximum — 30) degrees; horizontal — 5-10 (maximum — 20) degrees.

Angle of a license plate on the image: up to 5-10 degrees.

Input data: uncompressed video stream (RAW, 8 bpp, grayscale).

Output data

Each recognition result is represented by the following meta data:

- alphanumeric (Unicode) string containing the recognized license number;
- country code (according to ISO 3166 numeric);
- reliability rate of obtained recognition result (aka validity);
- direction of vehicle movement;
- timestamps:
 - of obtaining the best recognition result of the license plate,
 - of license plate's appearance in the frame (first recognition result),
 - of license plate's last recognition result (before vehicle has left the frame);
- whether the license plate is inversed or not;
- license plate background color;
- license plate symbols color;
- other specific data, which may be useful for developers (license plate angle/coordinates on the frame, identifier of the corresponding license plate template, etc.).

Product structure

AutoSDK distribution package contains:

- PDF documentation for developers (SDK reference, change log, test video requirements in English and Russian);
- header files;
- object files;
- dynamic (support, debug, release) libraries;
- helpful command-line utilities;
- usage examples;
- redistributable software (e.g. environment for license keys);
- license key.

Licensing parameters

Protected parameters of AutoSDK:

- **Operation modes:** AutoSDK Lite ("Parking") or AutoSDK ("Freeflow").
- Maximum number of recognition threads available (see [Key features](#)).
- Countries which license plates are to be recognized on the video.

License protection of AutoSDK is carried out using Sentinel LDK and Sentinel HASP solutions by Gemalto⁵. Available license key types:

HASP hardware key. Provided as a USB-token. This key must be physically installed onto a server with the protected software to be used. The key may be transferred to another server without losing the license.

HASP software key. Provided as a software (service) which attaches to the hardware of the recognition server. If protected product is to be transferred to another server, a new license should be purchased.

About developer

Video Internet Technologies Ltd. started its activity on video analytics market in 2005 with its head office in Kyiv, Ukraine. Our road safety products (AutoSDK, AutoCode, Overseer, OutdoorBox, EDGE) operate as standalone solutions and components in more than 1000+ installations worldwide. We have technological partnerships with world's leading vendors of surveillance equipment and video management software.

Find more at <http://vit.community>.

We provide full national and international technical support:

- | | | |
|--|---|---|
| • unlimited number of cases | gration with third party systems/devices | critical patches provision |
| • consultation on software functionality, installation and configuration | • identification and elimination of software malfunctions | • online support: phone consultation, chat, remote desktop connection |
| • consultation on products inte- | • updates to new versions and | • offline support: email |

Copyright ©2016 Video Internet Technologies Ltd. Technical specifications are subject to change without prior notice. All data is for information purposes only and not guaranteed for legal purposes. Information has been carefully checked before publication and is believed to be accurate. However, no responsibility is assumed for inaccuracies.

⁵ You may find the details on VIT licensing on our wiki.