LPM SDK

Technical sheet

Version 7.x

ADVANCED COMPUTER VISION SOLUTIONS

Contact:

Address:
Eyedea Recognition, s.r.o.
Vyšehradská 320/49
128 00, Prague 2
Czech Republic

web: http://www.eyedea.cz

email: info@eyedea.cz

Table of Contents

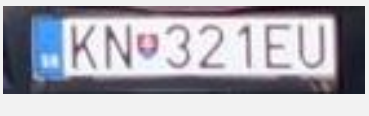
1 Product Description

The LPM SDK is a cross-platform software library designed to provide a comfortable detection of car license plates, ADR and Trash plates, and/or cars via bounding boxes, as well as optical character recognition (OCR) of plates including plate type and physical size recognition from input images. It defines an interface between the client's software and our state-of-the-art detection and recognition modules. This special API allows simple module administrations and updates without any need for changes to the client's software.

Each client receives an FTP account automatically created at Eyedea Recognition's server. This FTP access serves as two-way communication between the client and Eyedea Recognition, s. r. o.. Clients have an easy way to regularly upload data samples (or problematic data) to the FTP server, and subsequently receive the corresponding updates of LPM modules. This systematic approach makes it possible to verify result statistics and continuously adapt the LPM modules to the client's specific data, ensuring the best possible performance.

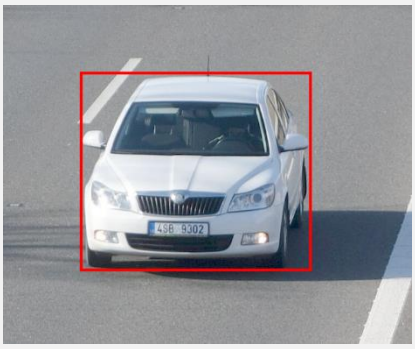
1.1 Technical Details

LPM SDK consists of two parts – base LPM engine and detection/recognition modules. Both are cross-platform libraries with C interface. The base LPM library is the only entry point, the user never uses the detection/recognition LPM modules directly. The module is loaded, configured, and executed using the LPM library. Each module can contain a detection routine, a OCR routine, or both.



License plate

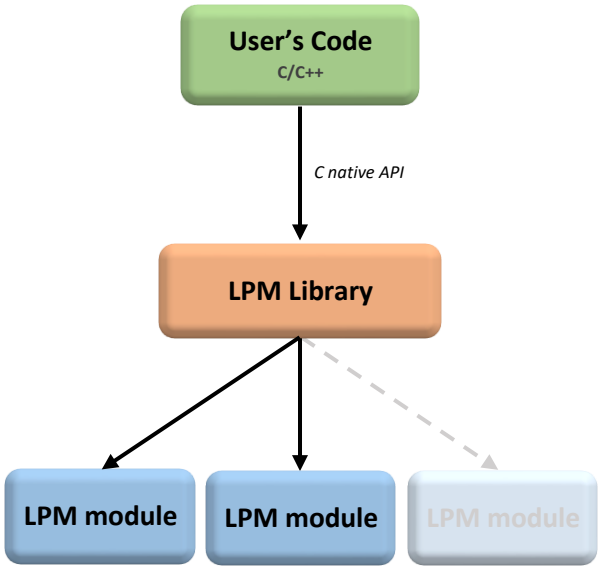
Multi-line license plates



Car bounding box

ADR plate

Trash plate



The LPM library provides the following APIs:

- C native API
- Python wrapper
- Java wrapper

Officially supported operating systems and platforms:

- Windows 7, 8, 8.1 and 10
  - 32-bit and 64-bit (Visual Studio 2019)
- Ubuntu 18.04 and higher
  - 64-bit
- Other platforms on request

1.2 System Workflow

The workflow of the LPM system consists of: image acquisition, plates or bounding boxes detection, and OCR of detected plates (where applicable). The image acquisition is not part of this SDK and must be solved separately.

The process starts with detection of license plates or ADR/Trash plates or car bounding boxes. Some types of detections can then be supplied to the OCR stage which returns hypotheses of the plate text and plate type, together with their confidences. There is no need to crop the detected plates for the OCR stage, as the OCR stage takes the whole input image and the detection results.

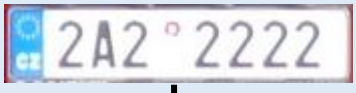
1) Image acquisition

Not part of the SDK

For every single input image:

2) License plate detection

Detection of plates positions in the input image



3) License plate OCR

License plate text and type recognition for detected plates

TEXT: 2A22222
TYPE: CZ
SIZE: 520x110mm

2 Distribution Contents

The following list is an excerpt from the LPM SDK directory structure, highlighting the most important directories and files contained in the software distribution. A brief description of the items is provided.

- [LPM SDK] ..... *distribution main folder*
  - LPM ..... *LPM engine folder*
    - include ..... *LPM header files*
    - lib ..... *LPM libraries*
  - examples ..... *LPM examples folder*
    - example-*anpr-implink* .....*example of implicit LPM library link*
      - images ..... *example images folder*
      - example-anpr-implink.vcxproj ..... *Visual Studio project (only Windows version)*
      - example.cpp ..... *example source code*
      - Makefile ..... *example makefile (only Linux version)*
  - modules-v7 ..... *LPM modules*
    - x64 ..... *modules for appropriate architecture*
    - config_camera_view.ini ..... *camera view parameters file*
  - hasp ..... *license management software folder*
  - documentation ..... *SDK documentation folder*
  - wrappers ..... *SDK wrappers folder*
  - LICENSE.txt ..... *SDK license*
  - WhatsNew.txt ..... *file with release notes for each SDK version*
  - README.txt ..... *SDK readme file*

3 Input Requirements

The input requirements of LPM are divided into two parts: image data, and scene specification. The first part, which is covered in the *Input Image* section, specifies the image capture criteria and the way to represent the image data. The second part, which is covered in the *Scale in Pixels per Meter* and *Image Rotation* sections, describes how to set the input parameters of the LPM SDK.

3.1 Input image

The result of OCR is dependent on two factors: the way the image was taken, and the way the image is stored. Image capture requirements are specified in the *Scene criteria* section.

3.1.1 Scene Criteria

In order to get the highest possible recognition accuracy, several rules must be respected during the vehicle input image data collection. The criteria that specify the camera scene settings are described in the *LP Alignment* and *Image Borders* paragraphs. The camera capture quality criteria are described in the Blurred image and Aspect ratio paragraphs.

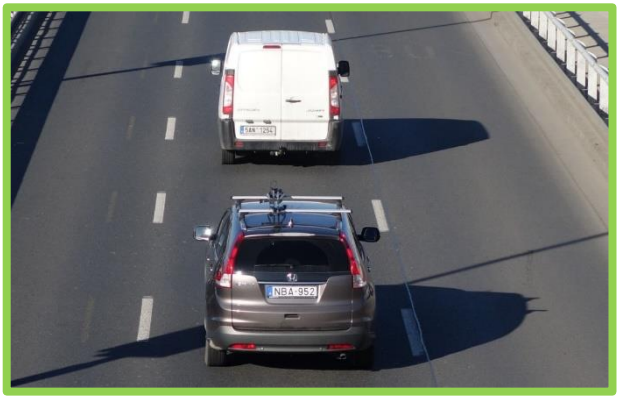
LP Alignment

By default, the module is set with the assumption that the plate in the input image is horizontally aligned. The exception to this rule is module 040, which supports up to 45° rotation in both directions by default.

Accepted rotation range in other modules is 5°. Aligning can be achieved on the user's side. The other option is to configure LPM detector, which can be set to run in a specified range of rotations (see Developers Guide).

WRONG:

The plate is rotated more than 5°.



CORRECT:

The plate is horizontally aligned.

Image Borders

The plate should be sufficiently distant from image borders, and should not be cropped or occluded, so that the whole text is visible. The surrounding red rectangle in the illustration should not go out of the image and should be fully visible to achieve the best recognition accuracy.

WRONG:

The plates are partially outside of the image.

CORRECT:

The whole plates with a sufficient surrounding area are located inside the image.

Blurred Image

The plate in the input image must not be blurred by motion or due to wrong camera settings. The whole text on the plate must be clearly visible and readable for successful recognition.

Resolution

Resolution of the input images should be at least 120 pixels per meter, i.e. the minimal EU license plate's width should be approximately 60 pixels. For more information about the pixels per meter units, see the chapter *Pixels per Meter*.

Aspect Ratio

Pixel aspect ratio of the image can be adjusted in the main configuration file (see Developers Guide).

3.2 Pixels per Meter

In the main LPM configuration, the detection size range is configured in pixels per meter units – how many pixels in the image represent one meter in the real world.

3.2.1 Description

Pixels per meter unit is used in the LPM SDK to define the resolution of the input image with respect to the dimensions of the observed object in the real world.

The detector uses a pixel per meter range that is provided by the user. The physical dimensions of the license plate and its parts are defined by law in many countries, but it is often the case that the same size format is used throughout a region, even if not enforced by law (e.g. 520 x 110 in Europe).

The formula below defines the relation between the size of the object (e.g. plate) in the image in pixels and in real world where the result is the *px_per_m*.

3.2.2 Formula

$$px\_per\_m = \frac{\text{size of the object in the image [px]}}{\text{size of the object in the real world [m]}}$$

There are several important facts that must be considered to get a correct result:

- The *px_per_m* varies across the image due to perspective projection
- The estimate of *px_per_m* for small objects has a higher error rate than for large ones. It is therefore recommended to estimate the *px_per_m* from the license plate width rather than from its height.

3.2.3 Example

License plate width used to get the pixels per meter

$$\text{px\_per\_m} = \frac{width\ in\ image\ [px]}{physical\ width\ [m]} = \frac{80}{0.52} \approx 15$$

4 Hardware Requirements

4.1 Minimal Requirements

Processor:	single core 1.0 GHz, x86 platform (e.g. Intel Atom) /
	single core 1.0 GHz, AArch64 platform (e.g. Nvidia Jetson, Raspberry Pi)
RAM:	2 GB
Hard disk:	1 GB free space
GPU:	NVIDIA with CUDA and 2GB RAM (for GPU modules)

4.2 Recommended Requirements

Processor:	dual core 2.0 GHz, x86 platform (e.g. Intel i5) /
	dual core 2.0 GHz, AArch64 platform (e.g. Nvidia Jetson, Raspberry Pi)
RAM:	4 GB
Hard disk:	2 GB free space
GPU:	NVIDIA with CUDA and 4GB RAM (for GPU modules)

4.3 Supported Operating Systems

4.3.1 Windows

- Microsoft Windows 7/8/8.1/10
  - Win32 and x64 platform

4.3.2 Linux

- Ubuntu 18.04 and higher
  - x86_64 and AArch64 platform
- **other platforms on request**

Windows is registered trademark of Microsoft Corporation.
Linux is registered trademark of Linus Torvalds.