ADAPTIVE RECOGNITION | Automatic Number Plate Recognition
(ANPR/ALPR) Software

Carmen® FreeFlow Engine Properties

Guidelines & Best Practices

www.adaptiverecognition.com

Introduction

In this comprehensive guide, you will learn more about the properties–or parameters–of Adaptive Recognition's Carmen® FreeFlow Image Recognition API. These allow you to fine-tune the software to provide the best license plate recognition results in particular circumstances.

Each property is explained thoroughly in writing and images, highlighting what you should pay attention to. The illustrations featured herein are taken from the Carmen® FreeFlow ADI Demo (ANPR Demo for Images), a standard feature available for all available Carmen® FreeFlow Image engines for testing and demonstrational purposes.

Please note that certain properties heavily rely on each other, meaning that changing the value of one parameter will directly affect the other. These are also referenced in the document.

If you have any questions regarding the referred properties, additional parameters that may not be featured in this guide, Carmen® FreeFlow Image Recognition API, or other Adaptive Recognition license plate recognition solution in general, do not hesitate to contact us by visiting our **General Contact** page.

We hope you will enjoy using Carmen® FreeFlow Image SDK.

Table of Contents

1. Processing Time

The Carmen® FreeFlow Image engine can measure two types of time: CPU time and actual time. The former determines how long the computer's CPU is used for ANPR processes, whereas the latter tells the actual time the same processes are performed on a given image.

When the CPU has to simultaneously perform other processes, overloading it, CPU time and actual time can significantly differ from each other. As such, you should properly control the time intended for the ANPR process. To do that, set the 'timeout' and 'timeout_wall' properties.

Note that the time displayed in the results reflects the actual processing time it took for Carmen® FreeFlow Image to perform the recognition, which could be significantly longer than the value set for 'timeout.' To limit the actual elapsed time, use the 'timeout_wall' property.

1.1 timeout

'Timeout' is the CPU use time provided for a single ANPR process. The value for this property is thus for setting the maximum working time for the CPU, defining the length of time (in milliseconds) for which processing power may be dedicated entirely to license plate recognition on a single image. This way, it's guaranteed that the engine will receive the same amount of CPU time for processing every image—provided that you have the necessary processing power and don't deviate from the default timeout settings.

In practice, the timing for each process starts when **cm_findfirst()** is called. Once the given time interval is up, the engine will try and finish the last ANPR process, whether **cm_findfirst()** or **cm_findnext()** has been called, and then returns to standby mode. In other words, the value for 'timeout' only marks an approximate time provided for processes. So, when an ANPR process is close to it's conclusion, the engine may finish it, even if it means exceeding the given timeout value. Any additional call of **cm_findnext()** close to the 'timeout' value will result in "no plates found."

'Timeout' value "0" [zero] means no time limit. The default value varies by each engine, but you can set it to any positive integer for slower but more thorough processing—ideal for subpar quality images—or faster processing should you lack enough CPU power.

Note that for computers with less than 2GHz CPU clock speed, we recommend that you set the 'timeout' value to ~5000 or set a 'timeout_wall' value instead, leaving the 'timeout' value in its default state.

1. Processing Time

1.2 timeout_wall

The 'timeout_wall' value determines, in milliseconds, the time allowance for processes related to license plate recognition for one image before the engine moves onto the following image. Unlike the 'timeout' property, this setting affects real-time processing, meaning that ANPR processes can still be finished within a real time limit even if the CPU is overloaded with other tasks. Also, by setting this property, you can ensure that the module won't run much longer than the desired time, allowing for better time control per event that can be useful in real traffic situations.

Like with the 'timeout' property, 'timeout_wall' starts when **cm_findfirst()** is called.
Once the given time interval is up, the engine will try and finish the last ANPR process, whether **cm_findfirst()** or **cm_findnext()** has been called, and then returns to standby mode.
Any additional call of **cm_findnext()** close to the timeout value will result in "no plates found."

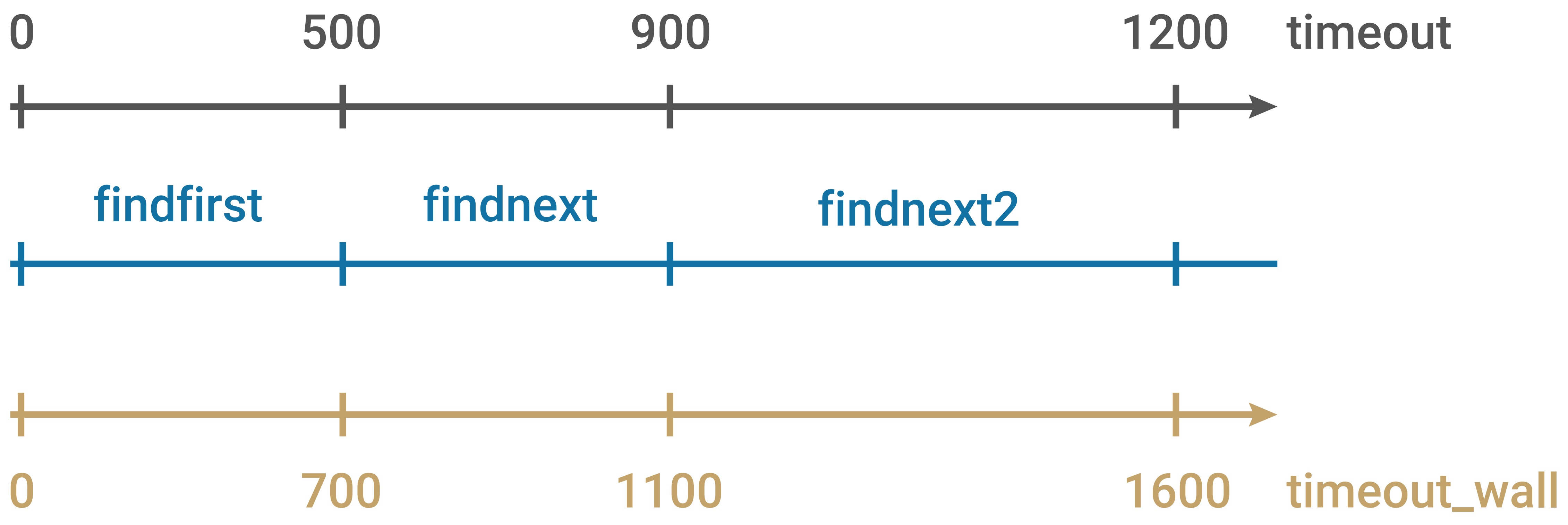
'Timeout_wall' value "0" [zero] means no time limit. The default value varies by each engine, but it can be set to any positive integer. Note, however, that results could be worse if the system is busy and the 'timeout_wall' is set too low—especially compared to timeout.

GOOD TO KNOW: it is possible to either use the 'timeout' and the 'timeout_wall' parameters together or just one of them. If either 'timeout' limit is reached, the engine will finish searching for new plates, and any additional call of **cm_findnext()** will result in "no plates found." If both properties are used simultaneously, the 'timeout_wall' value must equal or exceed the 'timeout' value.

1. Processing Time

1.2 timeout_wall

Processing one image looks like this:



Note that the above image perfectly illustrates the difference between the 'timeout' and 'timeout_wall' properties. Whereas 'timeout' can be dispersed throughout a given time period—due to high CPU usage—'timeout_wall' considers real time and, therefore, cannot be interrupted.

Example 1: timeout=1000, timeout_wall=1000

In this example, there is not enough time for the first **findnext** call due to the 'timeout_wall' value, even if the CPU could perform the process under 1 second. (500 ms for **findfirst** + 400 ms for **findnext** = 900 ms.) As such, the engine returns to standby mode after about 1 second, either potentially ignoring the **findnext** result altogether or giving a subpar result.

Example 2: timeout=1000, timeout_wall=2000

In this example, it's the 'timeout' value that limits the process. Even though the value for "real time" processing is more than enough for Carmen® FreeFlow Image to perform all three **findnext** calls–for which 1600 ms is enough–the engine still returns to standby after the first **findnext** call due to the limit set by the 'timeout' value.

Example 3: timeout=1000, timeout_wall=0

As with the other two examples, this scenario results in the process being terminated at the end of the first **findnext** call due to the chosen 'timeout' value. However, since there is no 'timeout_wall' value to tell the engine to go back to standby after a while, processing could take any amount of time. In our example, "real" processing time could fall between 1000 ms—the value for 'timeout'—and, depending on the CPU load, anything. As such, it's recommended to set a 'timeout_wall' value together with the 'timeout' value.

Example 4: timeout=1000, timeout_wall=1500

In this case, the ANPR process returns to standby after ~1500 ms, but, as the engine has found an object on the image that may be a license plate during the second findnext process, it performs the examination even if it somewhat exceeds the time value provided for 'timeout_wall.' In such circumstances, the result can be "No result"—if there is indeed no license plate on the image or if the engine couldn't successfully verify it—or the required plate information.

1. Processing Time

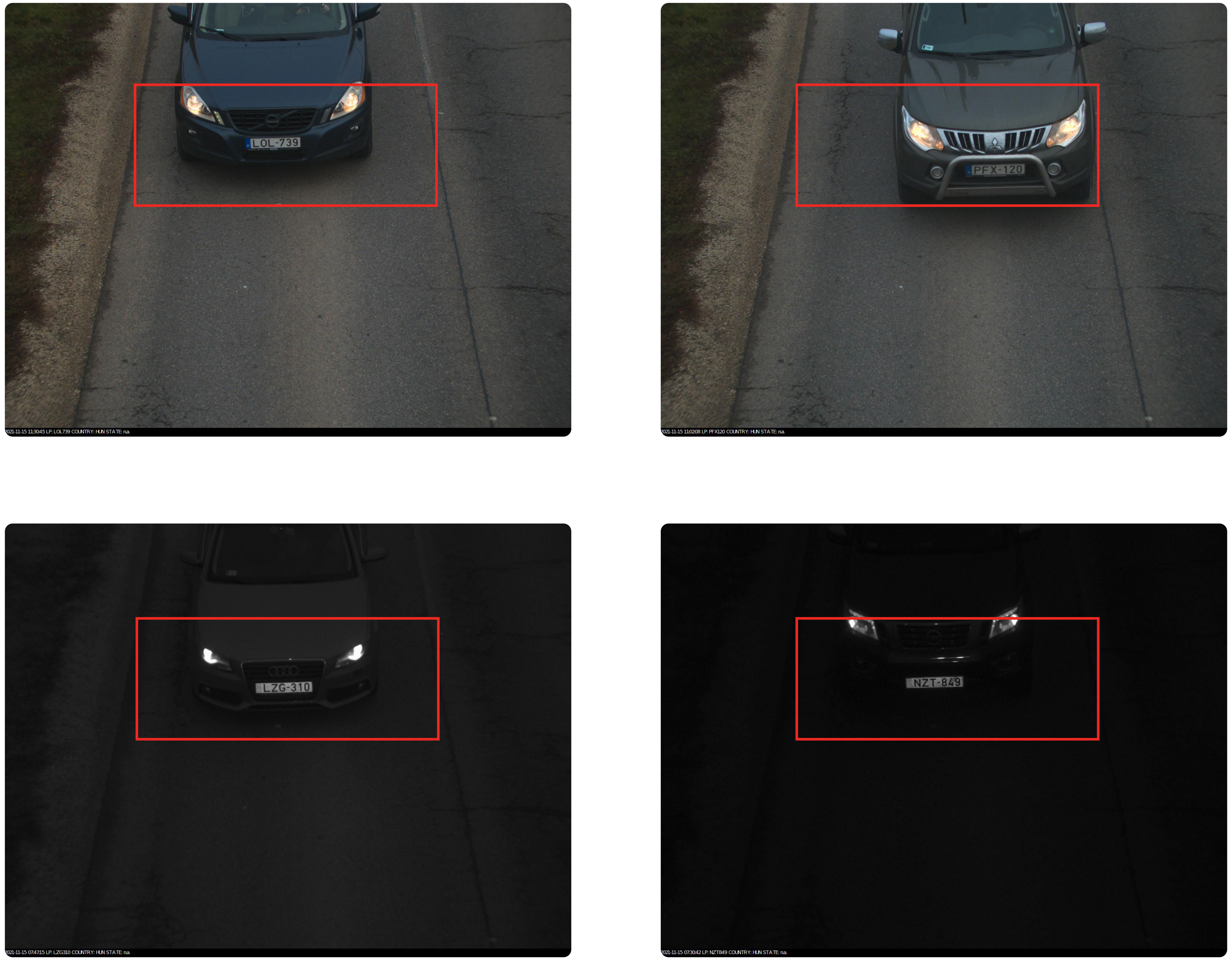
1.3 adapt_environment

With this setting, you can substantially boost recognition speed via machine learning. By turning on this option and feeding a couple of images to the engine, Carmen® FreeFlow Image can anticipate the type, location, and specific physical attributes of license plates on the image. This will accelerate the recognition process—provided that recognition occurs at the same physical location.

The values of ‘adapt_environment’ are as follows:

- 0: the property turned off
- 1: Carmen® FreeFlow Image starts anticipating license plates based on their country/state and type on the processed image
- 2: based on the frequency of license plates appearing on a specific part of previously processed images, the engine predicts where to expect plates to appear on images
- 4: based on previously processed pictures, Carmen® FreeFlow Image automatically sets a value for the height of license plate characters
- 8: based on previously processed images, the software sets a value for the width of license plate characters
- 16: based on previous pictures, Carmen® FreeFlow Image sets the value for the slant of license plate characters
- 32: based on processed images, the software auto-sets the predicted value for the slope of license plate characters
- 64: Carmen® FreeFlow Image automatically sets up the difference between the grayscale value of the license plate characters and the plate background based on previously processed images
- -1: activates all the above values. Use this value if images are collected at the same physical location.

Note that there is the option to use the above attributes simultaneously. For example: if you want Carmen® FreeFlow Image to adapt to location, license plate position in the image, height, and slope, set the value to the total of the individual values. In our case, that value would be 39 (1+2+4+32=39).



1. Processing Time

1.3 adapt_environment

This montage shows how the ‘adapt_environment’ property helps the Carmen® FreeFlow Image engine adapt to each image by automatically modifying specific settings based on all images processed so far. This includes where the search for the license plate on the image should start. In short, by continuously processing image after image, the engine learns to start ignoring irrelevant details in the pictures—anything aside from the zones we indicated with a red rectangle—and, therefore, can process images much faster each time.

Note that this montage depicts each vehicle at the same location in the almost exact position because the images were captured with Adaptive Recognition’s Vidar high-speed traffic monitoring cameras that feature built-in laser triggering, allowing highly-precise image capturing.

Results from another camera without laser triggering may deviate from this example.

1. Processing Time

1.3 adapt_environment

IN PRACTICE:

The following example demonstrates how ‘adapt_environment’ affects ANPR time’s average and standard deviation. During our test conducted on an average computer running the latest version of Windows 10, we processed a total of 2884 images in 28 sequential blocks consisting of the same 103 test images.
To ensure the most optimal ANPR images, we used Vidar high-speed traffic monitoring ANPR cameras with built-in laser triggering.
(More about how to achieve the best ANPR-ready images in our **designated manual**.)

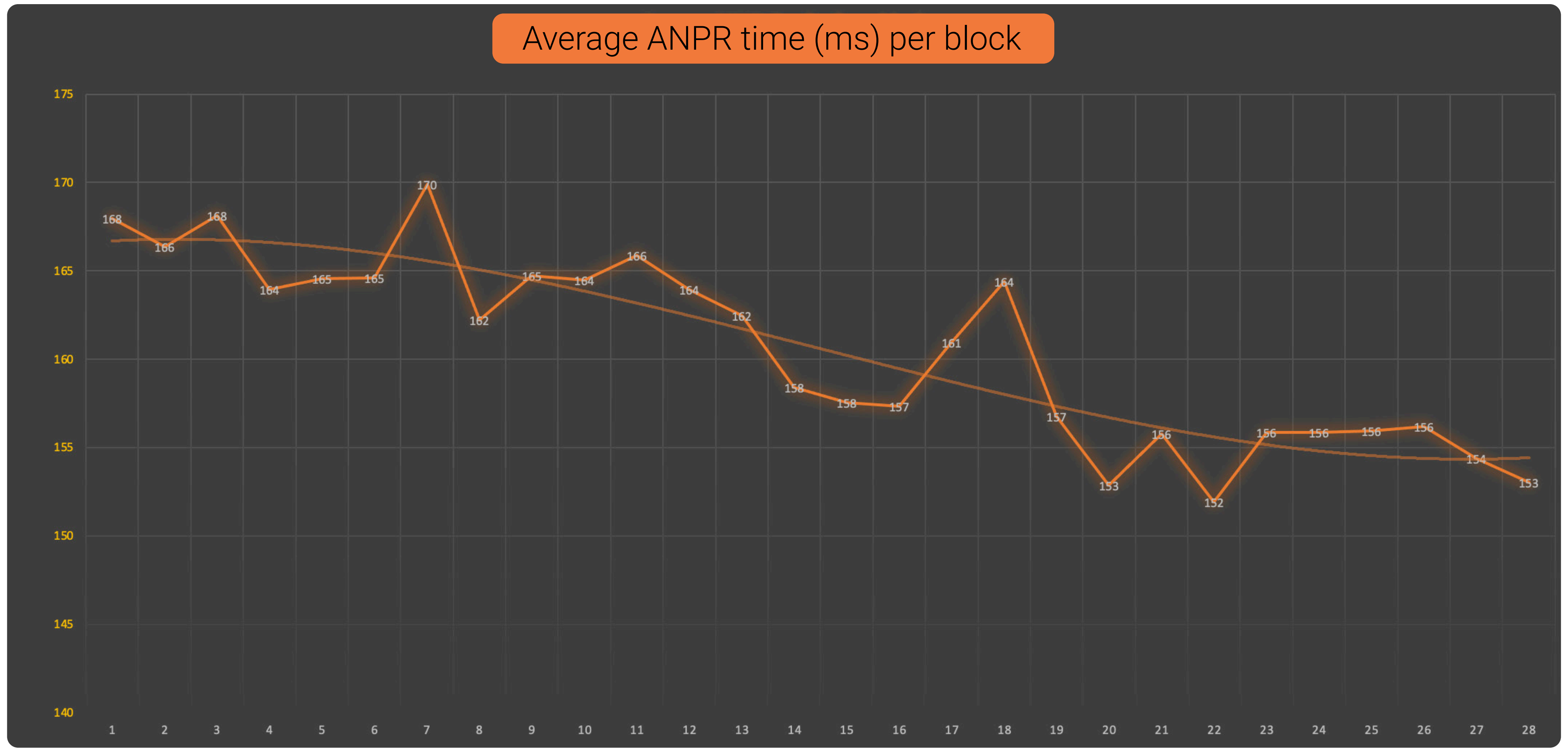
In our example, the value given for ‘adapt_environment’ was set to 63 for the following reasons:

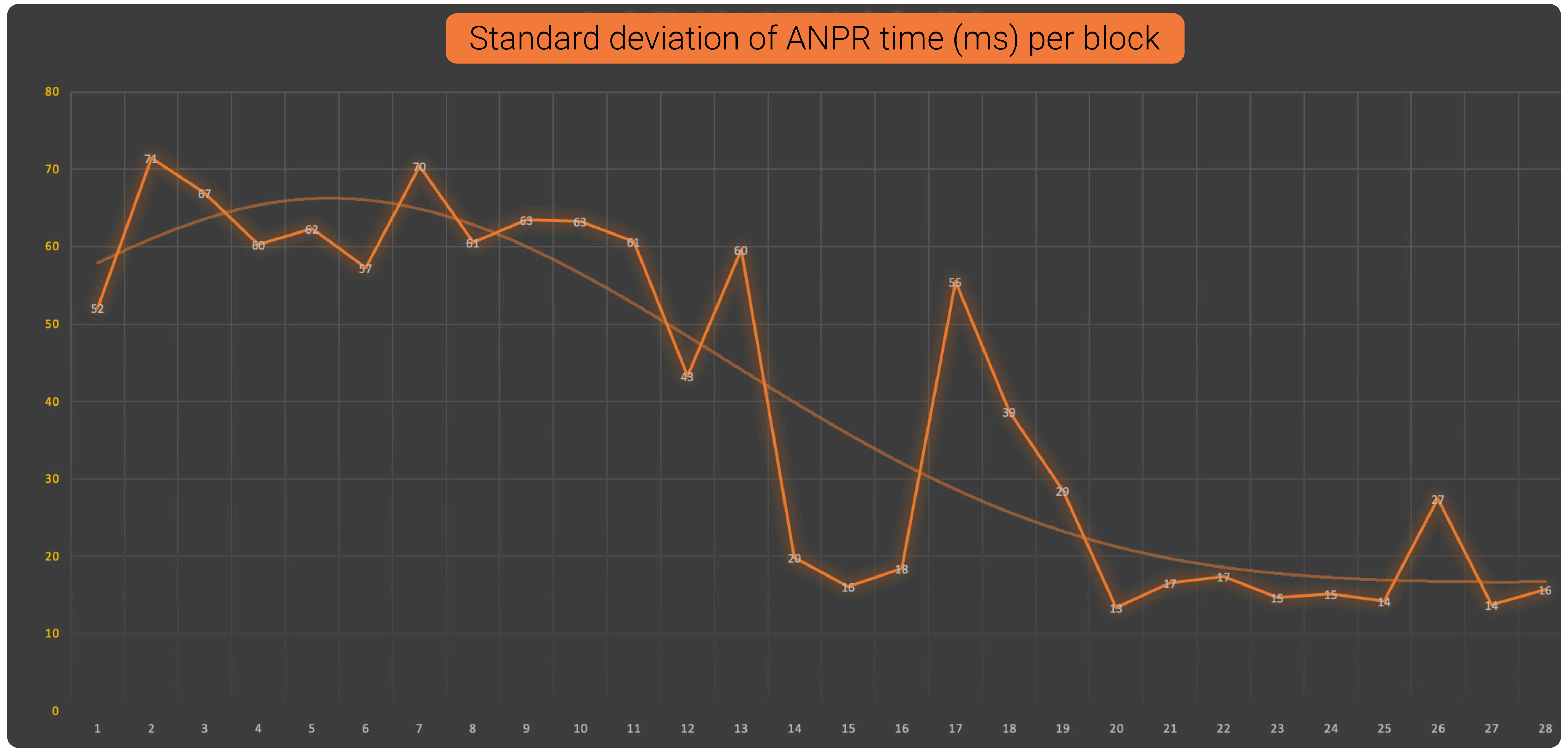
- The license plates were mostly the same type and of the same country (country/type)
- The size, slant, slope, and position (posfreq) of the license plates were almost identical
- ‘Contrast_min’ wasn’t adapted to guarantee accurate results regardless of time.

A polynomial trend line was applied in the graphs below to represent the change in the current and estimated future values.

1. Processing Time

1.3 adapt_environment





In conclusion, it's clearly visible from comparing blocks 1 and 28 that the average CPU (ANPR) time improved by around 10%, and standard deviations significantly decreased.
In other words, each CPU time got closer to the average after each instance, meaning a minimum amount of deviations compared to the average.

ATTENTION: before using 'adapt_environment,' consider the following:

- If the 'timeout' value is too low, the engine may skip infrequent but processable license plates before recognition is finished
- The accuracy of country/state identification becomes more sensitive, requiring either a carefully selected timeout value or a secondary screening to spot false-positive results
- Errors or misreads may not be reproduced due to the continuously changing recognition parameters.

1. Processing Time

1.4 contrast_min

You can set the difference between the grayscale value of the license plate characters and the plate background with this parameter. In practice, when the ANPR process is running, Carmen® FreeFlow Image compares two adjacent pixels and their respective grayscale values. The given ‘contrast_min’ value then helps the engine determine whether the pixel belongs to an alphabetical/numeric character or the background of the license plate based on the difference between the grayscale values.

Example:

Grayscale value:	1 18	1 34
Difference:	17	33
Verdict	both character	left character, right background

‘Contrast_min’ is set to 20. On the left, the difference between the grayscale values of two adjacent pixels is 18, so Carmen® FreeFlow Image considers both pixels belonging to a license plate character. On the left, the difference between grayscale values is 34.
This means that Carmen® FreeFlow Image assigns the left pixel to a character and the right pixel to the license plate background—if the license plate background is white. In the case of a darker license plate with lighter characters, Carmen® FreeFlow Image assigns the right pixel to the character and the left to the license plate background.

The value can be any positive whole number between “1” and “255.”

Note that the default value varies with each engine release. To query the default ‘contrast_min’ property value, use the **GetProperty()** function.

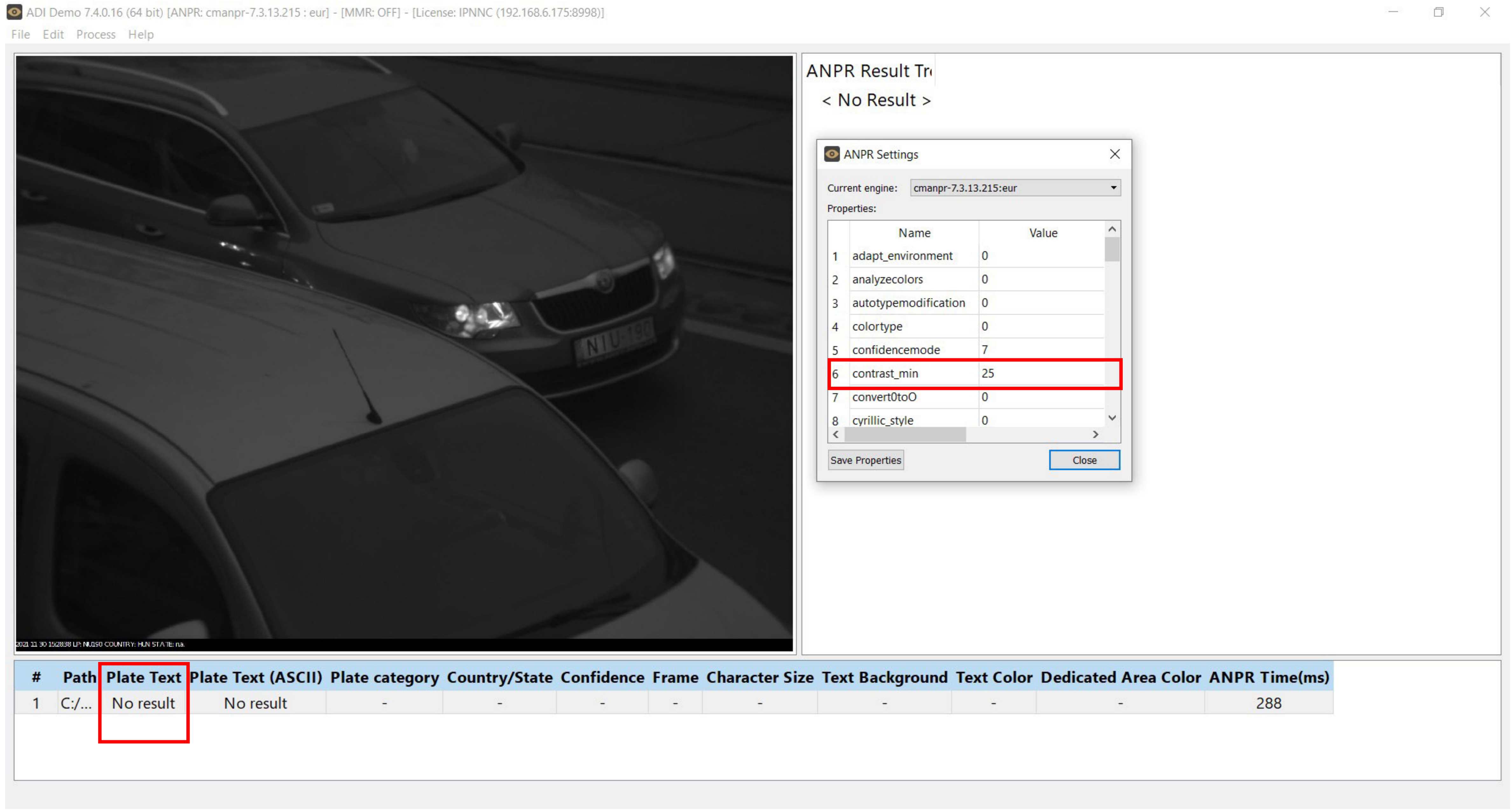
ATTENTION: if the grayscale contrast between the plate characters and the plate background is smaller than the value given for ‘contrast_min,’ no result will be returned.

1. Processing Time

1.4 contrast_min

In this example, after setting the ‘contrast_min’ value to a lower number, Carmen® FreeFlow Image was able to successfully read and recognize the license plate.

Before



ADI Demo 7.4.0.16 (64 bit) [ANPR: cmanpr-7.3.13.215 : eur] - [MMR: OFF] - [License: IPNNC (192.168.6.175:8998)]

< No Result >

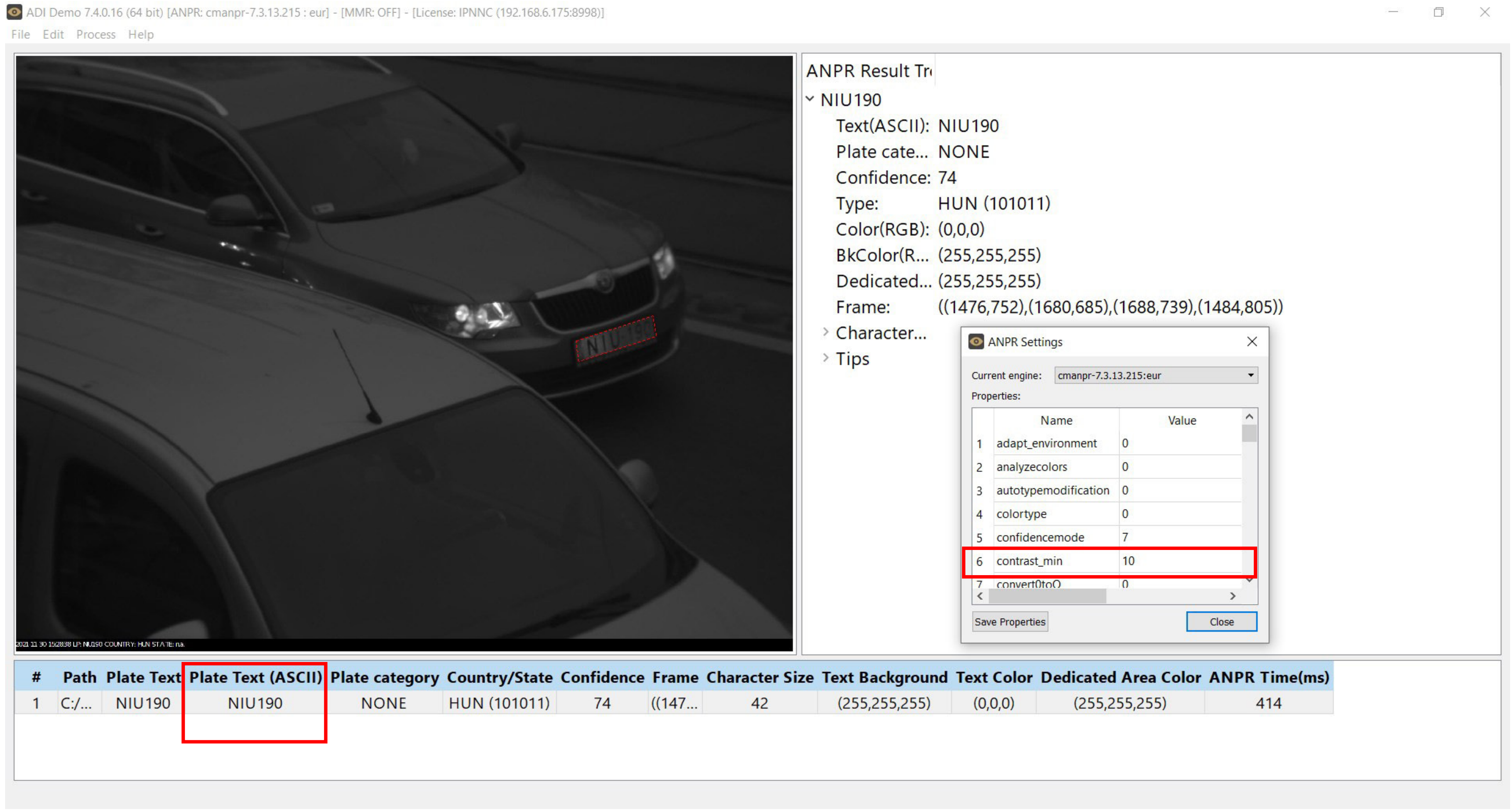
Current engine: cmanpr-7.3.13.215:eur

Properties:

	Name	Value
1	adapt_environment	0
2	analyzecolors	0
3	autotypemodification	0
4	colortype	0
5	confidencemode	7
6	contrast_min	25
7	convert0toO	0
8	cyrillic_style	0

#	Path	Plate Text	Plate Text (ASCII)	Plate category	Country/State	Confidence	Frame	Character Size	Text Background	Text Color	Dedicated Area Color	ANPR Time(ms)
1	C:/...	No result	No result	-	-	-	-	-	-	-	-	288

After



ADI Demo 7.4.0.16 (64 bit) [ANPR: cmanpr-7.3.13.215 : eur] - [MMR: OFF] - [License: IPNNC (192.168.6.175:8998)]

NIU190
Text(ASCII): NIU190
Plate cate... NONE
Confidence: 74
Type: HUN (101011)
Color(RGB): (0,0,0)
BkColor(R... (255,255,255)
Dedicated... (255,255,255)
Frame: ((1476,752),(1680,685),(1688,739),(1484,805))
Character...
Tips

Current engine: cmanpr-7.3.13.215:eur

Properties:

	Name	Value
1	adapt_environment	0
2	analyzecolors	0
3	autotypemodification	0
4	colortype	0
5	confidencemode	7
6	contrast_min	10
7	convert0toO	0

#	Path	Plate Text	Plate Text (ASCII)	Plate category	Country/State	Confidence	Frame	Character Size	Text Background	Text Color	Dedicated Area Color	ANPR Time(ms)
1	C:/...	NIU190	NIU190	NONE	HUN (101011)	74	((147...	42	(255,255,255)	(0,0,0)	(255,255,255)	414

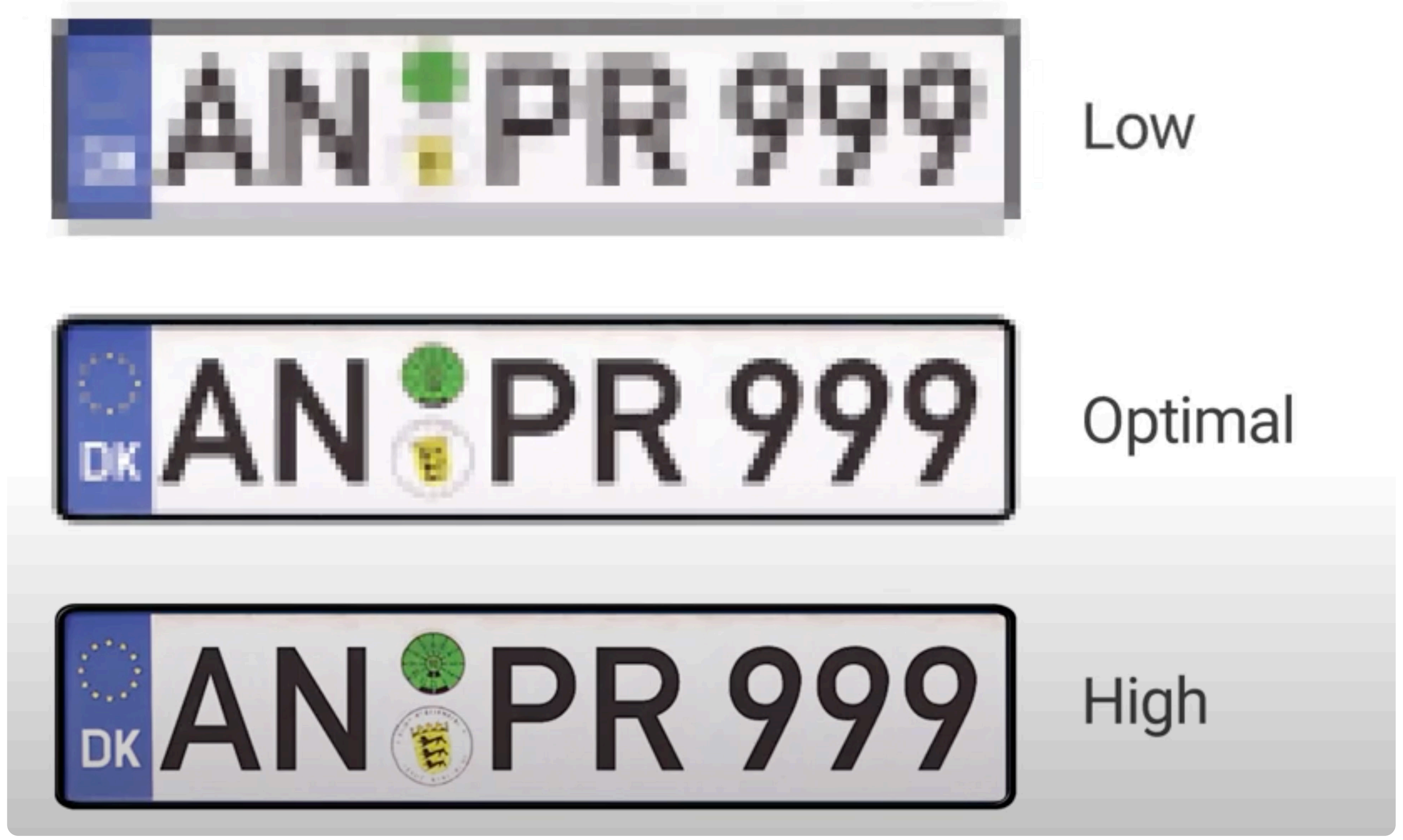
2. License Plate Geometry

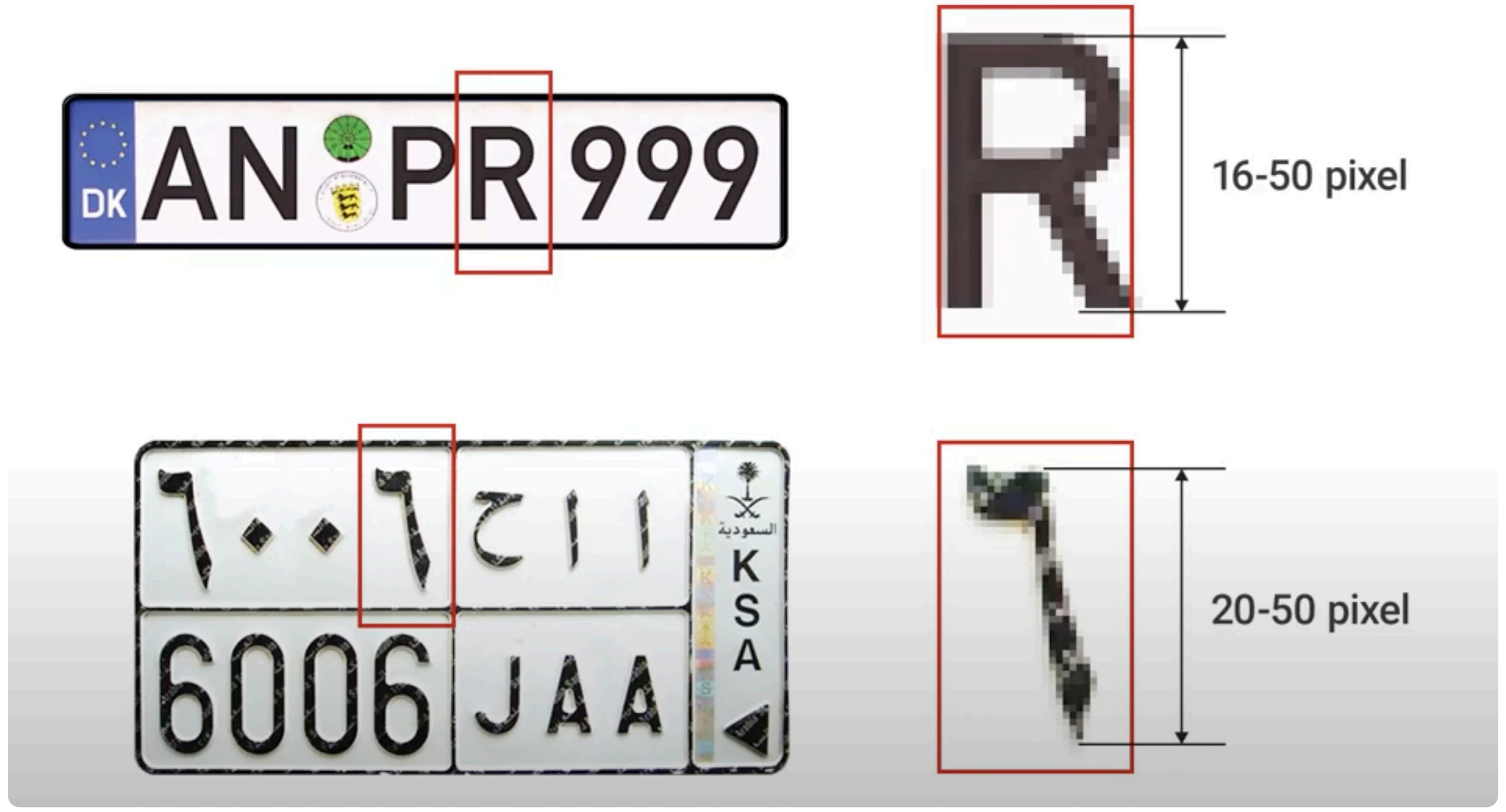
2.1 size

This value gives the average height of license plate characters in the image in pixels. You can choose any value, provided it's a positive integer between 10 and 80.

Note that the default value varies with each engine release.
To query the default property value, use the **GetProperty()** function.
For best results, choose a value between 20 and 50, depending on the character heights in the input images.

ATTENTION: the value of 'size' must equal or be greater than the value of 'size_min' and must equal or be less than the value provided for 'size_max.'





2. License Plate Geometry

2.2 size_max

With this value, you can determine the maximum height of the license plate characters in the image in pixels.

As of engine version 7.3.14.103, you can choose any value for this property, provided it’s a positive integer greater or equal to the actual ‘size’ value or -1.

ATTENTION: we highly recommend leaving the default value for the best results. The default value varies with each engine release and can be queried by the **GetProperty()** function.

Note that if you set the value of ‘size_max’ to -1, the ANPR engine will try to find a number plate on the image no matter its size. However, this may result in a longer running time. We recommend using this setting if you are unsure about the size of the characters on the plates occurring on the image.

ATTENTION: the value of ‘size_max’ must equal or be greater than the value of ‘size.’

2.3 size_min

With this value, you can provide the minimum height of license plate characters in the image in pixels. You can choose any value, provided it’s a positive integer and falls between 10 and the value given under ‘size.’

Note that the default value varies with each engine release. To query the default property value, use the **GetProperty()** function. For best results, leave the default value as it is.

ATTENTION: the value of ‘size_min’ must equal or be less than the value of ‘size.’

2. License Plate Geometry

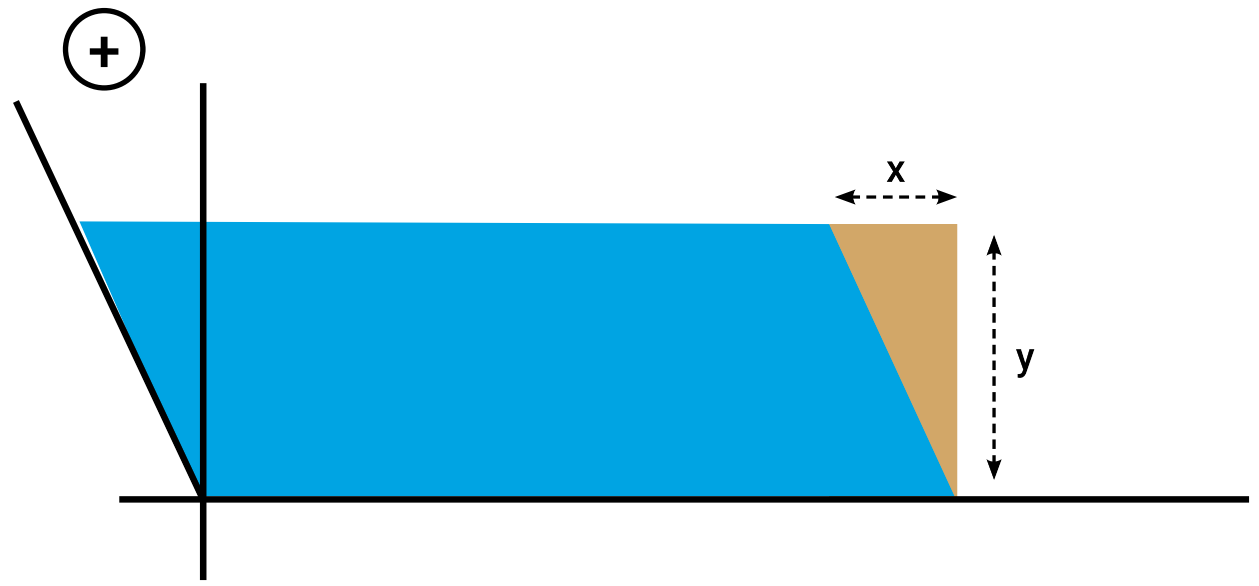
2.4 slant

This value is for determining the average slant of the license plate characters in the image. The value is always given in percent (%): if the value is a positive number (0 – 100), the vertical axis of the characters slants to the left. If the value is a negative number (0 – -100), the vertical axis of the characters slants to the right.

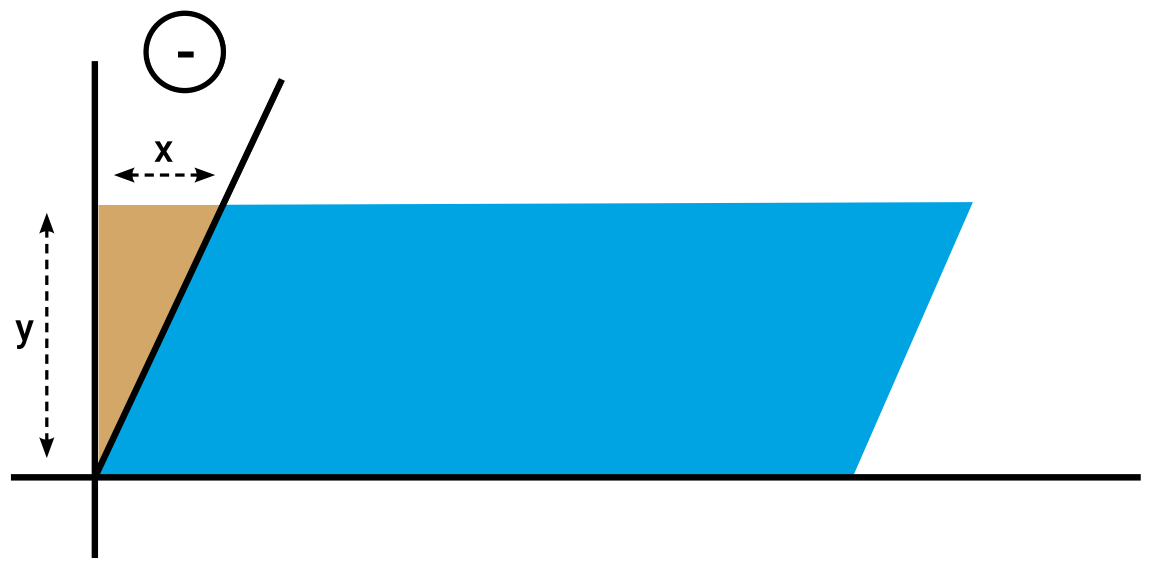
Please note that the default value varies with each engine release. To query the default property value, use the **GetProperty()** function. For the best results, we recommend leaving the default value as is.

ATTENTION: the value of ‘slant’ must equal or be greater than the value of ‘slant_min’ and must equal or be less than the value provided for ‘slant_max.’

Positive SLANT



Negative SLANT



$$\text{Slant} = \frac{x}{y} * 100$$

2. License Plate Geometry

2.5 slant_max

With this value, you can determine the maximum slant of the license plate characters in the image. The value is always in percent (%): if the value is a positive number, the vertical axis of the characters slants to the left; in the case of characters slanting to the right, the numeric value will be negative. You can choose any value, provided it’s an integer number falling between the value given under ‘slant’ and 100.

Note that the default value varies with each engine release. To query the default property value, use the **GetProperty()** function. For best results, it’s recommended to leave the default value as it is.

ATTENTION: the value of ‘slant_max’ must equal or be greater than the value of ‘slant.’

2.6 slant_min

The ‘slant_min’ value determines the minimum slant of the license plate characters in the image. The value is always in percent (%): if the value is a positive number, the vertical axis of the characters slants to the left; in the case of characters slanting to the right, the numeric value will be negative. You can choose any value, provided it’s an integer number falling between -100 and the value given under ‘slant.’

Note that the default value varies with each engine release. To query the default property value, use the **GetProperty()** function. For the best results, leave the default value as it is.

ATTENTION: the value of ‘slant_min’ must equal or be less than the value of ‘slant.’

2. License Plate Geometry

2.7 slope

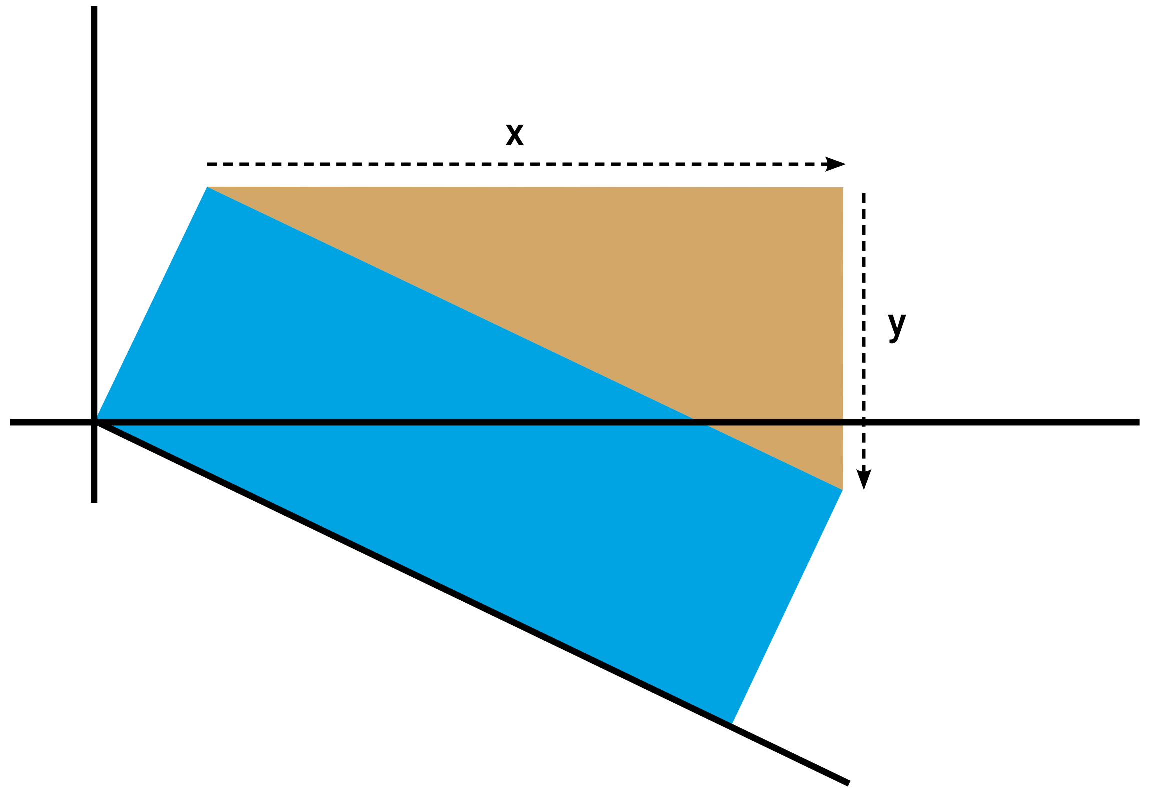
This property is for setting the average slope of the number plate in the image. It is represented in percent (%), and it is positive if the number plate is rotated clockwise and negative if it is rotated anti-clockwise. The value can be any positive integer falling between -400 and 400.

For example, if the 'slope' value is set to +200, the license plate is rotated 90° clockwise on the image. A +400 slope means that the number plate appears upside down.

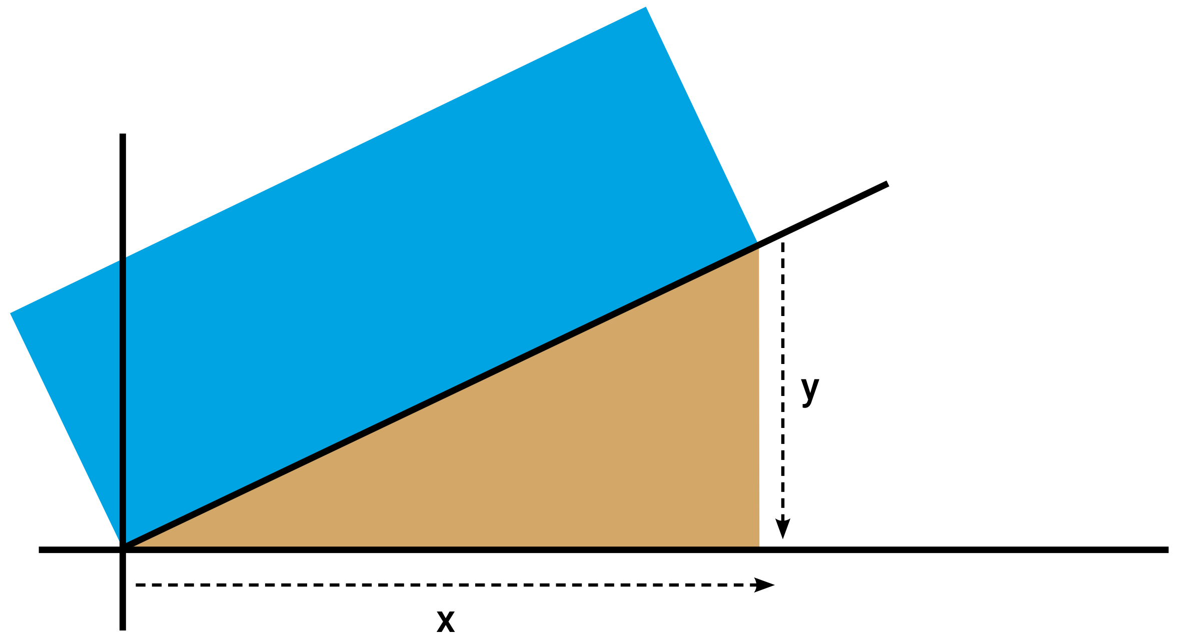
Please note that the default value varies with each engine release. The query the default property value, use the **GetProperty()** function. For the best results and an optimal running time, we recommend leaving the default value as is.

ATTENTION: the value of 'slope' must equal or be greater than the value of 'slope_min' and must equal or be less than the value provided for 'slope_max.'

⊕ **Positive SLOPE**



⊖ **Negative SLOPE**



$$\text{Slope} = \frac{y}{x} * 100$$

2. License Plate Geometry

2.8 slope_max

This setting determines the maximum slope of the number plate in the image. It is represented in percent (%), and it is positive if the number plate is rotated clockwise and negative if it is rotated anti-clockwise.

The value can be any positive integer falling between the actual 'slope' value and 400. However, setting the value of 'slope_max' above 100 could result in a significant increase in ANPR running time, so only use these values if you absolutely have to.

Note that the default value varies with each engine release. To query the default property value, use the **GetProperty()** function. For the best results, it is recommended to leave the default value as it is.

ATTENTION: the value of 'slope_max' must equal or be greater than the value of 'slope.'

2.9 slope_min

This parameter is for determining the minimum slope of the number plate in the image. It is represented in percent (%), and it is positive if the number plate is rotated clockwise and negative if it is rotated anti-clockwise.

The value can be any positive integer falling between -400 and the actual 'slope' value. It's worth adding that setting the value of 'slope_min' below -100 could significantly increase ANPR running time, so only use these values if it's a must.

Note that the default value varies with each engine release. To query the default property value, use the **GetProperty()** function. For best results, leave the default value as it is.

ATTENTION: the value of 'slope_min' must equal or be less than the value of 'slope.'

3. Filtering Outputs Based on License Plate Characteristics

3.1 general

This setting lets you select a mode – for general or special delivery results – to read license plates, ADR/E-ADR signs, and plates with unknown (cmNP::type) types.

The 'general' property allows you to filter four license plate recognition results:

1. Text results with unidentified license plate type (type=0)
2. Results with identified license plate type (0<type<997000)
3. Results with identified ADR sign (997000≤type≤997999)
4. Results with identified E-ADR sign (999980≤type≤999999)

Please refer to the table below regarding the results returned for each possible value of the 'general' property:

Expected type of the result

Values of general	Unidentified license plate type (type=0)	Identified license plate type (0 < type < 997000)	ADR plate (997000 < type < 997999)	Empty ADRplate (999980 < type < 999999)
0		x		
1	x	x		
2 (do not use)				
3	x			
4		x	x	
5	x	x	x	
6			x	
7	x		x	
8		x		x
9	x	x		x
10				x
11	x			x
12		x	x	x
13	x	x	x	x
14			x	x
15	x		x	x

3. Filtering Outputs Based on License Plate Characteristics

3.1 general

The values of ‘general’ refer to as follows:

- 0, 1, 3: unidentified and/or identified license plate recognition without ADR/E-ADR
- 4–15: unidentified and/or identified license plate recognition and/or ADR/E-ADR recognition

ATTENTION: value “2” returns no results. Please do not use it!

Note that the value of this property directly affects the value of ‘typeweight’ (see below). The default value of ‘general’ varies with each engine release. To query the default ‘general’ value, use the **GetProperty()** function.

3. Filtering Outputs Based on License Plate Characteristics

3.2 typeweight

Available from version 7.2.8.46

By setting the ‘typeweight’ value, you can order Carmen® FreeFlow Image to prioritize or omit certain license plate types and countries/states/provinces during the recognition process. This feature is useful for projects where you want to monitor vehicles registered in a specific country/state/province or filter special vehicles from regular ones (for example, permitting diplomatic cars to access restricted areas).

By default, the weight for all types is set to 1000. The higher the value, the more the engine will focus on license plates originating from the chosen country/state/province.
Negative values allow Carmen® FreeFlow Image to recognize license plates but omit them from results. If the value is set to “0” [zero], Carmen® FreeFlow Image omits the defined license plate type but, at the same time, matches them to a type most similar to the omitted plate type.

To properly use this parameter, type **typeweight=”[abcdef],[value]”**, where “abcdef” stands for an exemplary 6 digit AR plate type. In this code, “a” stands for the region, “bc” is the country/state/province code, and “def” stands for the feature(s) of the license plate. When providing this property, only “a” and the weight value is obligatory; “bc” and “def” are for further defining the recognition.

Note that you can create entire strings with the typeweight property–**typeweight=”[abcdef],[value];[abcdef],[value];…”**–or reset previous settings by simply overwriting them with the new typeweight values.

3. Filtering Outputs Based on License Plate Characteristics

3.2 typeweight

Example 1:

If you have the EUR engine and you would like to get license plates only from European countries, you can ignore Asia (Iraq) and Africa (Morocco, Tunisia) by using the following prompt:

typeweight="2,-1000; 4,-1000; 1, 1000;"

Example 2:

There are license plates in Belgium and Hungary which look almost identical. If your installation is in Hungary, it is more likely to have Hungarian license plates than Belgian. As such, you need to raise the typeweight for Hungary and decrease it for Belgium by typing the following:

typeweight=" 101,5000; 123,500;"

ATTENTION: by changing the values for 'typeweight', the engine automatically changes the value of the 'general' property to keep consistency. The two properties directly affect each other, influencing recognition time and accuracy. Because of this, please use either 'general' or 'typeweight', but never both.

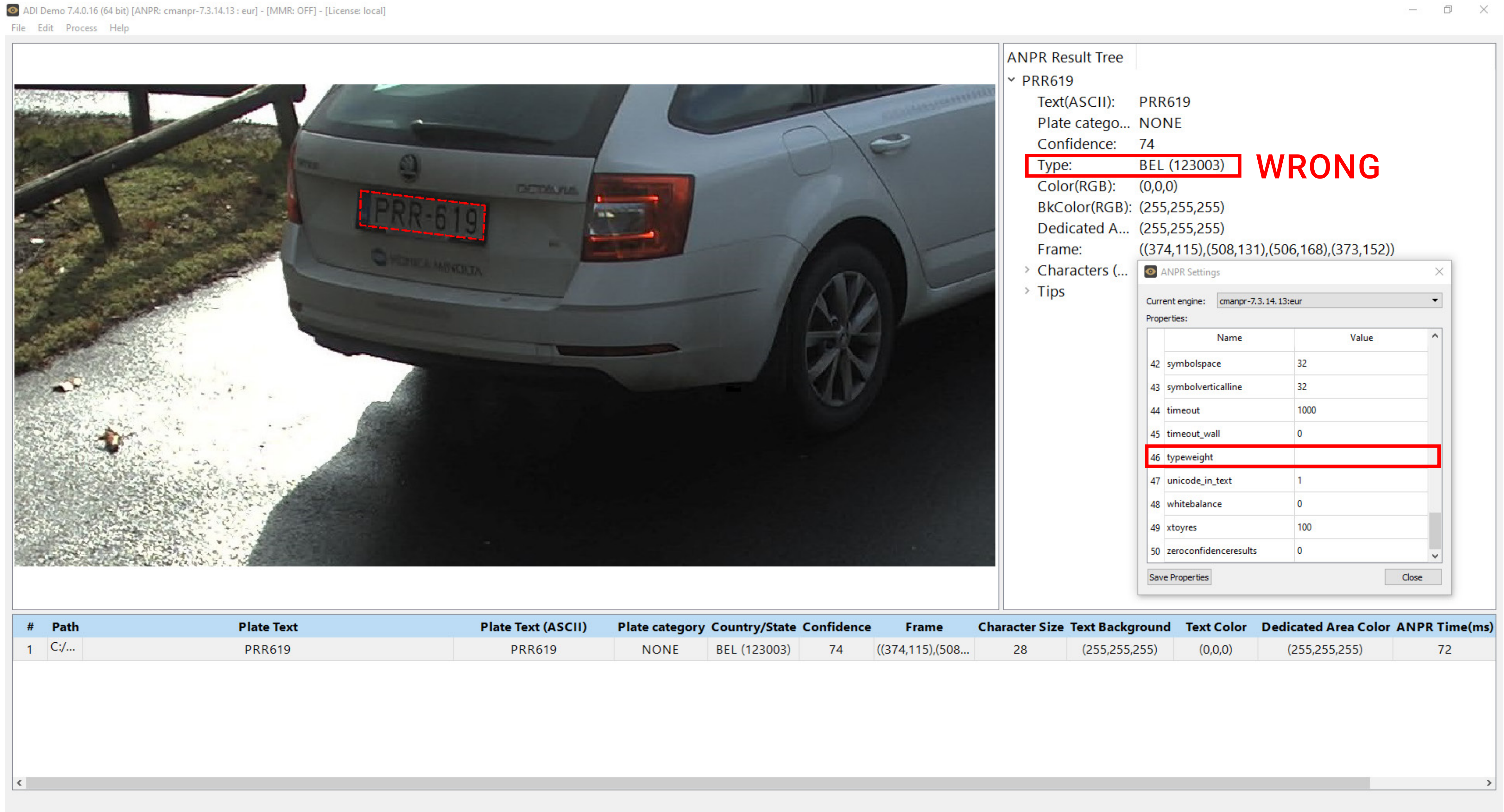
For country and state codes, please refer to the **chapter titled "Country and State IDs"** in the appendices of our comprehensive list.

3. Filtering Outputs Based on License Plate Characteristics

3.2 typeweight

In this example, the car was incorrectly identified as Belgian due to the similarities between Belgian and Hungarian license plates. However, after applying certain ‘typeweight’ values, Carmen® FreeFlow Image correctly identified the vehicle as Hungarian.

Before


ADI Demo 7.4.0.16 (64 bit) [ANPR: cmanpr-7.3.14.13 : eur] - [MMR: OFF] - [License: local]
File
Edit
Process
Help
PRR-619

ANPR Result Tree

PRR619

Text(ASCII):	PRR619
Plate catego...	NONE
Confidence:	74
Type:	BEL (123003)
Color(RGB):	(0,0,0)
BkColor(RGB):	(255,255,255)
Dedicated A...	(255,255,255)
Frame:	((374,115),(508,131),(506,168),(373,152))

WRONG

Characters (...

Tips

ANPR Settings

Current engine: cmanpr-7.3.14.13:eur

Properties:

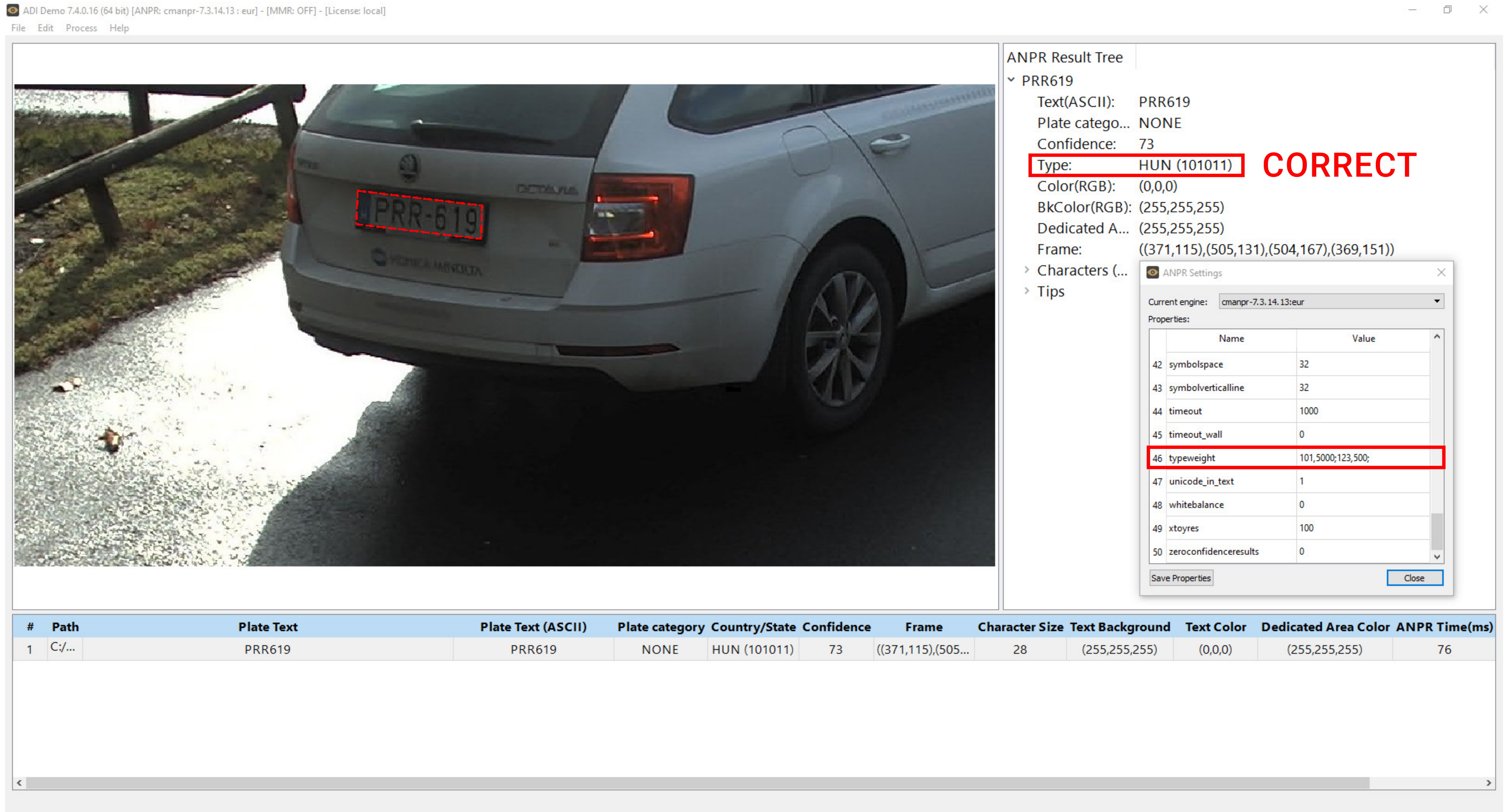
	Name	Value
42	symbolspace	32
43	symbolverticalline	32
44	timeout	1000
45	timeout_wall	0
46	typeweight	
47	unicode_in_text	1
48	whitebalance	0
49	xtoyres	100
50	zeroconfidenceresults	0

Save Properties

Close

#	Path	Plate Text	Plate Text (ASCII)	Plate category	Country/State	Confidence	Frame	Character Size	Text Background	Text Color	Dedicated Area Color	ANPR Time(ms)
1	C:/...	PRR619	PRR619	NONE	BEL (123003)	74	((374,115),(508...	28	(255,255,255)	(0,0,0)	(255,255,255)	72

After


ADI Demo 7.4.0.16 (64 bit) [ANPR: cmanpr-7.3.14.13 : eur] - [MMR: OFF] - [License: local]
File
Edit
Process
Help
PRR-619

ANPR Result Tree

PRR619

Text(ASCII):	PRR619
Plate catego...	NONE
Confidence:	73
Type:	HUN (101011)
Color(RGB):	(0,0,0)
BkColor(RGB):	(255,255,255)
Dedicated A...	(255,255,255)
Frame:	((371,115),(505,131),(504,167),(369,151))

CORRECT

Characters (...

Tips

ANPR Settings

Current engine: cmanpr-7.3.14.13:eur

Properties:

	Name	Value
42	symbolspace	32
43	symbolverticalline	32
44	timeout	1000
45	timeout_wall	0
46	typeweight	101,5000;123,500;
47	unicode_in_text	1
48	whitebalance	0
49	xtoyres	100
50	zeroconfidenceresults	0

Save Properties

Close

#	Path	Plate Text	Plate Text (ASCII)	Plate category	Country/State	Confidence	Frame	Character Size	Text Background	Text Color	Dedicated Area Color	ANPR Time(ms)
1	C:/...	PRR619	PRR619	NONE	HUN (101011)	73	((371,115),(505...	28	(255,255,255)	(0,0,0)	(255,255,255)	76

3. Filtering Outputs Based on License Plate Characteristics

3.3 location

Available from version 7.3.14.13

If you set this parameter, the ANPR engine will modify its search based on the type of plates that generally occurs at the set location. Similarly to the ‘typeweight’ parameter, it modifies the weights associated to the types, but in a more automated and detailed way. Depending on the set location, this parameter may also change other settings to improve the recognition rate.

To use this parameter, type **SetProperty(“location” , “[XYZ]”)**, where XYZ stands for the multi-letter code* of the country/state you are performing license plate recognition in (i.e., HUN for Hungary, US-OK for Oklahoma). By entering **GetProperty(“location”)**, the engine will let you know the actual location, whereas typing **(“location_all”)** right after the **GetProperty** function will provide a list of all locations known by the engine. To set everything back to default, use the code **SetProperty(“location” , “0”)**.

ATTENTION: if you try to set a location unknown to the engine, it will return an “unknown property” error. Also, if the location hasn’t been set prior, it will return “unset.”

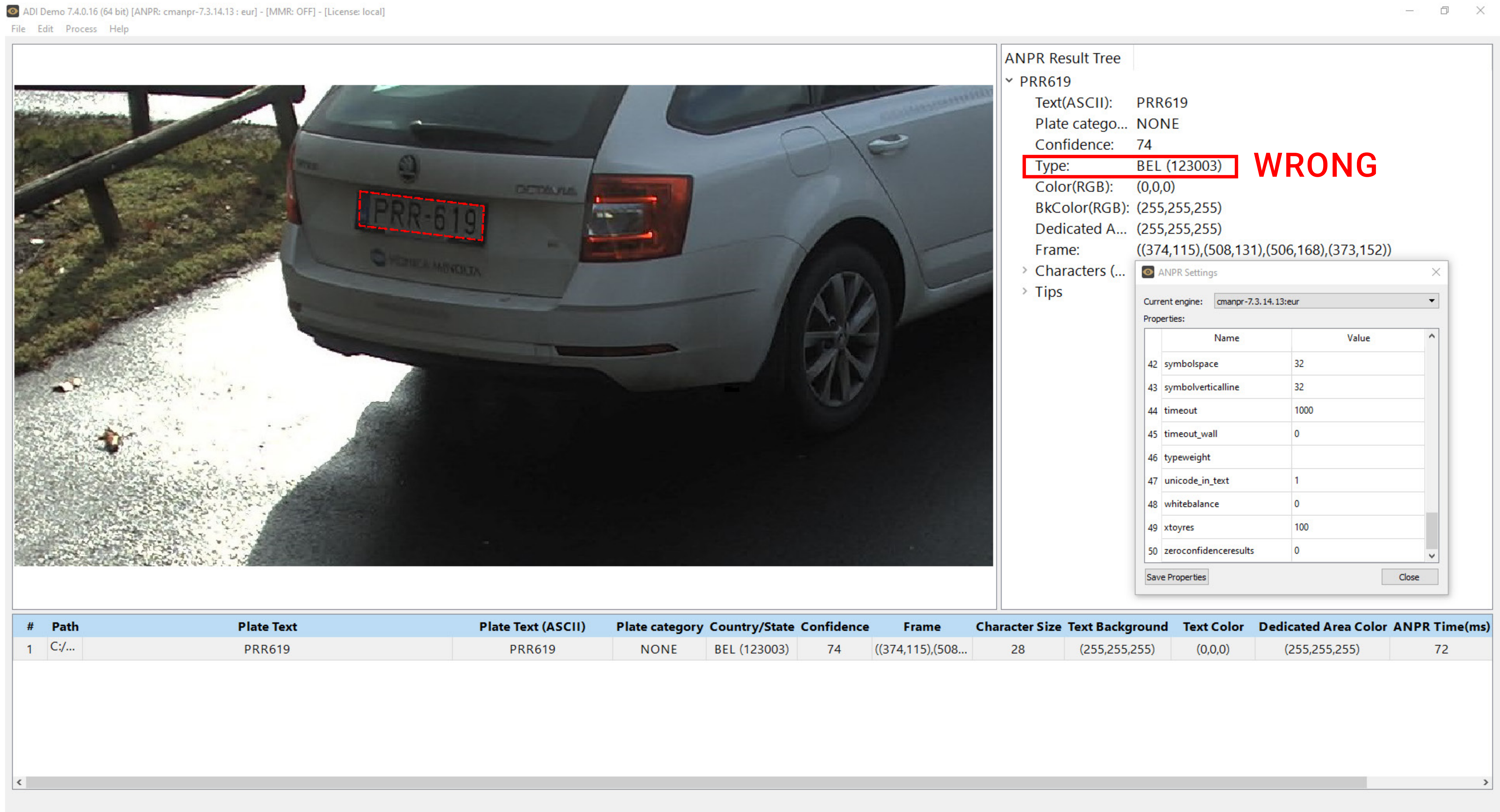
Note that despite ‘location’ being a better alternative to ‘typeweight’ due to its easier use and better results, the two parameters can be used simultaneously. Note, however, that setting the ‘location’ parameter will reset all previously set typeweights. Although it isn’t recommended, if you want to set some typeweights manually and use the ‘location’ parameter, you should set the ‘location’ first and set additional typeweights after.

* Usually that code is the given country’s/state’s ISO code. To ensure the appropriate location is provided for the engine, either run a query for the known locations and pick what’s appropriate or check Appendix “Country and State IDs” in our Carmen® Reference Manual for a comprehensive list of all location codes.

3. Filtering Outputs Based on License Plate Characteristics

3.3 location

Before


ADI Demo 7.4.0.16 (64 bit) [ANPR: cmanpr-7.3.14.13 : eur] - [MMR: OFF] - [License: local]
File Edit Process Help
PRR-619

ANPR Result Tree
- PRR619
  - Text(ASCII): PRR619
  - Plate catego... NONE
  - Confidence: 74
  - Type: BEL (123003)
  - Color(RGB): (0,0,0)
  - BkColor(RGB): (255,255,255)
  - Dedicated A... (255,255,255)
  - Frame: ((374,115),(508,131),(506,168),(373,152))
  - Characters (...
  - Tips

WRONG

ANPR Settings

Current engine: cmanpr-7.3.14.13:eur

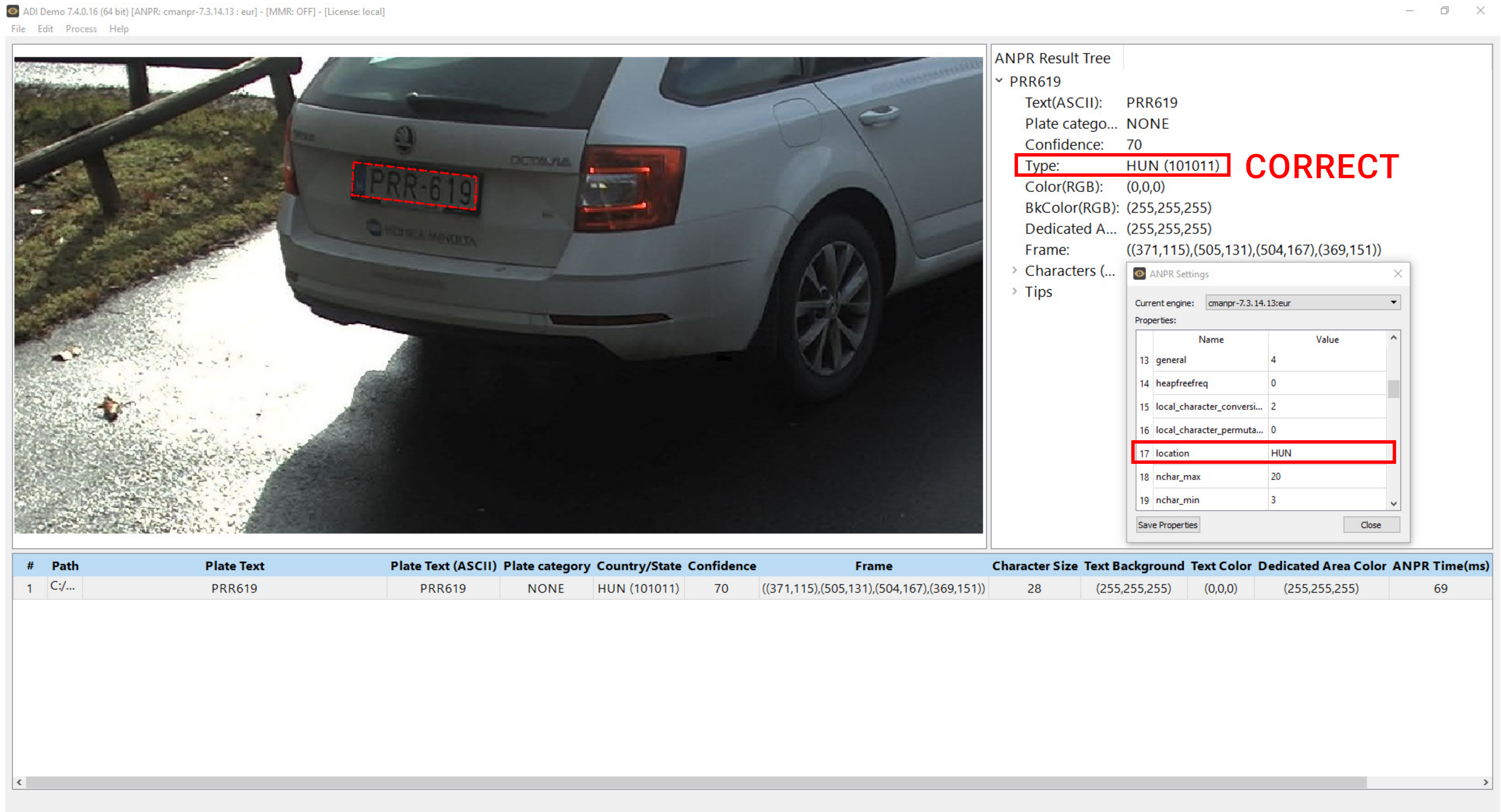
Properties:

	Name	Value
42	symbolspace	32
43	symbolverticalline	32
44	timeout	1000
45	timeout_wall	0
46	typeweight	
47	unicode_in_text	1
48	whitebalance	0
49	xtoyres	100
50	zeroconfidenceresults	0

Save Properties Close

#	Path	Plate Text	Plate Text (ASCII)	Plate category	Country/State	Confidence	Frame	Character Size	Text Background	Text Color	Dedicated Area Color	ANPR Time(ms)
1	C:/...	PRR619	PRR619	NONE	BEL (123003)	74	((374,115),(508...	28	(255,255,255)	(0,0,0)	(255,255,255)	72

After


ADI Demo 7.4.0.16 (64 bit) [ANPR: cmanpr-7.3.14.13 : eur] - [MMR: OFF] - [License: local]
File Edit Process Help
PRR-619

ANPR Result Tree
- PRR619
  - Text(ASCII): PRR619
  - Plate catego... NONE
  - Confidence: 70
  - Type: HUN (101011)
  - Color(RGB): (0,0,0)
  - BkColor(RGB): (255,255,255)
  - Dedicated A... (255,255,255)
  - Frame: ((371,115),(505,131),(504,167),(369,151))
  - Characters (...
  - Tips

CORRECT

ANPR Settings

Current engine: cmanpr-7.3.14.13:eur

Properties:

	Name	Value
13	general	4
14	heapfreefreq	0
15	local_character_conversi...	2
16	local_character_permuta...	0
17	location	HUN
18	nchar_max	20
19	nchar_min	3

Save Properties Close

#	Path	Plate Text	Plate Text (ASCII)	Plate category	Country/State	Confidence	Frame	Character Size	Text Background	Text Color	Dedicated Area Color	ANPR Time(ms)
1	C:/...	PRR619	PRR619	NONE	HUN (101011)	70	((371,115),(505,131),(504,167),(369,151))	28	(255,255,255)	(0,0,0)	(255,255,255)	69

4. Results Customization

4.1 gaptospace

This parameter is for setting up gaps between the characters read and recognized by Carmen® FreeFlow Image.

The value can be set as follows:

- 0: default, the license plate data is returned in one block with no spaces in the result
- 1: based on the frames of the characters, Carmen® FreeFlow Image calculates the positions of the gaps and, if they are deemed wide enough, replaces them with spaces in the result
- 2: same as value 1, but thanks to the availability of additional parameters, the geometry of the license plate – characters in a horizontal position, coat of arms, hyphens, separators, etc. – is returned more precisely. **This value is only available for engines 7.2.8.6 and higher.**

Additional parameters for value “2:”

symbolnewrow	beginning of a new row
symbollowcharstart	beginning of the small characters
symbollowcharend	end of the small characters
symbolarms	coat of arms
symbolhyphen	hyphen
symbolspace	space
symbolverticalline	vertical line

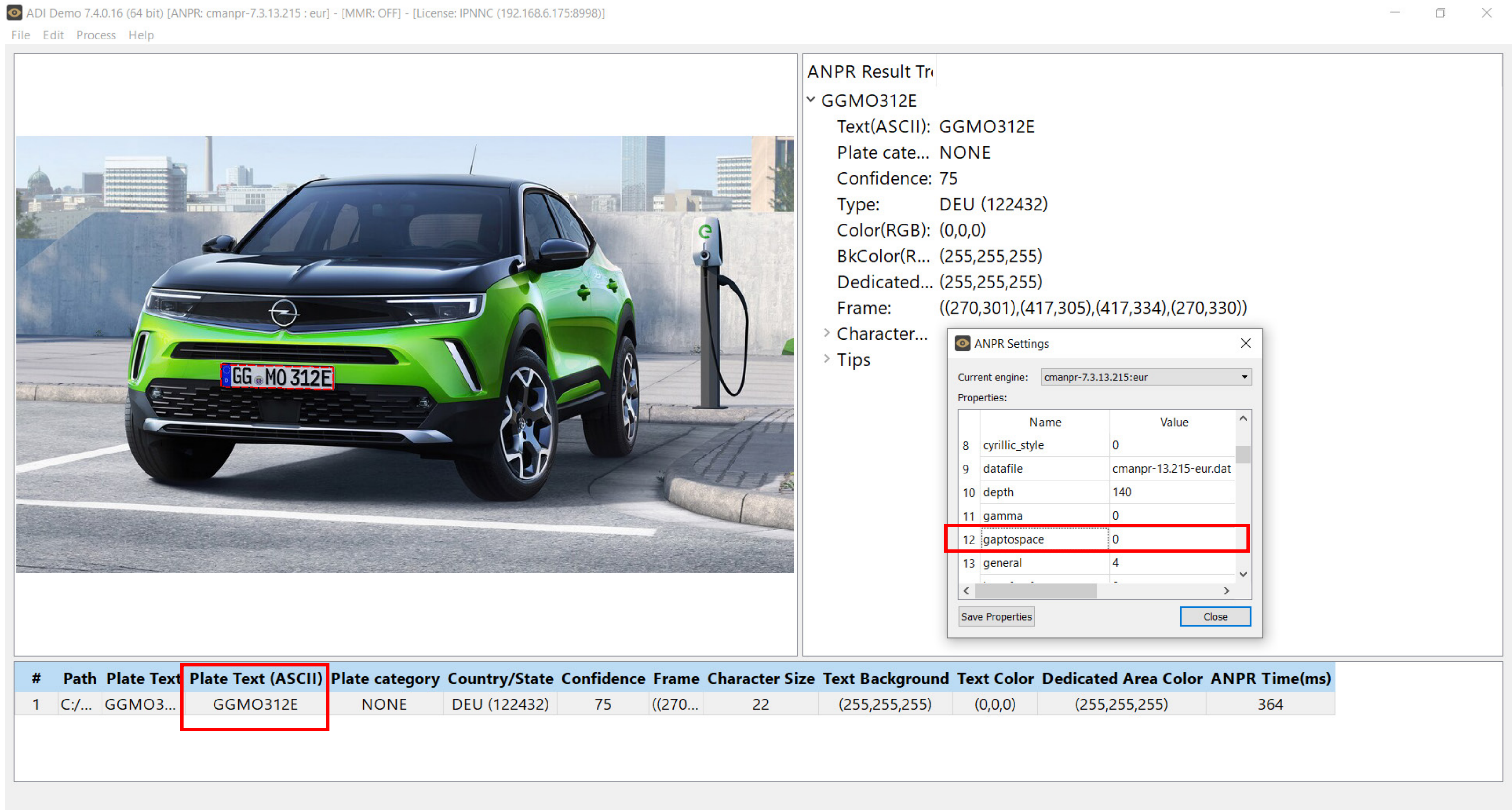
Note that in the case of plates with both Arabic and corresponding Latin characters, value “1” for ‘gaptospace’ won’t work correctly unless ‘local_character_permutation’ (see below) is set to value “3”. If either the Arabic or the Latin character sequences are wanted, we suggest using value “2” for ‘gaptospace.’

4. Customization Results

4.1 gaptospace

The following example shows what big difference setting the ‘gaptospace’ value from “0” to “1” makes with a German license plate.

Value “0”



ADI Demo 7.4.0.16 (64 bit) [ANPR: cmanpr-7.3.13.215 : eur] - [MMR: OFF] - [License: IPNNC (192.168.6.175:8998)]

ANPR Result Tr

GGMO312E

- Text(ASCII): GGMO312E
- Plate cate... NONE
- Confidence: 75
- Type: DEU (122432)
- Color(RGB): (0,0,0)
- BkColor(R... (255,255,255)
- Dedicated... (255,255,255)
- Frame: ((270,301),(417,305),(417,334),(270,330))
- Character...
- Tips

ANPR Settings

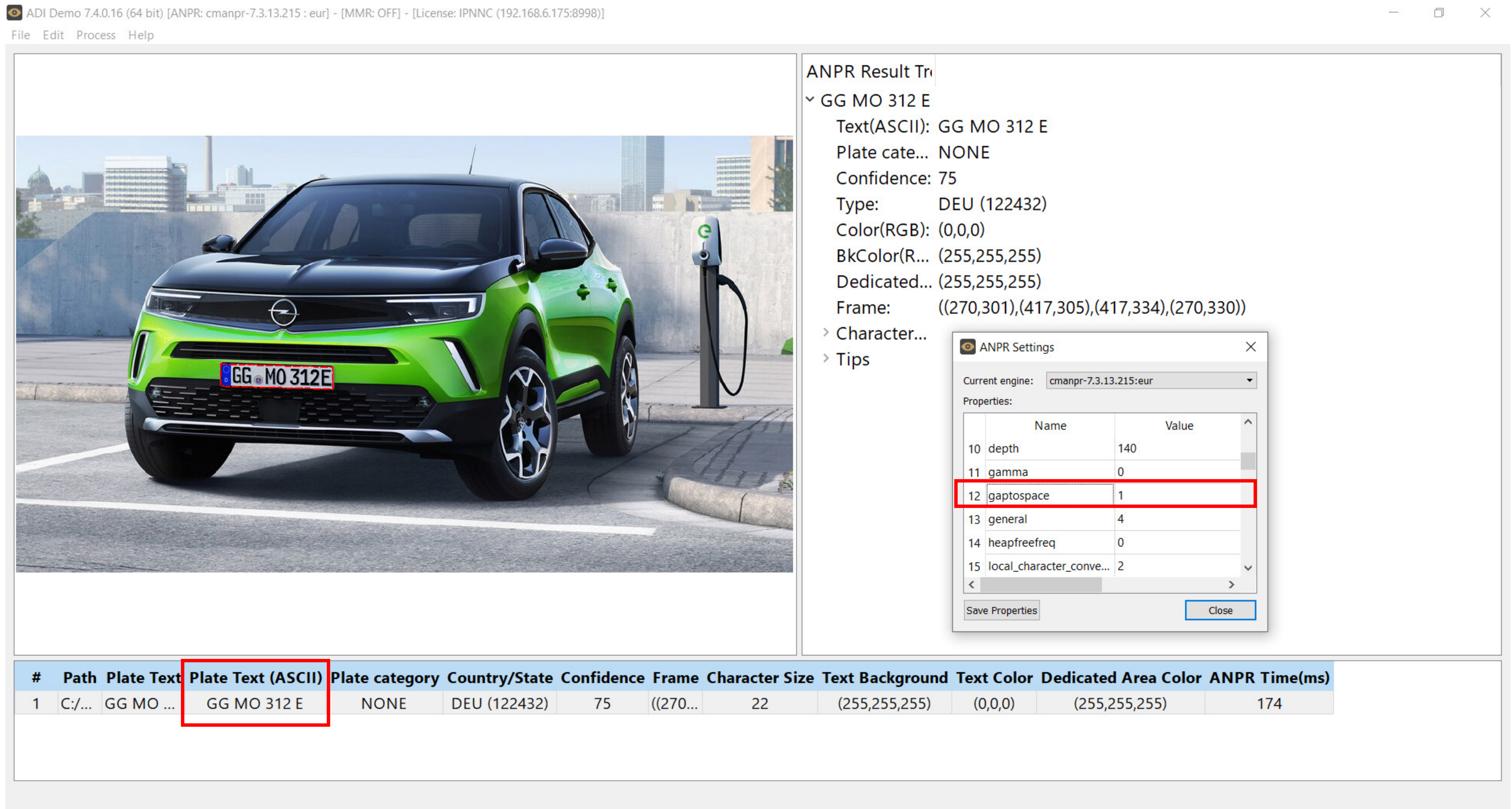
Current engine: cmanpr-7.3.13.215:eur

Properties:

	Name	Value
8	cyrillic_style	0
9	datafile	cmanpr-13.215-eur.dat
10	depth	140
11	gamma	0
12	gaptospace	0
13	general	4

#	Path	Plate Text	Plate Text (ASCII)	Plate category	Country/State	Confidence	Frame	Character Size	Text Background	Text Color	Dedicated Area Color	ANPR Time(ms)
1	C:/...	GGMO3...	GGMO312E	NONE	DEU (122432)	75	((270...	22	(255,255,255)	(0,0,0)	(255,255,255)	364

Value “1”



ADI Demo 7.4.0.16 (64 bit) [ANPR: cmanpr-7.3.13.215 : eur] - [MMR: OFF] - [License: IPNNC (192.168.6.175:8998)]

ANPR Result Tr

GG MO 312 E

- Text(ASCII): GG MO 312 E
- Plate cate... NONE
- Confidence: 75
- Type: DEU (122432)
- Color(RGB): (0,0,0)
- BkColor(R... (255,255,255)
- Dedicated... (255,255,255)
- Frame: ((270,301),(417,305),(417,334),(270,330))
- Character...
- Tips

ANPR Settings

Current engine: cmanpr-7.3.13.215:eur

Properties:

	Name	Value
10	depth	140
11	gamma	0
12	gaptospace	1
13	general	4
14	heapfreefreq	0
15	local_character_conve...	2

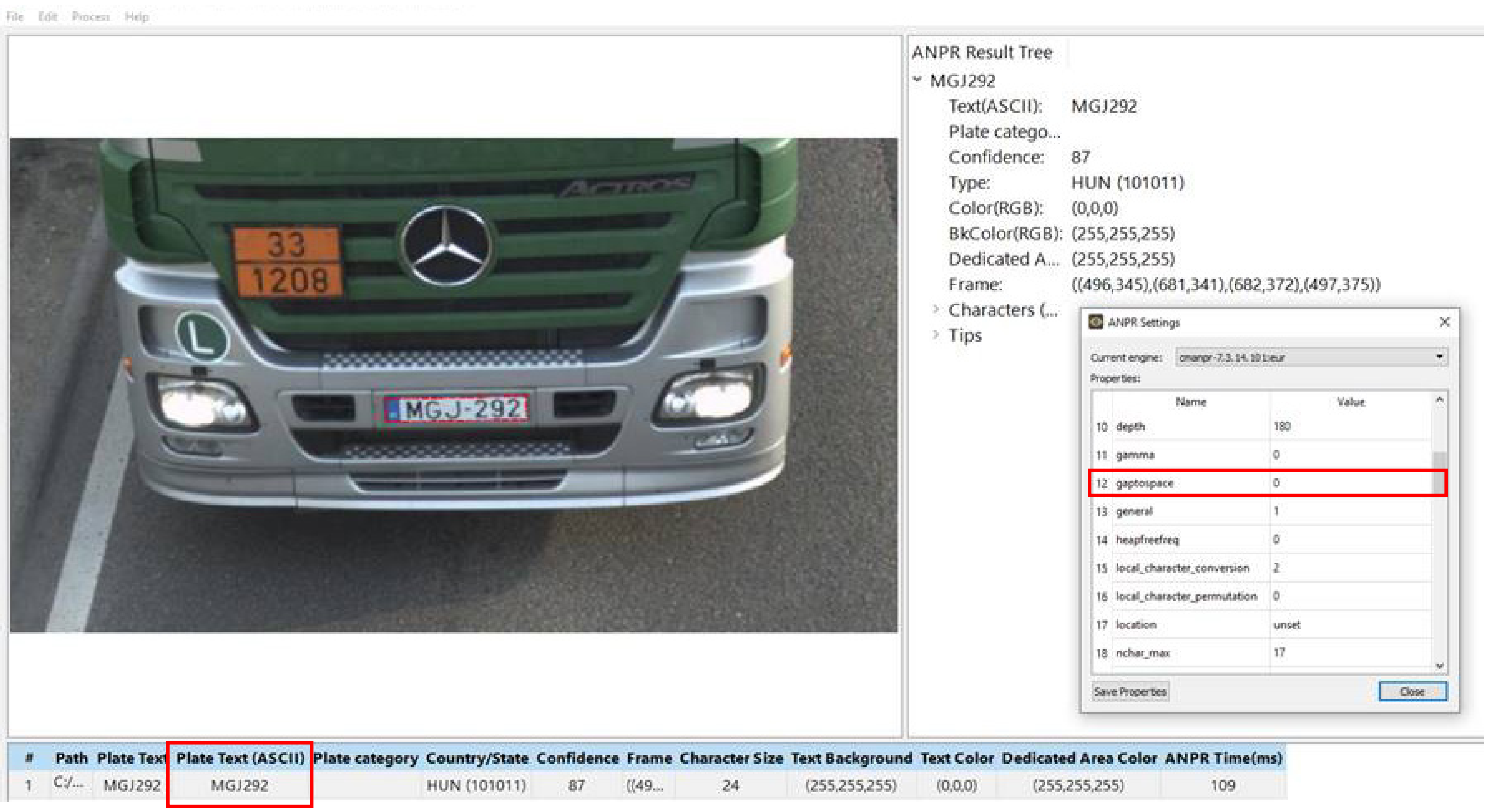
#	Path	Plate Text	Plate Text (ASCII)	Plate category	Country/State	Confidence	Frame	Character Size	Text Background	Text Color	Dedicated Area Color	ANPR Time(ms)
1	C:/...	GG MO ...	GG MO 312 E	NONE	DEU (122432)	75	((270...	22	(255,255,255)	(0,0,0)	(255,255,255)	174

4. Customization Results

4.1 gaptospace

This example shows the difference between ‘gaptospace’ values “0” and “2” on a Hungarian license plate. Note that the ‘symbolhyphen’ value was also changed from the default to add the “-” character between the three letters and three numbers of the plate.

Value “0”



ANPR Result Tree

- MGJ292
  - Text(ASCII): MGJ292
  - Plate catego...
  - Confidence: 87
  - Type: HUN (101011)
  - Color(RGB): (0,0,0)
  - BkColor(RGB): (255,255,255)
  - Dedicated A... (255,255,255)
  - Frame: ((496,345),(681,341),(682,372),(497,375))
  - Characters (...
  - Tips

ANPR Settings

Current engine: cmanpr-7.3.14.101:eur

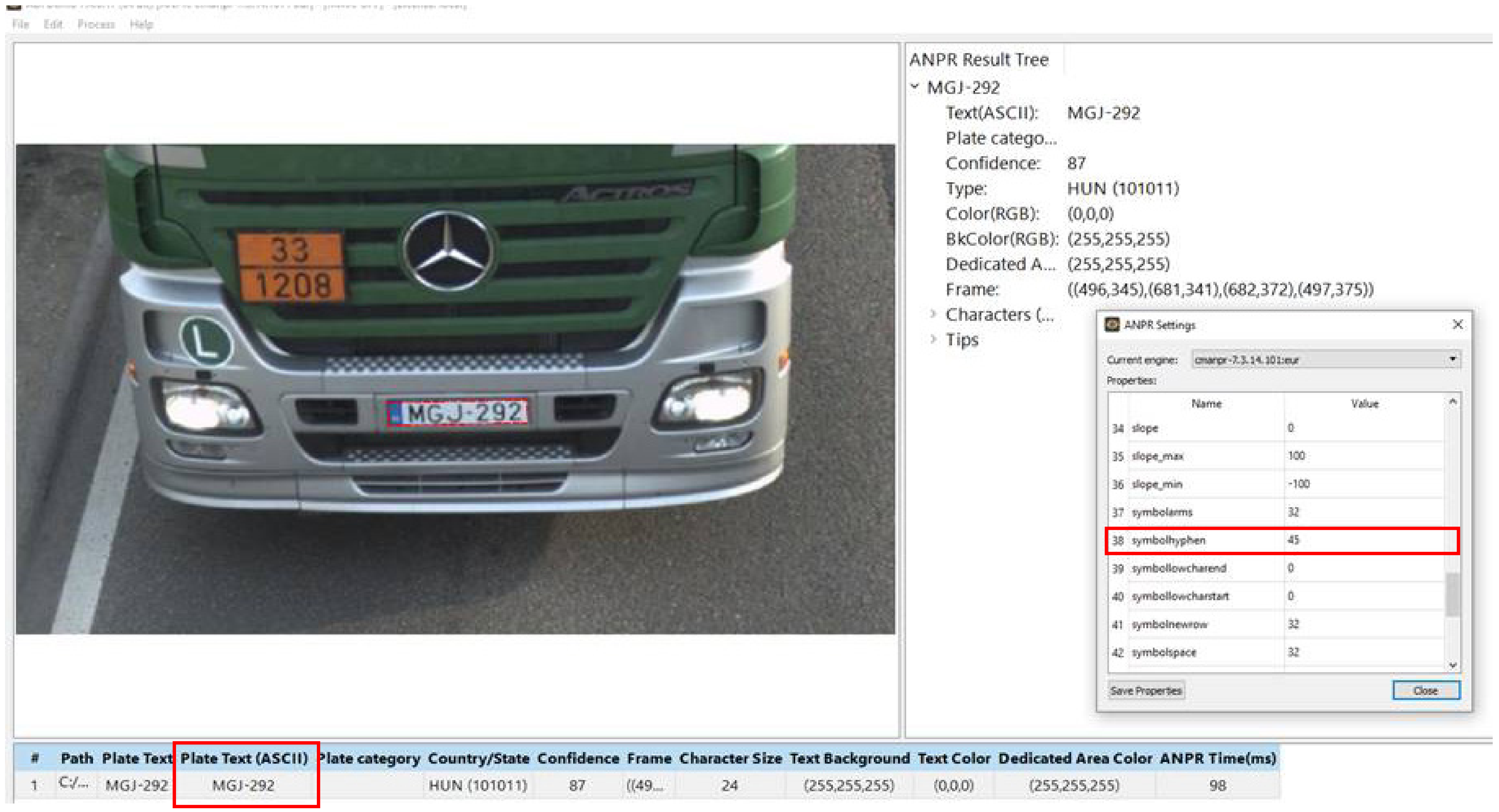
Properties:

	Name	Value
10	depth	180
11	gamma	0
12	gaptospace	0
13	general	1
14	heapfreefreq	0
15	local_character_conversion	2
16	local_character_permutation	0
17	location	unset
18	nchar_max	17

Save Properties · Close

#	Path	Plate Text	Plate Text (ASCII)	Plate category	Country/State	Confidence	Frame	Character Size	Text Background	Text Color	Dedicated Area Color	ANPR Time(ms)
1	C:/...	MGJ292	MGJ292		HUN (101011)	87	((49...	24	(255,255,255)	(0,0,0)	(255,255,255)	109

Value “2”



ANPR Result Tree

- MGJ-292
  - Text(ASCII): MGJ-292
  - Plate catego...
  - Confidence: 87
  - Type: HUN (101011)
  - Color(RGB): (0,0,0)
  - BkColor(RGB): (255,255,255)
  - Dedicated A... (255,255,255)
  - Frame: ((496,345),(681,341),(682,372),(497,375))
  - Characters (...
  - Tips

ANPR Settings

Current engine: cmanpr-7.3.14.101:eur

Properties:

	Name	Value
34	slope	0
35	slope_max	100
36	slope_min	-100
37	symbolarms	32
38	symbolhyphen	45
39	symbollowcharend	0
40	symbollowcharstart	0
41	symbolnewrow	32
42	symbolspace	32

Save Properties · Close

#	Path	Plate Text	Plate Text (ASCII)	Plate category	Country/State	Confidence	Frame	Character Size	Text Background	Text Color	Dedicated Area Color	ANPR Time(ms)
1	C:/...	MGJ-292	MGJ-292		HUN (101011)	87	((49...	24	(255,255,255)	(0,0,0)	(255,255,255)	98

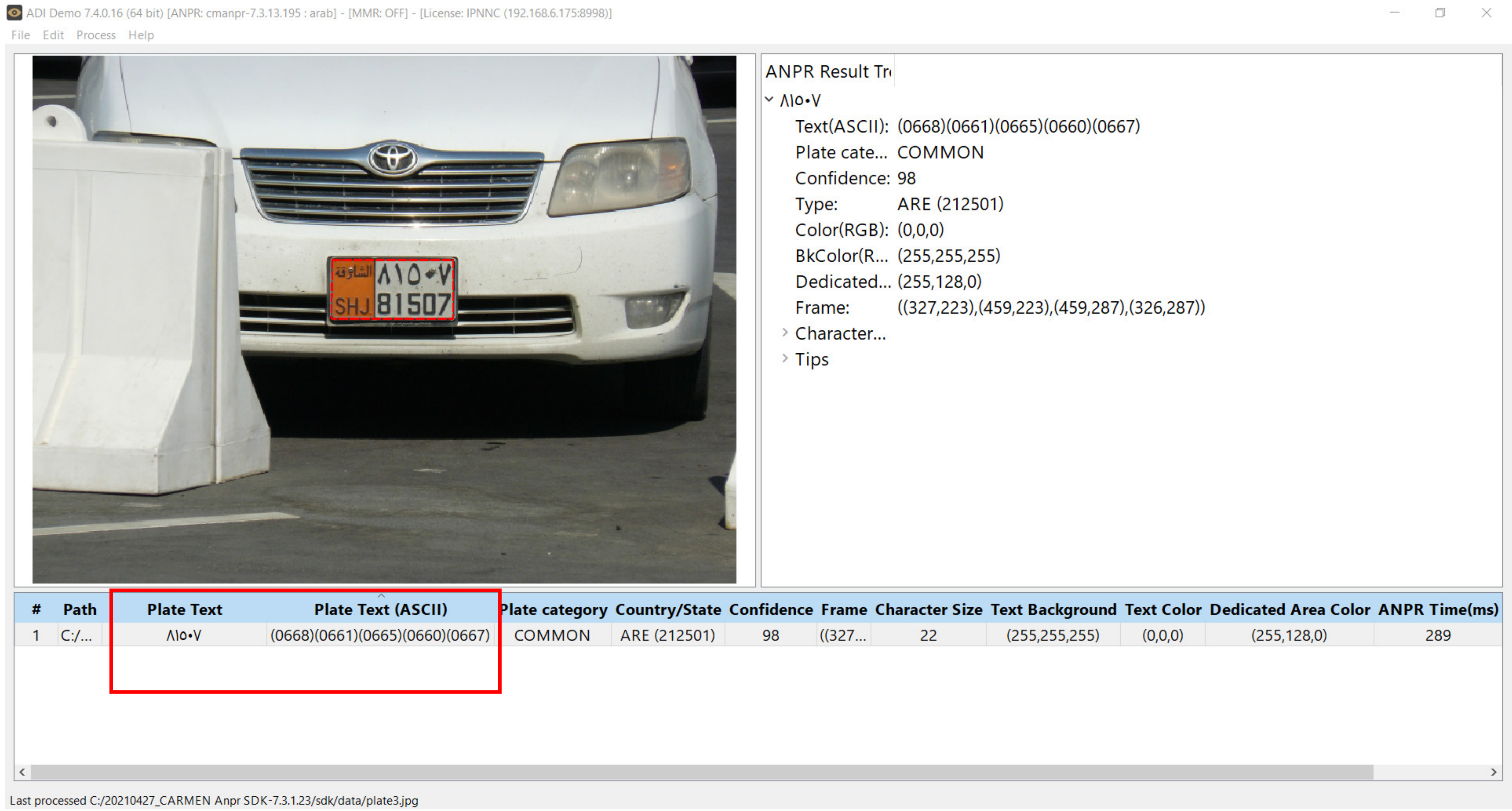
4. Customization Results

4.2 local_character_permutation

This parameter allows you to change the engine’s behavior if the license plate features Arabic character sequences with their corresponding Latin characters. Characters that have no corresponding Latin characters are returned normally.

The value can be set as follows:

0: default, the engine decides which character sequence to return for every Arabic-Latin pair



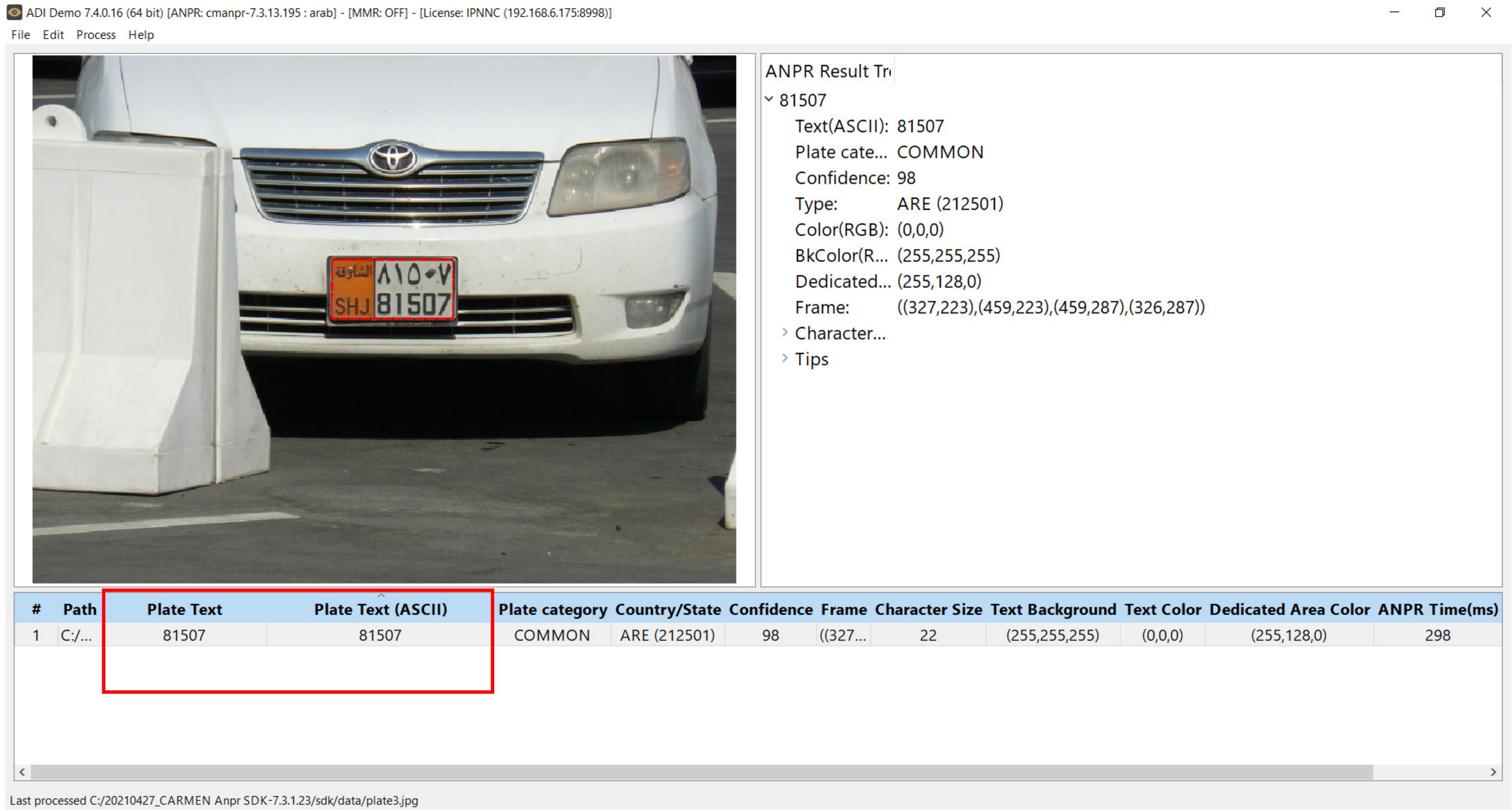
ADI Demo 7.4.0.16 (64 bit) [ANPR: cmanpr-7.3.13.195 : arab] - [MMR: OFF] - [License: IPNNC (192.168.6.175:8998)]

ANPR Result Tr
٨١٥٠٧
Text(ASCII): (0668)(0661)(0665)(0660)(0667)
Plate cate... COMMON
Confidence: 98
Type: ARE (212501)
Color(RGB): (0,0,0)
BkColor(R... (255,255,255)
Dedicated... (255,128,0)
Frame: ((327,223),(459,223),(459,287),(326,287))
Character...
Tips

#	Path	Plate Text	Plate Text (ASCII)	Plate category	Country/State	Confidence	Frame	Character Size	Text Background	Text Color	Dedicated Area Color	ANPR Time(ms)
1	C:/...	٨١٥٠٧	(0668)(0661)(0665)(0660)(0667)	COMMON	ARE (212501)	98	((327...	22	(255,255,255)	(0,0,0)	(255,128,0)	289

Last processed C:/20210427_CARMEN Anpr SDK-7.3.1.23/sdk/data/plate3.jpg

1: Carmen® FreeFlow Image returns the Latin characters from the Arabic-Latin pairs



ADI Demo 7.4.0.16 (64 bit) [ANPR: cmanpr-7.3.13.195 : arab] - [MMR: OFF] - [License: IPNNC (192.168.6.175:8998)]

ANPR Result Tr
81507
Text(ASCII): 81507
Plate cate... COMMON
Confidence: 98
Type: ARE (212501)
Color(RGB): (0,0,0)
BkColor(R... (255,255,255)
Dedicated... (255,128,0)
Frame: ((327,223),(459,223),(459,287),(326,287))
Character...
Tips

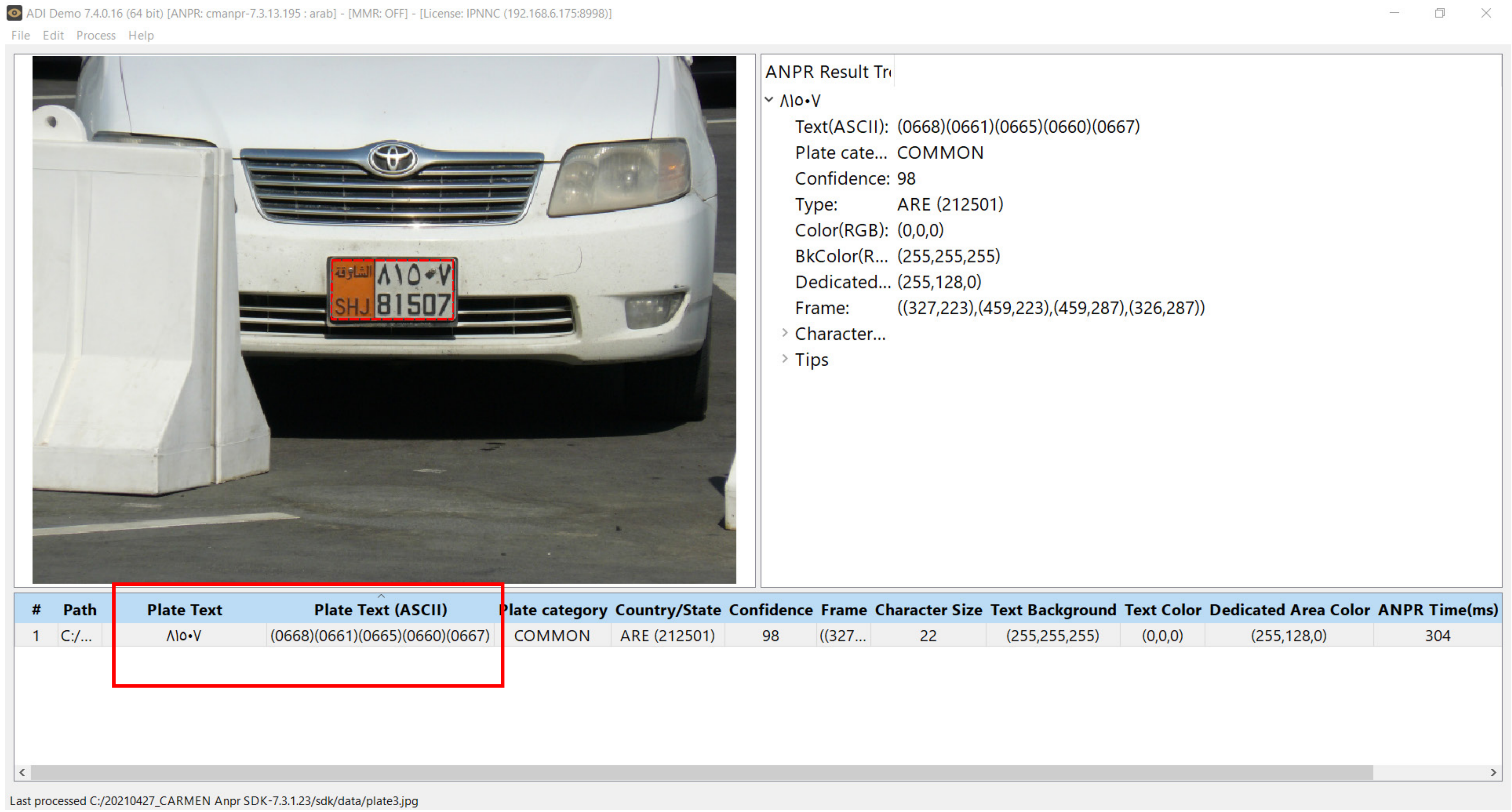
#	Path	Plate Text	Plate Text (ASCII)	Plate category	Country/State	Confidence	Frame	Character Size	Text Background	Text Color	Dedicated Area Color	ANPR Time(ms)
1	C:/...	81507	81507	COMMON	ARE (212501)	98	((327...	22	(255,255,255)	(0,0,0)	(255,128,0)	298

Last processed C:/20210427_CARMEN Anpr SDK-7.3.1.23/sdk/data/plate3.jpg

4. Customization Results

4.2 local_character_permutation

2: Carmen® FreeFlow Image returns the Arabic characters from the Arabic-Latin pairs



ADI Demo 7.4.0.16 (64 bit) [ANPR: cmanpr-7.3.13.195 : arab] - [MMR: OFF] - [License: IPNNC (192.168.6.175:8998)]

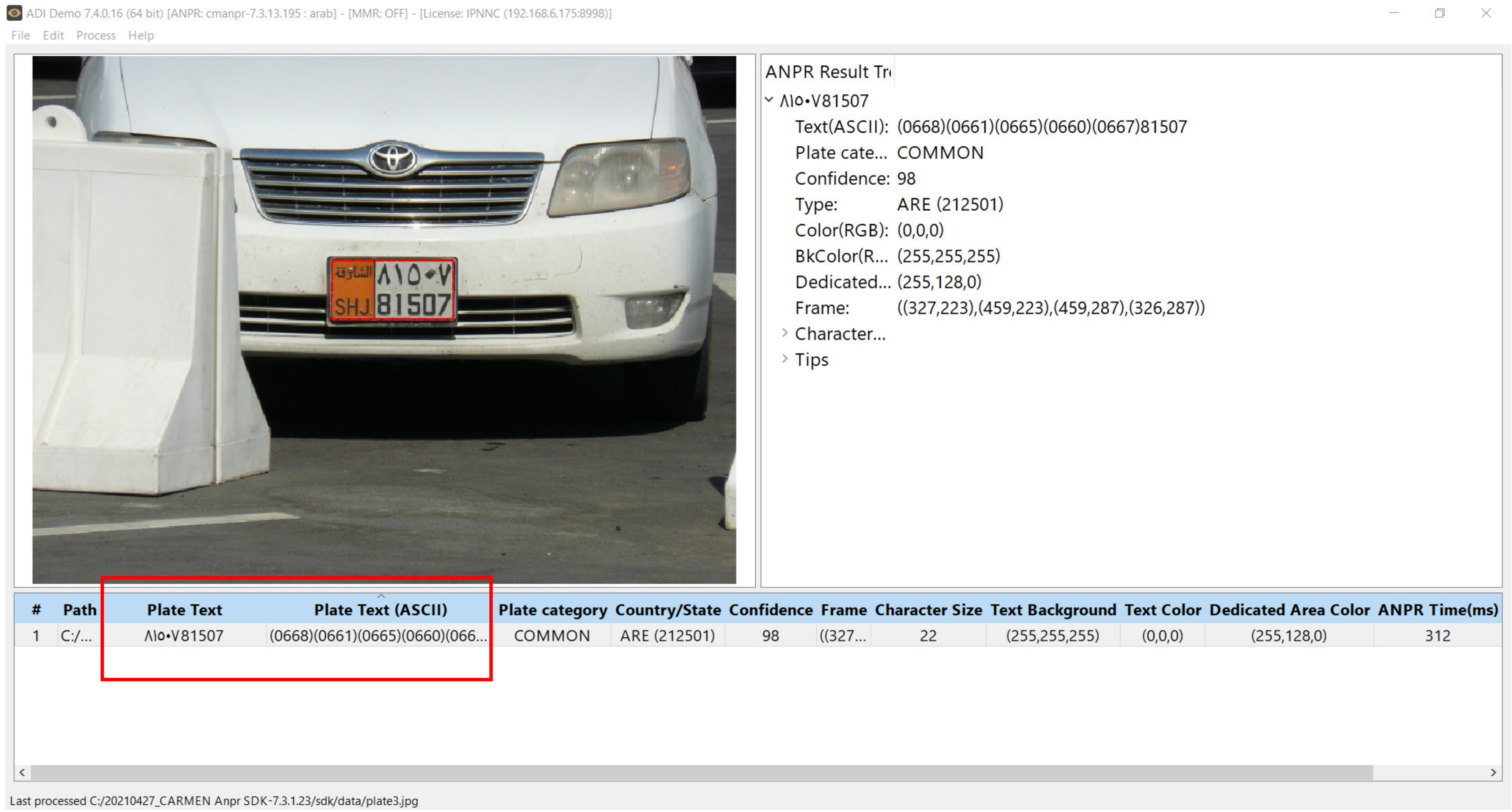
ANPR Result Tr

- ٨١٥٠٧
  - Text(ASCII): (0668)(0661)(0665)(0660)(0667)
  - Plate cate... COMMON
  - Confidence: 98
  - Type: ARE (212501)
  - Color(RGB): (0,0,0)
  - BkColor(R... (255,255,255)
  - Dedicated... (255,128,0)
  - Frame: ((327,223),(459,223),(459,287),(326,287))
  - Character...
  - Tips

#	Path	Plate Text	Plate Text (ASCII)	Plate category	Country/State	Confidence	Frame	Character Size	Text Background	Text Color	Dedicated Area Color	ANPR Time(ms)
1	C:/...	٨١٥٠٧	(0668)(0661)(0665)(0660)(0667)	COMMON	ARE (212501)	98	((327...	22	(255,255,255)	(0,0,0)	(255,128,0)	304

Last processed C:/20210427_CARMEN Anpr SDK-7.3.1.23/sdk/data/plate3.jpg

3: the engine returns both Arabic and Latin characters of each pair



ADI Demo 7.4.0.16 (64 bit) [ANPR: cmanpr-7.3.13.195 : arab] - [MMR: OFF] - [License: IPNNC (192.168.6.175:8998)]

ANPR Result Tr

- ٨١٥٠٧81507
  - Text(ASCII): (0668)(0661)(0665)(0660)(0667)81507
  - Plate cate... COMMON
  - Confidence: 98
  - Type: ARE (212501)
  - Color(RGB): (0,0,0)
  - BkColor(R... (255,255,255)
  - Dedicated... (255,128,0)
  - Frame: ((327,223),(459,223),(459,287),(326,287))
  - Character...
  - Tips

#	Path	Plate Text	Plate Text (ASCII)	Plate category	Country/State	Confidence	Frame	Character Size	Text Background	Text Color	Dedicated Area Color	ANPR Time(ms)
1	C:/...	٨١٥٠٧81507	(0668)(0661)(0665)(0660)(066...	COMMON	ARE (212501)	98	((327...	22	(255,255,255)	(0,0,0)	(255,128,0)	312

Last processed C:/20210427_CARMEN Anpr SDK-7.3.1.23/sdk/data/plate3.jpg

ATTENTION: if this property is set to “0,” “1,” or “2,” please select value “2” instead of “1” for the ‘gaptospace’ parameter.

4. Customization Results

4.3 platetypeinfos

Available from version 7.3.12.193

With this property, you can directly retrieve all sorts of information for the recognized plates or even for an exact plate type.

You have the following options to choose from:

- countryname: e.g. “US-TX” [Texas, USA]
- CountryS: the short, official three-letter ISO code of a country, e.g., “USA”
- CountryL: the full name of a country, i.e., “United States of America”
- StateS: short, three-letter name of the state, i.e., “TEX”
- StateL: the full name of the state, e.g., “Texas”
- category: the license plate category, such as “TAXI.”

To use this property, type either **GetProperty(“platetypeinfos_[chosen option]”)** to get the required information about the last result or **GetProperty(“platetypeinfos_[typevalue]_[chosen option]”)** to see information about an exact type.
Regardless of the command, requests will provide the following results:

- “No type result:” type not found
- “Invalid type:” type does not exist
- “Valid type, no info:” valid type, no available information
- “Valid result:” valid type, information available.

Note that changing the name of the default value is also possible. Just type **SetProperty(“platetypeinfos_[chosen option]_[value’s original name]”, “[value’s new name]”)**. You can also use the same command to change the new value back to default one by one. To set everything back to default, use the **SetProperty(“platetypeinfos” , “default”)** command.

As of Q2 2022, all engines are capable of displaying all known categories. To get a comprehensive list of the categories known by your specific engine,
type **GetProperty (“platetypeinfos_category_all”)**.

In this example, Carmen® FreeFlow Image should return “TAXI” as category.

4. Customization Results

4.3 platetypeinfos

Legacy categories before Q2 2022:

0	– NONE – we don’t know yet
1	– COMMON
2	– CUSTOM
4	– TAXI
8	– DIPLOMATIC
16	– EXPORT
32	– OLD_TIMER
64	– MOTORCYCLE
128	– PUBLIC_TRANSPORT
256	– COMMERCIAL
512	– POLICE
1024	– GOVERNMENT
2048	– PROBATION
4096	– DRIVING-SCHOOL
8192	– TRAILER
16384	– ARMY
32768	– CONSULAR
65536	– EQUIPMENT
131072	– SECURITY_FORCES
262144	– PRIVATE_TRANSPORT
524288	– UNDER_EXPERIMENT
1048576	– EMIRI_GUARDS
2097152	– UN
4194304	– LIMOUSINE
8388608	– TEMP_TRANSIT

The combination of the categories are their sum.

Example: Police Motorcycle_ 576 (512 + 64).
Binary:
Police: 000000001000000 = 64
Motorcycle: 000001000000000 = 512
Police Motorcycle: 000001001000000 = 576

4. Customization Results

4.3 platetypeinfos

New categories available as of Q2 2022 for the NAM and ARAB engines*:

Note that it’s still possible to create combinations of categories, e.g., “police motorcycle.” However, as of Q2 2022, there is no need to make an exact calculation of the categories’ sum to get the desired results. Instead, refer to our dedicated function that provides information about which exact vehicle categories are recognized by the engine in use and the respective codes.

To get the known categories, use the command **GetProperty(“platetypeinfos_category_all”)** in our code. This will provide you the default or combined category’s name and number.

* New categories are continuously uploaded to all Carmen® FreeFlow Image engines.

Find the up-to-date information at https://adaptiverecognition.com/app/uploads/DOC/Software/Carmen/ANPR/carmen_anpr_reference_manual.pdf under PLATETYPEINFOS.

Possible values of the category information:

0	NONE
1	UNKNOWN
2	COMMON
3	CUSTOM
4	TAXI
5	DIPLOMATIC
6	EXPORT
7	OLD_TIMER
8	MOTORCYCLE
9	PUBLIC_TRANSPORT
10	COMMERCIAL
11	POLICE
12	GOVERNMENT
13	PROBATION
14	DRIVING_SCHOOL
15	TRAILER
16	ARMY
17	CONSULAR
18	EQUIPMENT
19	SECURITY_FORCES
20	PRIVATE_TRANSPORT
21	UNDER_EXPERIMENT
22	UN
23	LIMOUSINE
24	TEMP_TRANSIT
25	TEMPORARY
26	VETERAN
27	UNIVERSITY
28	DISABLED
29	TRACTOR

Special, country-specific categories:

201	EMIRI_GUARDS
202	APPORTIONED

USA indigenous people trible categories:

501	ABSENTEE-SHAWNEE_TRIBE_OF_INDIANS_OF_OKLAHOMA
502	APACHE_TRIBE_OF_OKLAHOMA
503	CADDO_NATION_OF_OKLAHOMA
504	CHEROKEE_NATION
505	CHEYENNE-ARAPAHO_TRIBES
506	CHICKASAW_TRIBE_OF_OKLAHOMA
507	CHOCTAW_TRIBE_OF_OKLAHOMA
508	CITIZEN_POTAWATOMI_NATION
509	COMANCHE_NATION
510	DELAWARE_NATION
511	EASTERN_SHAWNEE_TRIBE_OF_OKLAHOMA
512	IOWA_TRIBE_OF_OKLAHOMA
513	KAW_NATION
514	KICKAPOO_TRIBE_OF_OKLAHOMA
515	KIOWA_INDIAN_TRIBE_OF_OKLAHOMA
516	MIAMI_TRIBE_OF_OKLAHOMA
517	MODOC_TRIBE_OF_OKLAHOMA
518	MUSCOGEE_CREEK_NATION
519	OSAGE_NATION
520	OTTAWA_TRIBE_OF_OKLAHOMA
521	OTOE-MISSOURIA_TRIBE_OF_INDIANS
522	PAWNEE_NATION_OF_OKLAHOMA
523	PEORIA_TRIBE_OF_INDIANS_OF_OKLAMOHA
524	PONCA_TRIBE_OF_INDIANS_OF_OKLAMOHA
525	QUAPAW_TRIBE_OF_INDIANS
526	SAC_AND_FOX_NATION
527	SHAWNEE_TRIBE
528	SEMINOLE_NATION_OF_OKLAHOMA
529	SENECA-CAYUGA_TRIBE_OF_OKLAHOMA
530	TONKAWA_TRIBE_OF_OKLAHOMA
531	UNITED_KEETOOWAH_BAND_OF_CHEROKEE_INDIANS_IN_OKLAHOMA
532	WICHITA_AND_AFFILIATED_TRIBES
533	WYANDOTTE_NATION

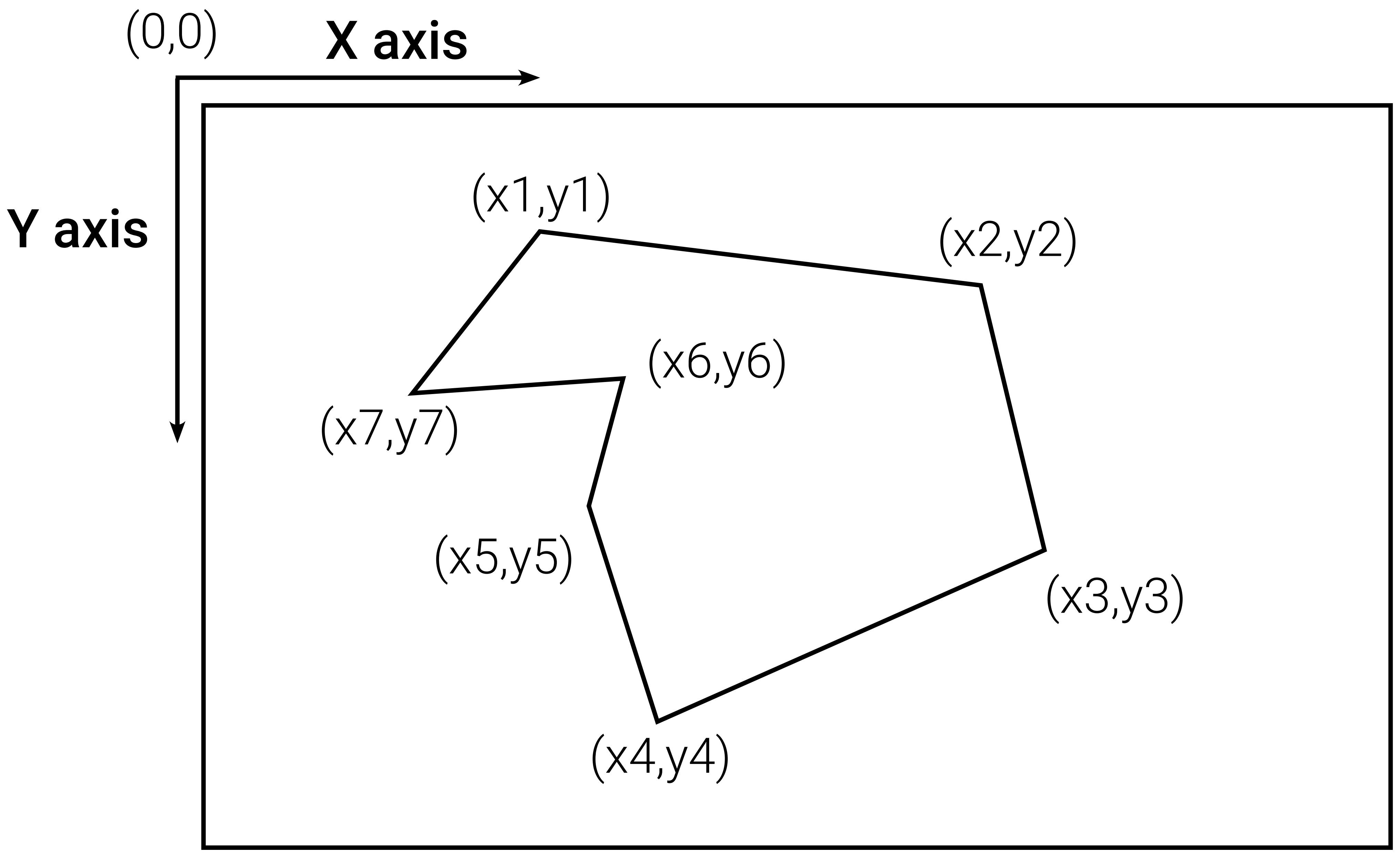
5. Position of License Plate in Input Images

Polygons are calculated as follows:

$\mathbf{P = x_1, y_1; x_2, y_2; x_3, y_3; \ldots; x_i, y_i; \ldots; x_n, y_n}$

In this equation, (x_i, y_i) is a point in the picture in the axis parallel coordinate system. The origo (0,0) is the top left corner. On axis X, the values increase rightward; on axis Y, the values increase downwards. Polygon points are set clockwise.

In this example, you have to set the ROI/ROI as follows for this polygon: 100, 40; 250, 55; 280,130; 130,195;110,110;120,80;80,95;

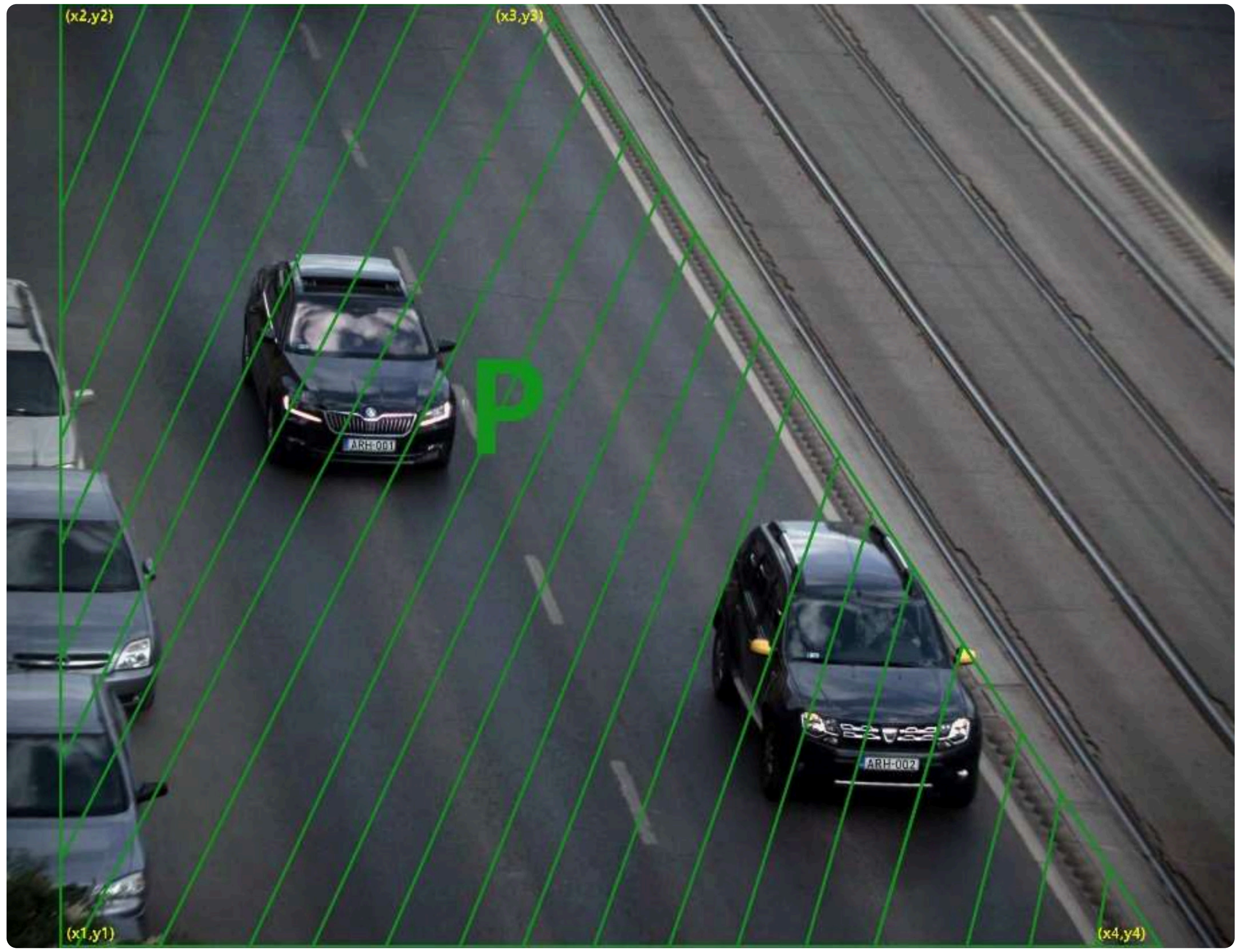


x1 = 100; y1 = 40
x2 = 250; y2 = 55
x3 = 280; y3 = 130
x4 = 130; y4 = 195
x5 = 110; y5 = 110
x6 = 120; y6 = 80
x7 = 80; y7 = 95

5. Position of License Plate in Input Images

5.1 ROI (Region of Interest)

The ‘ROI’ property allows you to create polygon(s) and determine where Carmen® FreeFlow Image should search for license plates. Setting up this value is a must if you want Carmen® FreeFlow Image to look for license plates in a specific area.



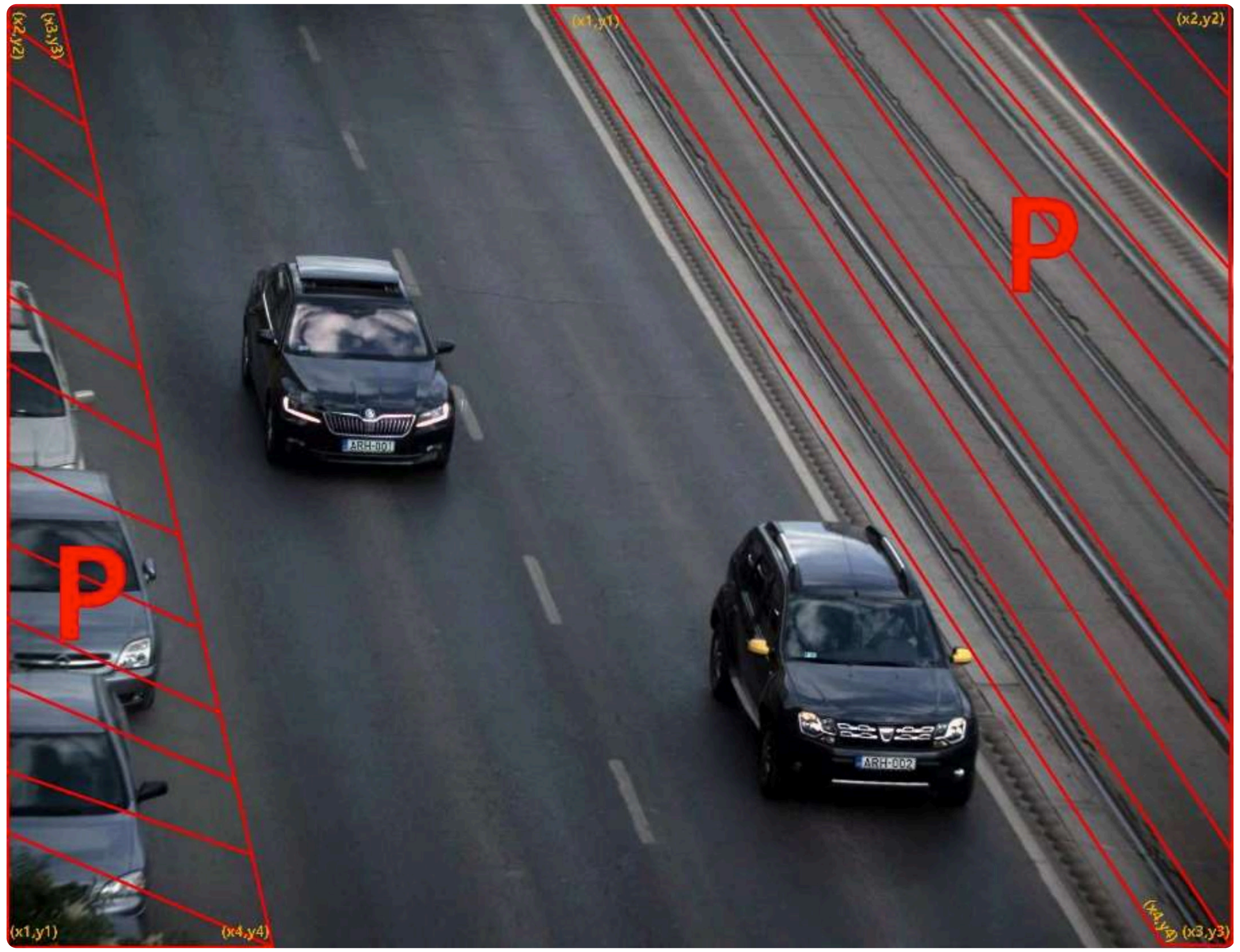
The following commands are available for this property:

- P: the engine searches for license plates inside the given polygon P with previous ‘ROI’ settings deleted
- P1 + P2 + … + P_n: setting up more polygons simultaneously with previous ‘ROI’ settings deleted
- +P: adding a new polygon to the existing one(s) without deleting previous ‘ROI’ settings
- ***del***: deletes previous ‘ROI’ settings.

5. Position of License Plate in Input Images

5.2 ROU (Region of Uninterest)

‘ROU’ is the exact opposite of ‘ROI,’ defining polygon(s) where Carmen® FreeFlow Image will not look for license plates. Setting up this value is a must if you want Carmen® FreeFlow Image to search license plates outside the areas defined in ‘ROU.’



The following commands are available for this property:

- P: the engine searches for license plates inside the given polygon P with previous ‘ROU’ settings deleted
- P1 + P2 + … + P_n: setting up more polygons simultaneously with previous ‘ROU’ settings deleted
- +P: adding a new polygon to the existing one(s) without deleting previous ‘ROU’ settings
- ***del***: deleting previous ‘ROU’ settings.

Contact Information

Our Offices

Adaptive Recognition Nordic

Winghouse, Ørestads Boulevard 73
København 2300, DENMARK
Phone: +45 31 100 300

Adaptive Recognition Hungary

Alkotas utca 41
Budapest 1123, HUNGARY
Phone: +36 1 201 9650

Adaptive Recognition America

28059 US Highway 19 North Suite 203
Clearwater FL 33761, USA
Phone: +1 727 724 4219

Adaptive Recognition Singapore

1 Toa Payoh Lorong 2, #05-05 Braddell House
Singapore 319637, SINGAPORE
Phone: +65 62 705 224

ADAPTIVE RECOGNITION

For general and product inquiries, the simplest way to get in touch with us is by filling in the form on our homepage adaptiverecognition.com. Alternatively, contact us through one of our regional offices.

Disclaimer

The information contained in this brochure is provided as is and without any warranties of any kind, whether expressed or implied, including but not limited to, implied warranties of satisfactory quality, fitness for a particular purpose and/or correctness. The contents of this brochure is for general information purposes only and do not constitute advice. Adaptive Recognition does not represent or warrant that the information and/or specifications contained in this brochure are accurate, complete or current and specifically stipulate that certain scanner details and specifications contained in this brochure may differ in available models. Therefore, Adaptive Recognition makes no warranties or representations regarding the use of the content, details, specifications or information contained in this brochure in terms of their correctness, accuracy, adequacy, usefulness, timeliness, reliability or otherwise, in each case to the fullest extent permitted by law.