

# **CIDAR**

(Container ID Automatic Reader)

## **Functional Specifications**

**Neural Labs s.l.**



# Table of Contents

- 1 GENERAL INFORMATION..... 3**
  - 1.1 SCOPE ..... 3
  - 1.2 RELATED DOCUMENTS ..... 3
  - 1.3 ABBREVIATIONS ..... 3
  - 1.4 DEFINITIONS ..... 3
- 2 HISTORY OF DOCUMENT CHANGES..... 4**
- 3 FUNCTIONAL SPECIFICATIONS ..... 4**

# 1 General Information

## 1.1 Scope

The scope of this document is the functional specifications of a software component for automatic recognition of Identification Codes of containers: CIDAR (Container ID Automatic Reader).

## 1.2 Related documents

- Containers ID.doc

## 1.3 Abbreviations

Abbreviation	Definition
CIDAR	Container ID Automatic Reader
NL	Neural Labs S.L.
OCR	Optical Character Recognition
DLL	Dynamic Link Library
NA	Not Applicable
TBD	To Be Defined

## 1.4 Definitions

**Recognition Engine.** Software component that executes a series of tasks related with the recognition of characters, patterns etc, and it is integrated into an application or program.

**Artefacts.** Distortions or other undesired unusual effects in the audio or video reproduction due to an error in transmission or in signal-processing operation. Artefacts are often generated due to the use of compression algorithms with information loss (e.g. JPEG). Artefacts in video images can manifest as irregular blocks or a mosaic effect known as *aliasing*, of bands or patches of colour.

## 2 History of Document Changes

Version	Date	Change	Reason
1.0	02/12/2008	NA	First Version

## 3 Functional Specifications

Nº	Description
FS001	<p><b>Application Environment.</b></p> <p>The application environment of this recognition engine is the automatic reading of identification codes of cargo containers.</p>
FS002	<p><b>Recognition Codes.</b></p> <p>The only ID codes that the software recognises are the international standard <i>ISO 6346 codes</i>.</p> <p>The identification of containers is achieved using an alphanumeric combination comprised of 11 characters:</p> <ul style="list-style-type: none"><li>• The first three letters identify the owner company and are assigned to the company by the BIC (<i>Bureau International des Containers et du Transport Intermodal</i>).</li><li>• The fourth letter has the following meaning: U identifies cargo containers. J for auxiliary attachable equipment. Z identifies chassis or road transport trailers.</li><li>• Next there are 6 numerical characters.</li><li>• Finally there is a verification digit to assure the correspondence with the previous 10 characters. Also known as the control digit or checksum digit.</li></ul>
FS003	<p><b>Format and Presentation of the Codes to be Recognised</b></p> <p>The codes to be recognised can be:</p>

Nº	Description
	<ul style="list-style-type: none"><li>• Horizontal and all in one line.</li><li>• Horizontal and distributed in two lines (the first line containing the first four letters and the second line containing the remaining seven numbers).</li><li>• Vertical (all characters on a vertical line, to be read from the top downward).</li></ul> <p>The characters of the identification codes can be darker than the background, or vice versa.</p>
<b>FS004</b>	<b>Code Verification using the Control Digit.</b>  Verification is achieved using the final digit of the code, the control digit. The verification number is calculated using the first 10 characters and it is then compared with the final digit, the control digit.  The engine returns the result of this verification (correct or incorrect). If it is incorrect, this means that there is some error in the reading of one or more characters in the code.
<b>FS005</b>	<b>Minimum Resolution Required</b>  The resolution that is required in order to guarantee satisfactory results is a minimum of 30 pixels for the height of the characters.  If this requirement is not satisfied, the correct recognition rate will not be optimal.
<b>FS006</b>	<b>Format of Input Image</b>  The input images can be in colour format or greyscale.  If the image buffer is passed to the engine directly from memory, the image format can be: <ul style="list-style-type: none"><li>• 8 bits - greyscale.</li><li>• 24 bits - colour RGB.</li><li>• 32 bits - colour RGBA.</li></ul> If the image is passed to the engine as an image file (from disk) the image file format can be: <ul style="list-style-type: none"><li>• Standard Bitmap (.BMP).</li><li>• Standard Jpeg (.JPG).</li></ul> It is possible to process an image acquired with only half of the lines (for interleaved cameras capturing in motion). In this case, the resolution would have to be 30 pixels only counting the acquired lines (60 pixels if it had been

N°	Description
	acquired with all the lines).
FS007	<b>Maximum Angle Allowed.</b>  The image should not have any angle either in the horizontal or the vertical axis, or it should at least be negligible (less than de +/- 1°).
FS008	<b>Permitted Perspective.</b>  The code should not have perspective distortion either in the horizontal or vertical axis, or if there is perspective it should be negligible. This means that the camera should be situated perpendicular to the image plane. (FOR CASES WITH KNOWN PERSPECTIVE THE LIBRARY INCLUDES A PRE-PERSPECTIVE CORRECTION TOOL)
FS009	<b>Other Image Requirements.</b>  The code to be recognised should appear completely within the image.  The code to be recognised should be clear and in focus and there should be sufficient contrast with the background.  If the image has been compressed using Standard JPEG, ensure that the image quality is sufficiently clear to avoid artefacts in the image.  If any of these requirements is not satisfied, the recognition rate may not be satisfactory.
FS010	<b>Correct Recognition Rate.</b>  The correct recognition rate will depend greatly on whether the above requirements are satisfied.  If all requirements described above are satisfied, the correct recognition rate should not be less than 95% of the legible codes.  This rate does not include codes with one or more characters totally or partially occluded or codes with one or more damaged characters, dirty or distorted characters.
FS011	<b>Processing Time.</b>  The processing time will depend on: <ul style="list-style-type: none"><li>• The size of the images to be processed.</li><li>• The computing power of the machine in which the software is running (CPU, RAM, etc.)</li><li>• Quality of the images.</li></ul>

Nº	Description
	<ul style="list-style-type: none"><li>• Configuration parameters that have been selected.</li></ul>
<b>FS012</b>	<b>The Results Generated.</b> Once the image is processed, the results that the engine will generate are: <ul style="list-style-type: none"><li>• Code recognised or not recognised.</li><li>• Code verified or not verified.</li><li>• Code in ASCII format.</li><li>• Confidence Factor of the result (in percentage).</li><li>• Confidence Factor for each character in the code (in percentage).</li><li>• Average height of the characters (in pixels).</li><li>• Processing time (in milliseconds).</li></ul>
<b>FS013</b>	<b>Configuration.</b> The following parameters can be configured as required: <ul style="list-style-type: none"><li>• The approximate height of the characters to be read (in pixels).</li><li>• Area or rectangle within the input image when the recognition will take place.</li><li>• Whether the image has been acquired with all of the lines or only half.</li><li>• Whether the codes to be read are horizontal, vertical or can be either of the two.</li><li>• Whether the horizontal codes to be read are always all in one line, always in two lines, or if it could be either.</li><li>• Whether the characters contained within the codes to be read are darker than the background, lighter than the background, or either.</li></ul>
<b>FS014</b>	<b>Error Management.</b> If an error is found during image processing, and error code will be returned indicating the reason for the error.
<b>FS015</b>	<b>Deployment</b> The recognition engine is deployed as a Microsoft Win32 Dynamic Link

Nº	Description
	<p>Library (DLL), or as an .o library for Linux. It is distributed with the files required for integrating the library (the engine) into any development project over Win32 o Linux environment.</p> <p>Also included is an API Reference Manual for the library, as well as sample source code for several different development platforms.</p>
<b>FS016</b>	<p><b>Integration.</b></p> <p>The engine can be easily integrated into applications developed in:</p> <ul style="list-style-type: none"><li>• Microsoft Visual C++</li><li>• Microsoft Visual Basic</li><li>• Microsoft.NET</li><li>• Borland C++ Builder</li><li>• Borland FoxPro</li><li>• Microsoft C#</li><li>• Delphi</li><li>• G++</li></ul>