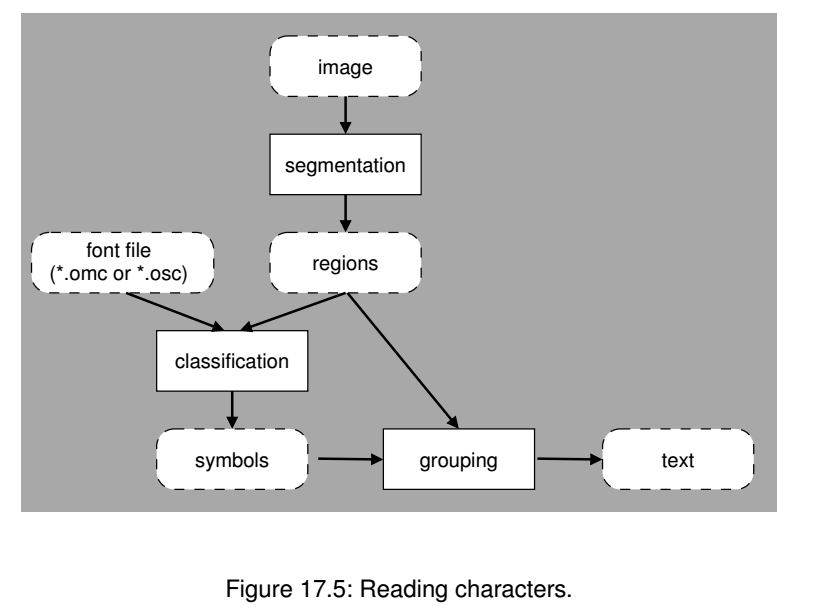
**17.2.6 Segment Image(s)**

For the segmentation, various methods can be used. The Automatic Text Reader is very easy to use  
and provides robust results. It combines the two steps segmentation and classification into one call of  
the operator find\_text. The Manual Text Finder can read dot prints and engraved text, which cannot  
be read by the Automatic Text Reader, but it needs a greater effort in setting the parameters. Therefore,  
the Automatic Text Reader should be used, if possible. Both methods use a text model, which can be  
specified precisely. Two other common segmentation methods are described under ’General CharacterSegmentation’.

Segmentation with the Automatic Text ReaderThe Automatic Text Reader segments and classifies text robustly, typically without the need for extensive parameter tuning. solution\_guide/basics/simple\_reading.hdev (see page 274) and hdevelop/Applications/OCR/bottle.hdev (see page 274) provide a good starting point to become  
familiar with the Automatic Text Reader.  
For using the Automatic Text Reader, a model must be created with create\_text\_model\_reader  
with the parameter Mode set to ’auto’. Here, an OCR classifier must already be passed. Segmentation  
parameters can then be specified with the operator set\_text\_model\_param and can be queried with  
get\_text\_model\_param. After this preparation, the text can be read with the operator find\_text.  
This operator selects character candidates based on region and gray-value features and verifies them  
with the given OCR classifier. The character candidates are then further combined to lines which are  
subsequently tested if they qualify as a text line.  
If the text must match a certain pattern or structure, the parameter ’text\_line\_structure’ of the  
operator set\_text\_model\_param can be set, which determines the structure, i.e., the number of characters for each character block of the text that shall be detected. If the text also includes punctuation  
marks and separators, set the parameters ’return\_punctuation’ and ’return\_separators’ with  
the operator set\_text\_model\_param.  
The Automatic Text Reader assumes approximately horizontal text. If the text is not horizontally aligned,  
the operators text\_line\_orientation and rotate\_image can be used before the use of find\_text.  
The result of find\_text is returned in TextResultID which can be queried with get\_text\_result  
and get\_text\_object, respectively. get\_text\_result returns, e.g., the classification result.  
get\_text\_object returns the iconic result of the Automatic Text Reader, i.e., the character regions. To  
delete the result and the text model, use the operators clear\_text\_result and clear\_text\_model,  
respectively.  
Please refer to the Reference Manual of the above mentioned operators for more information about their  
parameters.

Segmentation with the Manual Text FinderIf dot prints or engraved text must be segmented or if no suitable OCR classifier can be provided, the  
Automatic Text Reader cannot be used. Instead, the Manual Text Finder can be used in these cases.  
In order to segment images robustly, the parameters of the Manual Text Finder should be set carefully. For a practical introduction to the Manual Text Finder, please refer to the HDevelop example  
hdevelop/Applications/OCR/find\_text\_dongle.hdev and the corresponding example description page 278.  
For using the Manual Text Finder, a model must be created with create\_text\_model\_reader with the  
parameter Mode set to ’manual’. Note that no OCR classifier must be passed in this case. Segmentation  
parameters should then be specified with the operator set\_text\_model\_param and can be queried  
with get\_text\_model\_param. After this preparation, the text can be segmented with the operator  
find\_text. This operator selects character candidates based on region and gray-value features. The  
character candidates are then further combined to lines which are subsequently tested if they qualify as  
a text line.  
If the text must match a certain pattern or structure, the parameter ’text\_line\_structure’ of the  
operator set\_text\_model\_param can be set, which determines the structure, i.e., the number of characters for each character block of the text that shall be detected. If the text also includes punctuation  
marks and separators, set the parameters ’return\_punctuation’ and ’return\_separators’ with  
the operator set\_text\_model\_param.  
The Manual Text Finder assumes approximately horizontal text. If the text is not horizontally aligned,  
the operators text\_line\_orientation and rotate\_image can be used before the use of find\_text.  
The result of find\_text is returned in TextResultID which can be queried with get\_text\_result  
and get\_text\_object, respectively. get\_text\_result returns, e.g., the number of found text lines.  
get\_text\_object returns the iconic result of the Automatic Text Reader, i.e., the character regions,  
Owhich can then be classified with a suitable OCR classifier. To delete the result and the text model, use  
the operators clear\_text\_result and clear\_text\_model, respectively.  
Please refer to the Reference Manual of the above mentioned operators for more information about their  
parameters.

General Character SegmentationFor the common character segmentation you can either use the operator segment\_characters to get  
a region containing all character candidates and then apply select\_characters to select those parts  
of the region that are candidates for the individual characters, or you use blob analysis. There, the most  
simple method is the operator threshold, with one or more gray value ranges specifying the regions  
that belong to the foreground objects. Another very common method is dyn\_threshold. Here, a second  
image is passed as a reference. With this approach a local instead of a global threshold is used for each  
position. Further information can be found in the description of this step for blob analysis on page 46.



As a final step it might be necessary to group digits to numbers or characters to words. This can be  
realized with the region processing operators like those described for the method blob analysis on page  
49.