Improve the Barcode Recognition Rate of Your Application

When developing a comprehensive barcode reading application, a common practice to more rapidly and cost-effectively implement it may include the use of barcode reader software development kits (SDK). It may also include built-in or independent tools to test barcodes and implement barcode / data recovery techniques. During the barcode integration process, improving the accuracy of barcode recognition is crucial. How well a barcode can be decoded can go a long way toward reducing support incidents, improving workflow efficiencies, reducing human-error factors, and enhancing robustness of captured data. A sophisticated barcode reading SDK will offer dozens of ways to improve barcode recognition accuracy.

There are common problems a barcode reader SDK might face when ensuring barcode recognition accuracy. Let's dissect some that might be involved during the decoding process. This will help maximize your SDK investment.

General Resolution Concerns

How effectively a barcode is recognized can also be impacted by the quality of the barcode image produced by your chosen barcode generator. Generally, a barcode reader processes the counting of pixels in an area to determine the width and location of a particular bar in a symbol. So, anything that can interfere with this process can impact barcode recognition accuracy.

Usually, a minimum of 200 dpi is needed for acceptable barcode recognition. The higher you can go in resolution the better. Most organizations will seek to balance good barcode recognition with storage savings realized by minimizing resolution. Just be sure not to sacrifice barcode recognition efficiencies for storage efficiencies. Remember that the lower the resolution the more likely the symbol will lack the pixel density needed for good recognition.

Generally, for 1D barcodes, it's understood at least three pixels are needed per smallest bar and gap in a symbol. For 2D barcodes, it's usually around five pixels. Sometimes, despite proper resolution settings, good pixel density just isn't there. This could be for many reasons – poor quality labels or printer, and so on. This is also where the following barcode pre-processing techniques come into play.

Dilation & Erosion

A barcode becomes dilated when it has extra black pixels or eroded when not enough of them. This can skew barcode recognition accuracy results. Let's describe this a bit more. Sometimes, too much ink can end up on a printed barcode label whereby extra black pixels appear in what should be white areas. Imagine, if you will, black paint spray can splatter on a white wall and you might be able to imagine a dilated barcode. On the other hand, sometimes the label could be missing black pixels. Again using your imagination, consider what might happen if tape on top of a barcode label is pulled off. Perhaps some of the black ink goes with it. These are common enough occurrences. But, there are techniques available to fix them. Special preprocessing filters can be used to either fill in missing black pixels (so-called dilation) or remove them (so-called erosion). By applying such preprocessing filters a barcode reader SDK, such as Dynamsoft's Barcode Reader, can improve the ability to decode barcodes for otherwise damaged barcodes.

Before erosion

After erosion

Moreover, sometimes you may have horizontal black lines across vertical lines of a barcode. For such eroded barcodes, developers can opt for a technique to try and remove stray horizontal black lines. Basically, with proper editing tools, you can reduce the vertical height of a barcode to the point it eliminates any stray horizontal lines. This can be effective with 1D barcodes but, not so much with 2D barcodes that are more reliant on higher resolutions.

De-speckle

Similar to the effects of a dilated barcode, specks of noise can impact recognition just the same. Such specks are commonly attributed to anything from dust on the capture device to signal noise or specks

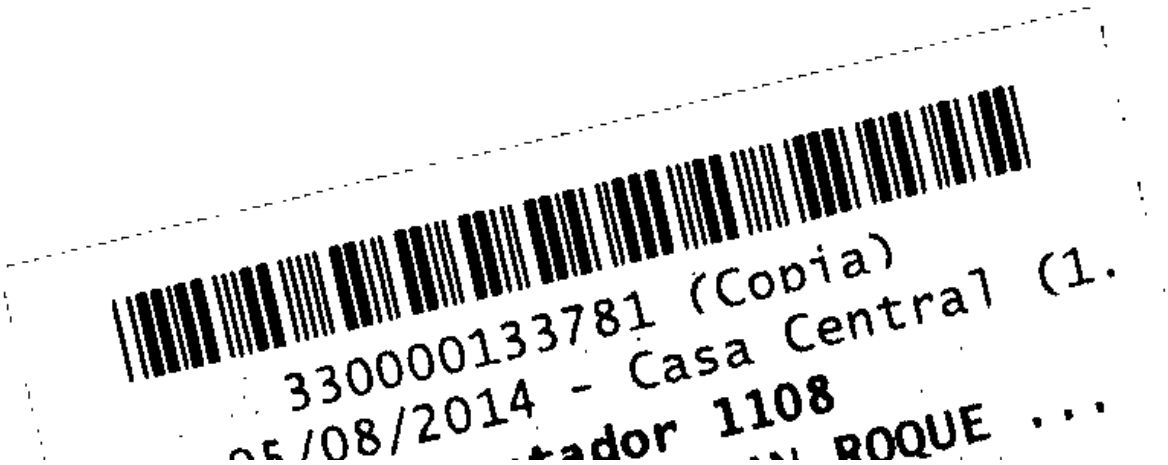
from low-quality resolutions. A de-speckle filter can be employed during pre-processing to automate removal of such unwanted small dots.

Before de-speckle

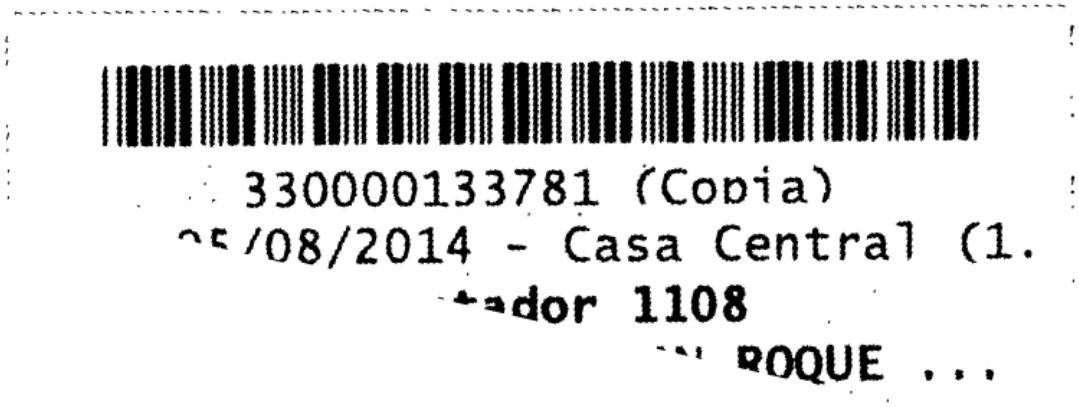
After de-speckle

Deskew

To deskew a barcode means to essentially straighten it out. When a barcode is skewed, depending on the angle and how skewed it is, the individual barcodes can appear thicker or thinner than normal. So, when deskewing occurs, it can actually worsen things. Also, deskewing almost always reduces the image quality, which can result in reduced barcode recognition accuracy. Again, you can use pre-processing features in an SDK like Dynamsoft's Barcode Reader, to auto-deskew barcodes while maximizing their recognition accuracy.



Before deskew



After deskew

Smooth-Zoom

Another similar fix involves smoothing the barcode. This can essentially add resolution and is ideally used with barcodes lacking pixel density. Think of a barcode that is perhaps heavily faded. It would make sense to darken it. But, this isn't just about duplicating pixels to darken it as often this can negatively impact barcode recognition. Smoothing a barcode that is lacking in contrast or pixel density must be done intelligently and an appropriate zoom level should be employed in a barcode reader SDK.

Before smooth-zoom

After smooth-zoom

Binarization

A binarization technique may be necessary if you need to separate what should be foreground black pixels from white background pixels. This is often employed to more easily read a barcode in a document that, for example, perhaps has little contrast. It can also be employed on barcode images captured from low quality devices, such as a smartphone camera or fax machine. Poor quality imaging devices can result in captured barcodes that are unevenly illuminated, blurry or geometrically transformed. A sophisticated binarization technique can be employed via Dynamsoft's Barcode Reader to segment pixels in gray-scale images into two levels (black and white) maximizing the contrast. This is to improve or recover barcode recognition or lost data from poorly captured barcodes.

Before binarization

After binarization

Quiet Zones

Employing all these tools and techniques would somehow be incomplete if you also didn't ensure a proper quiet zone. If you're not already aware, a quite zone is a blank area or margin on either end of a bar code. That blank area tells a barcode scanner where the barcode starts and stops. Quiet zone specifications vary depending on the barcode symbol being used. Usually, at least an eighth of an inch is a minimal requirement. A barcode reader relies on this "empty" information to pick up the relevant area for barcode decoding.

Conclusion

Improving barcode recognition precision involves a lot of considerations and can mean employing appropriate tools and techniques for improvements. We've covered how dilated and eroded barcodes and speckling can impact barcode recognition performance. Remember that skewed barcodes can also reduce recognition. We touched on how the implementation of smoothing techniques for barcodes that lack resolution or contrast can help developers recover otherwise lost data from a barcode. Also, binarization of barcodes is a more frequent occurrence nowadays. This is because of image capturing using low-quality capture devices like smartphone cameras or fax machines. Finally, there's the ever important role that quiet zones play.

These are just a snapshot of the many dynamics developers face when needing to ensure barcode recognition accuracy. The good news is you need not go it alone. There are pre-made barcode reader SDKs that can help you more quickly develop your barcode reading software. Often, such tools have built-in recovery and testing techniques. Other times other third party tools may be needed to help you thoroughly test your barcodes.

RESOURCES / CREDITS:

Notable Solutions
IEEE
Nano Scientific Research Center
Pegasus Imaging