EAN-13 Barcode Recognition: A Comparative Approach

Geometric Decoding vs. Optical Character Recognition (OCR)

Moldovan Darius-Andrei

Technical University of Cluj-Napoca

December 3, 2025

Outline

What is an EAN-13 Barcode?

- **EAN-13** (European Article Number) consists of 13 digits.
- Structure:
  - Left Guard (101)
  - 6 Left-side digits
  - Middle Guard (01010)
  - 6 Right-side digits
  - Right Guard (101)
- **1 = Bar, 0 = Space**
- **Unique Feature:** The first digit is *implicit*, encoded in the parity pattern of the left-side digits.

EAN−13: 8004020931773

Project Objective: Dual Approach

Goal

To develop and compare two distinct methods for recognizing EAN-13 barcodes from images.

1. **Method A: Geometric Pipeline (Implemented)**
   - Uses classical computer vision (Edge Detection, Contours).
   - Decodes the binary bar widths directly (Scanline Analysis).
2. **Method B: OCR Pipeline (Future Work)**
   - Detects the text numbers below the bars.
   - Uses Tesseract or Deep Learning OCR engines.

Why two methods? To evaluate robustness speed vs. accuracy in challenging conditions (blur, curvature, lighting).

Pipeline Overview

The primary implemented method follows a strict classical processing chain:

1. **Preprocessing:** Grayscale conversion, CLAHE, Gaussian Blur.
2. **Edge Detection:** Sobel gradients to find vertical bar structures.
3. **Localization:** Hough Transform (Rotation) + Contour Analysis (Bounding Box).
4. **Extraction:** Perspective warping to obtain a flat, rectangular crop.
5. **Decoding Algorithm:** Converting pixels into numbers.

Step 1: The Scanline & Binarization

How do we translate an image crop into data?

- **Scanlines:** We extract a single row of pixels (intensity profile) from the center of the cropped barcode.
- **Robustness:** If the center fails, we scan offsets (± 10 pixels) to avoid defects or glare.
- **Adaptive Binarization:** The line is converted to 0s (Spaces) and 1s (Bars) based on local intensity thresholds.

Pixel Stream: $[255, 255, 0, 0, 0, 255, 255, 0, 255] \rightarrow [0, 0, 1, 1, 1, 0, 0, 1, 0]$

Step 2: Run-Length Encoding (RLE)

We do not process individual pixels; we process **relative widths**.

- We count consecutive pixels of the same color.
- **Example:**
  - *Binary:* 0000 11 000 11111 00
  - *RLE:* [4, 2, 3, 5, 2]
- This creates a sequence of "Widths" representing Bars and Spaces.
- **Finding the Start:** We look for the Guard Pattern (1-1-1 ratio) to identify the beginning of the data.

Step 3: Normalization (The Core Logic)

A digit in EAN-13 always consists of **4 elements** (Space-Bar-Space-Bar) and has a total width of **7 modules**.

The Normalization Formula

To handle zoom and resolution differences, we normalize the raw RLE values (P_i) to the standard module width:

$$M_i = \text{round}\left(\frac{P_i}{\sum_{j=1}^{4} P_j} \times 7\right)$$

Example

Raw RLE for one digit: $[12px, 8px, 4px, 4px]$ (Sum $=$ 28px).
Normalized: $\frac{12}{28} \times 7 \approx 3$, $\frac{8}{28} \times 7 \approx 2$, ...
Result Pattern: 3-2-1-1 $\rightarrow$ Lookup Table $\rightarrow$ Digit **0**.

Step 4: Decoding and Parity

Left Side Encoding:

- Digits 0-9 have two encodings: **L-code** (Odd parity) and **G-code** (Even parity).
- The scanner detects which code was used (L or G).

Right Side: Always uses R-codes.

The 13th Digit (First Digit):

- The first digit is *not* printed as bars.
- It is determined by the sequence of L and G codes found in the first 6 digits.
- E.g., $LLGLGG \rightarrow 1$.

The OCR Approach

Instead of analyzing bar widths, we analyze the human-readable text below the bars.

Pipeline Plan

1. **Text Region Proposal:** Use morphological operations (dilation) to merge text characters into a solid block.
2. **Extraction:** Crop the bottom 20% of the detected barcode area.
3. **Recognition:** Feed the crop into Tesseract LSTM or a CNN trained on digits (MNIST).

Expected Comparison

Feature	Method A (Geometric)	Method B (OCR)
Speed	Extremely Fast (< 10ms)	Slower (NN inference)
Blur	Fails if edges merge	Often robust (context)
Lighting	Needs contrast	Needs contrast
Curvature	Complex to de-warp	Robust (deformable models)
Damage	Redundancy helps	Fails if text is scratched

Table: Hypothetical performance matrix

The final project will perform benchmarks on a dataset of real-world images to validate these hypotheses.