# Case Xerox

## Xerox JBIG2 Compression Bug: A Real Industry Failure

- In the early 2010s, users of certain Xerox multifunction printers and scanners began noticing a serious anomaly in scanned documents. While the scanned pages *looked visually correct*, the underlying digital text layer was wrong. Numbers and characters were silently substituted, for example a “6” appearing as an “8” or repeated words being replaced by a different word elsewhere in the document. These errors were not random noise; they were systematic and reproducible, making the issue especially dangerous for legal, financial, and archival documents.
- The root cause was traced to Xerox’s implementation of the JBIG2 compression standard, a lossy compression method designed for black-and-white (bi-level) documents such as scanned text and forms. JBIG2 works by detecting similar-looking shapes (glyphs), storing one prototype, and reusing it wherever similar patterns appear. This approach achieves very high compression ratios. However, in Xerox devices, the pattern-matching logic occasionally treated *different characters with similar visual structure* as identical. During decompression, the wrong glyph was substituted, effectively changing the document’s meaning.
- What made this failure particularly insidious was that it bypassed human detection. The visual appearance of the scanned image was usually acceptable to the human eye, especially at printer or screen resolution. Humans rely on context and semantics while reading, so subtle character-level substitutions often went unnoticed. Machines, however, relied on the corrupted text layer for search, copy-paste, indexing, and downstream processing. As a result, documents could be archived, transmitted, or legally relied upon with incorrect content, without any visible warning.

- This incident is a rare but powerful real-world example of compression logic breaking perception and semantics, not just image quality. It demonstrates that lossy compression, when applied inappropriately, can alter meaning rather than merely reduce fidelity. For computer vision and AI systems, the lesson is clear: techniques optimized for human perception may silently destroy information that machines depend on. The Xerox JBIG2 bug remains a cautionary tale about trusting compressed visual data in high-stakes pipelines.

- Why did the Xerox JBIG2 bug remain undetected by human users for a long time, and why was this more dangerous than obvious visual artifacts?
- How is this incident an example of psycho-visual redundancy helping humans but harming machine-based interpretation?
- If a modern OCR or vision model were trained on JBIG2-corrupted data, what kinds of failures or biases might you expect in the model’s output?

## Task:1 Pattern Substitution Risk

**Problem statement**

You are given a binary (black-and-white) document image containing repeated characters. Write a program that:

- Extracts connected components (individual symbols).
- Measures similarity between components using a simple shape descriptor (e.g., bounding box size, pixel overlap, or Hu moments).
- Groups components that are “similar enough” using a threshold.
- Reconstructs the image by replacing all components in a group with a single prototype.

**Tasks**

- Implement the grouping and replacement logic.
- Vary the similarity threshold.
- Observe when *different characters* start getting merged.

## Task 2: Human-Visible vs Machine-Relevant Differences

- **Problem statement**
  Take a grayscale image with text or fine patterns. Generate multiple compressed versions using JPEG at different quality levels.

  **Tasks**
- Compute PSNR and SSIM between the original and compressed images.
- Apply a simple edge detector or OCR preprocessing step on each version.
- Compare how perceptual metrics (PSNR/SSIM) and algorithmic outputs change with compression.

## Task 3: Silent Data Corruption Detection

**Problem statement**

You are given two scanned versions of the same document:

- One compressed using a safe lossless method
- One compressed using a lossy method
- Both *look visually similar*.

**Tasks**

- Extract connected components or contours from both images.
- Quantify structural differences using shape descriptors or pixel-wise comparison.
- Flag regions where substitution or distortion may have occurred.

## Task 4: When Compression Breaks a Downstream Task

**Problem statement**

Build a simple rule-based digit or character recognizer (not deep learning).
Test it on:

- Original document images
- Heavily compressed versions

**Tasks**

- Measure recognition accuracy on both sets.
- Identify which characters fail first and why.
- Relate failures to compression artifacts or pattern substitution.

## Task 5: Designing a “Safe Compression” Rule

**Problem statement**

Design a simple heuristic to decide whether a document image should be:

- Compressed losslessly
- Compressed lossily
- Not compressed at all

**Tasks**

- Use entropy, edge density, or connected-component count as signals.
- Implement the decision logic.
- Test it on different image types (text, forms, photos).