

Final Report – CERN Summer Student Programme

Cerabyte–CERN Collaboration: Optical Data Storage and QR Code Decoding

Mostafa Solitan
IT Storage Group, CERN

Abstract

Cerabyte is developing ultra-long-term optical data storage on ceramic-coated glass media. CERN’s scientific programme generates exabytes of data, requiring archival systems that are not only scalable but also sustainable for centuries. Cerabyte’s approach encodes data as matrices of QR codes written by femtosecond-laser (fs) ablation into durable ceramic layers and read back by high-speed optical microscopy. The writer and reader are therefore *different* technologies (fs-laser vs microscope), and unlike magnetic tape the ceramic medium is *write-once*, not erasable.

In this project, multiple approaches were investigated to achieve industrial-scale decoding throughput (on the order of 500 FPS, corresponding to 10.000 – 30.000 QR codes/s). Initial experiments with deep learning detectors (e.g., YOLOv8) showed limitations in both accuracy and throughput for dense layouts. The focus therefore shifted to classical computer vision to localize 4×4 QR matrices and extract per-code crops. Several decoding libraries were benchmarked; OpenCV’s WeChat QR decoder emerged as the most effective balance of robustness and speed. Through multithreaded CPU implementation, the pipeline achieved sub-5 ms per QR and exceeded 75% decoding accuracy on real microscope samples despite imperfections in the medium. Further acceleration using GPU inference with OpenCV DNN increased throughput to approximately 18.000 QR/s. A hybrid approach that combines classical computer vision with lightweight deep learning refinements outperformed either method alone.

The contributions include: (i) a repeatable imaging and decoding workflow for prototype carriers, (ii) identification of limitations of deep learning for dense QR scenarios, (iii) a scalable computer-vision-based solution with validated benchmarks, and (iv) a roadmap toward a fully GPU-first, production-grade pipeline.

Acknowledgments

I am grateful to my CERN supervisors Vladimir Bahyl and Michael Davis, and to the Cerabyte team—especially Sebastian Kirsch, Lukas Kreuziger, and Achim Oehler—for guidance and support.

Contents

1 Introduction

3

1.1	CERN Data Storage Landscape and Market Risk	3
1.2	Motivation for Novel Storage Media	3
1.3	Cerabyte’s Approach: Writer vs Reader	3
1.4	Collaboration Objectives	3
2	Technical Background	3
2.1	Optical Data Storage Principles	3
2.2	Acquisition Pipeline Overview	4
2.3	Microscope Acquisition Details	4
2.4	QR Codes, Redundancy, and Locality	4
3	Methods and Work Performed	4
3.1	Setup (Hardware and Software)	4
3.2	Image Processing Pipeline	4
3.3	Decoder Libraries and Differences	5
3.4	Parallelization and Acceleration	5
3.5	Experiments Conducted	5
4	Results	6
4.1	Decoding Success Rates	6
4.2	Performance Benchmarks	6
4.3	Illustrative Figures	6
5	Discussion	8
5.1	Strengths and Limitations of the Pipeline	8
5.2	What Makes WeChat More Tolerant	8
5.3	Redundancy Locality and Resilience	8
5.4	On Longevity vs Readability	8
6	Conclusion	9
7	Future Work	9
8	Lessons Learned	9
8.1	Technical	9
8.2	Collaboration	9
8.3	Personal Growth	10
	Appendix: Code Snippets and Benchmark Tables	10
	References	11

1 Introduction

1.1 CERN Data Storage Landscape and Market Risk

CERN operates one of the world’s largest scientific data infrastructures. Experiments at the Large Hadron Collider (LHC) generate hundreds of petabytes per year that must be preserved for decades. The CERN Tape Archive (CTA) provides cost-effective capacity; however, like all conventional media it requires periodic migration and ongoing resources. A further motivation is **market concentration risk** in the tape ecosystem: there is effectively a single supplier of tape-head technology, which represents a strategic risk. Exploring alternative, complementary archival media can reduce exposure to such consolidation.

1.2 Motivation for Novel Storage Media

As global data volumes grow, traditional technologies face limits in sustainability and lifetime. Archival data needs solutions with:

- **Longevity**: preservation for centuries without frequent migrations,
- **Low power**: no energy to preserve bits at rest,
- **High density**: exascale-capable capacity.

1.3 Cerabyte’s Approach: Writer vs Reader

Cerabyte proposes a new optical medium: a thin ceramic layer on glass. The **writer** is a femtosecond laser that physically ablates nanoscale features; the **reader** is a high-speed microscope (bright-field CMOS). This separation is a key difference from tape. The medium is *write-once* (WORM), whereas tapes can be erased and reused. For prototyping, data is structured as matrices of QR codes that incorporate Reed–Solomon redundancy.

1.4 Collaboration Objectives

The CERN–Cerabyte collaboration aims to evaluate feasibility of *fast and accurate* decoding from ceramic-on-glass carriers at scale. Targets considered were on the order of 500 FPS, mapping to 10.000 —30.000 QR/s. This requires advances in image acquisition, detection, decoding, and GPU acceleration.

2 Technical Background

2.1 Optical Data Storage Principles

Information is encoded as physical patterns written and read with light. In the prototype, fs-laser pulses ablate the ceramic layer (hole/no-hole). These marks are permanent and read by high-speed microscopy. Long-term *longevity* derives from the material’s physical stability; dust and ablation defects affect *readability* in the short term, but not the intrinsic stability of the medium.

2.2 Acquisition Pipeline Overview

The acquisition and decoding pipeline used in this work is:

1. **Acquire:** capture microscope frames of the carrier surface.
2. **Detect Squares / Matrices:** identify 4×4 QR matrices via geometric constraints.
3. **Preprocess:** grayscale, thresholding (e.g., Otsu), morphological closing, perspective correction.
4. **Crop Cells:** warp and extract per-QR crops from each matrix.
5. **Decode:** run a QR decoder (QUIRC, ZBar, OpenCV, WeChat).
6. **Reconstruct:** assemble decoded payloads.

2.3 Microscope Acquisition Details

We used a **KEYENCE VHX-7000** digital microscope (datasheet in §??) primarily at $400\times$ magnification, which proved the minimum viable level for reliable decoding. Only a small area is visible per frame; covering **about 10% of a sample took ~20 minutes** using stitched images. Frame resolution (per image) was **TODO: [insert e.g., 1920×1200 px]**, and a stitched montage for 10% measured **TODO: [insert px dimensions]**, which informed throughput targets.

2.4 QR Codes, Redundancy, and Locality

Information is organized into dense matrices of QR codes (4×4 per fs shot). Each QR uses Reed–Solomon redundancy, enabling recovery when parts are missing or distorted. A limitation of the current scheme is that redundancy is *local* to each QR. Systematic artefacts (e.g., a missing column across the field) can nullify many adjacent codes. A more resilient encoding would *interleave* data across a wider surface area to tolerate localized noise.

3 Methods and Work Performed

3.1 Setup (Hardware and Software)

Prototype carriers were imaged on CERN’s KEYENCE VHX-7000 microscope. Processing pipelines were implemented in C++ and Python using OpenCV; GPU acceleration used CUDA and OpenCV DNN. Benchmarks were run on multi-core CPUs and NVIDIA GPUs (datasheet for the tested GPU model in §??).

3.2 Image Processing Pipeline

The workflow consisted of:

1. microscope imaging of carriers,
2. detection of QR matrices,

3. perspective correction,
4. parallel QR decoding,
5. data reconstruction.

Initial trials with **deep learning** detectors (e.g., **YOLOv8** [6]) were too slow and inaccurate for dense multi-QR frames. The final approach used **classical computer vision**: grayscale + thresholding (Otsu), morphological operations, contour analysis, and `minAreaRect`-based warps to segment 4×4 matrices and crop each QR.

3.3 Decoder Libraries and Differences

We evaluated multiple decoders:

- **QUIRC** [3]: fast, pure CPU, binary image-based; performs well on clean, upright codes but is fragile to blur, perspective, or partial occlusion.
- **ZBar** [4]: CPU-only; robust on simple camera captures, but less tolerant to the microscope-specific noise patterns observed here (lower success on noisy crops).
- **OpenCV built-ins** (non-WeChat): moderate performance, limited tolerance to distortions in our data.
- **WeChat QR (OpenCV contrib)** [5]: learning-based detector/decoder pipeline that proved more tolerant to blur, mild perspective errors, and low contrast; best speed–accuracy trade-off on our microscope images.

In short: QUIRC is light and fast but brittle; ZBar is CPU-friendly but struggled with our noise; WeChat (OpenCV contrib) handled distortions better (see §4).

3.4 Parallelization and Acceleration

We used multithreading at the crop-decoding stage. The CPU baseline achieved **sub-5 ms** per QR. GPU acceleration via OpenCV DNN increased throughput to **~18.000 QR/s** on our tests. CUDA-based preprocessing further improved sustained performance (see §4). A hybrid of classical CV for detection with lightweight learned refinements offered the best balance.

3.5 Experiments Conducted

- deep learning detectors vs classical methods,
- decoder accuracy across libraries (QUIRC, ZBar, OpenCV, WeChat),
- impact of preprocessing on noisy microscope images,
- CPU vs GPU throughput for large-scale decoding.

4 Results

4.1 Decoding Success Rates

On synthetic QR images, decoding success exceeded **99%**. On real microscope images, success rates dropped due to dust, ablation variability, and illumination/focus. With optimized preprocessing and WeChat, the pipeline consistently achieved **>75%** on our sample sets. On one representative image:

- **ZBar**: 4/16 decoded (average ~ 23 ms/QR).
- **WeChat**: 12/16 decoded (total ~ 1409 ms, ~ 88 ms/QR in that Python test).

4.2 Performance Benchmarks

- **CPU multithreading**: sub-5 ms/QR, $\sim 12,000$ QR/s aggregate.
- **GPU (OpenCV DNN)**: $\sim 18,000$ QR/s aggregate.
- **CUDA preprocessing**: sustained **>1000 QR/s** with batch sizes 64–128 on an RTX-class GPU.

These results highlight scalability under parallel workloads and indicate a path toward the 20,000 —30,000 QR/s target.

4.3 Illustrative Figures

Placeholders for figures (add your images under `figures/`):

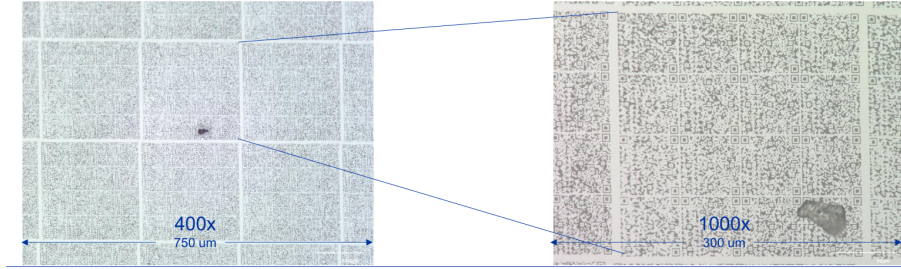


Figure 1: Raw microscope image of a 4×4 QR matrix on ceramic-on-glass.

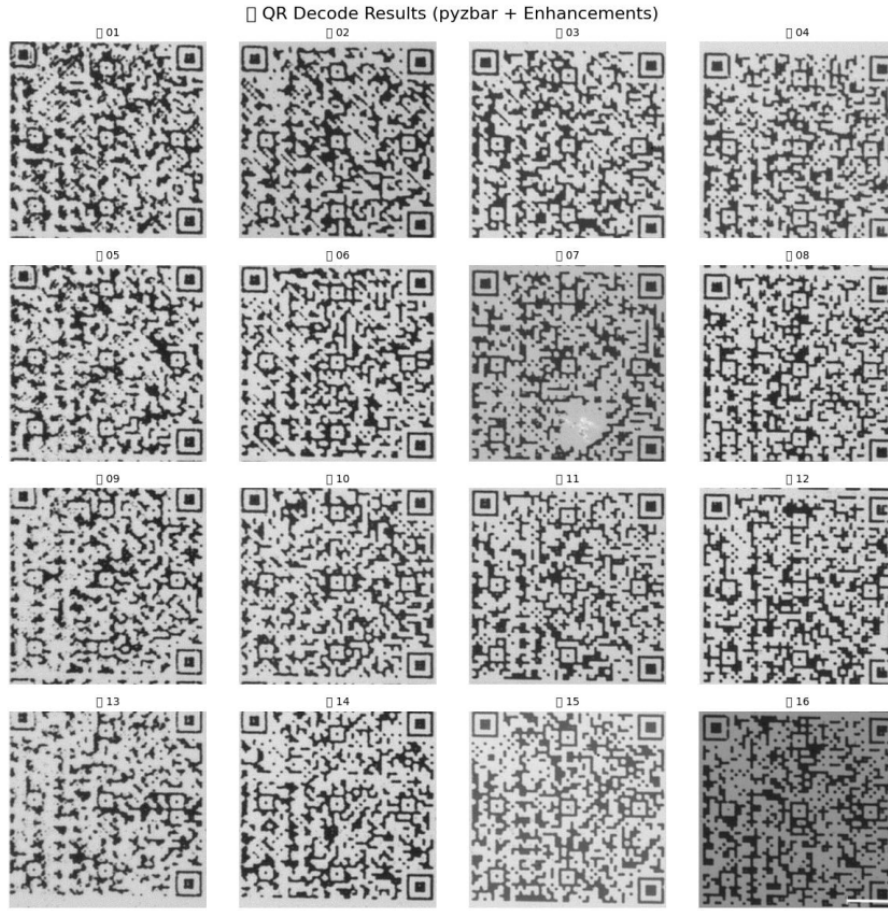


Figure 2: Preprocessed image (threshold + morphology + perspective correction) prior to decoding.

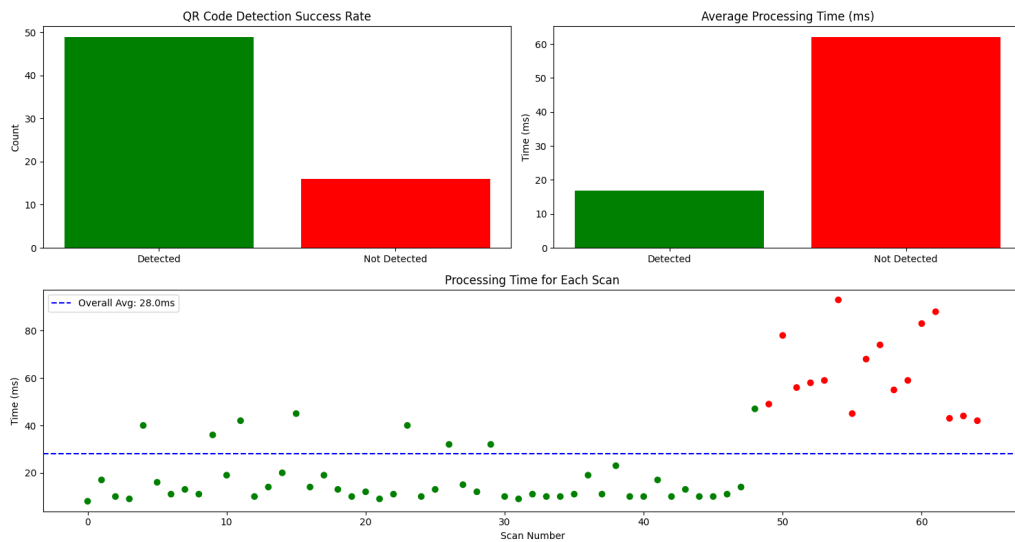


Figure 3: difference between detection time in different libraries

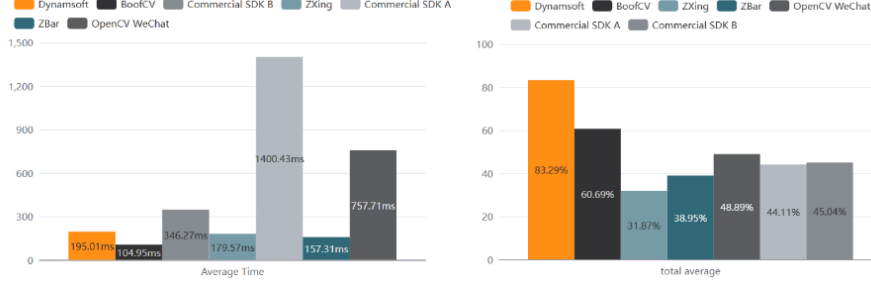


Figure 4: Decoder success rates on a representative sample (QUIRC vs ZBar vs WeChat).

5 Discussion

5.1 Strengths and Limitations of the Pipeline

Classical computer vision combined with WeChat decoding provides a practical balance of speed and robustness. The multithreaded design scales on CPUs and leverages GPUs for further gains. Remaining gaps include sensitivity to severe physical defects and achieving the upper bound of the target throughput on all datasets.

5.2 What Makes WeChat More Tolerant

Compared with purely classical decoders (QUIRC, non-WeChat OpenCV, ZBar), the WeChat pipeline incorporates learned features for detection and alignment, which improves tolerance to:

- mild blur and non-uniform illumination,
- small geometric distortions/perspective,
- partial occlusions or low-contrast finder patterns.

This explains its higher success on microscope images at the cost of higher per-QR latency in some Python tests; the C++/DNN pipeline reduces that overhead.

5.3 Redundancy Locality and Resilience

Because QR redundancy is local, structured artefacts (e.g., a missing column) can remove many adjacent codes at once. Interleaving blocks across a larger area or adding cross-matrix redundancy would increase resilience to localized damage.

5.4 On Longevity vs Readability

Ablation errors and dust affect *readability* and short-term error rates; they are not the reason for long-term *longevity*. Longevity follows from the physical stability of ceramic-on-glass, which this project did not evaluate experimentally.

6 Conclusion

This project investigated the feasibility of decoding QR code matrices from ceramic-on-glass carriers provided by Cerabyte. The main contributions include:

- establishing a microscope imaging and preprocessing workflow for prototype carriers,
- demonstrating that classical computer vision methods outperform deep learning for dense QR layouts under current constraints,
- benchmarking multiple decoders and identifying WeChat as the most robust option for our data,
- achieving sub-5 ms per QR on CPU and up to 18.000 QR/s with GPU acceleration,
- showing that a hybrid of classical vision and deep learning refinements provides the best balance between accuracy and throughput.

Mechanical features (library robotics, sheet stacking) are important at system level but were out of scope; they are noted in future work.

7 Future Work

- **GPU-first pipeline:** port the full preprocessing + decode chain to GPU (e.g., TensorRT) to push beyond 20.000 --30.000 QR/s.
- **Lightweight learned detectors:** train compact models tailored to dense, imperfect microscope layouts to improve robustness without large latency costs.
- **Full-cartridge evaluation:** test on complete cartridges with robotic handling and sheet stacking to validate end-to-end scaling.
- **Encoding schemes:** explore interleaving or cross-matrix redundancy to mitigate local artefacts (e.g., missing columns).

8 Lessons Learned

8.1 Technical

- Built and optimized a complete computer-vision pipeline for dense QR decoding.
- Benchmarked decoders and integrated CUDA acceleration.
- Demonstrated that a hybrid (CV + deep learning refinements) outperforms standalone approaches.

8.2 Collaboration

- Bridged workflows between CERN and Cerabyte teams.
- Improved interdisciplinary technical communication.

8.3 Personal Growth

- Balanced accuracy vs throughput trade-offs under real constraints.
- Gained confidence in combined hardware/software optimization.

Appendix: Code Snippets and Benchmark Tables

Sample Run: CPU Multithreading

```
podman run --rm qr-detector ./qr_reader -b
Processing image: input/Sample_001_mag500_b.tif
...
Total Matrices Found: 8
```

```
===== BENCHMARK RESULTS =====
Mode                Time (ms)          Speed-up
-----
Sequential          2551              1.00x
2 threads            2461              1.04x
4 threads            2508              1.02x
8 threads            2495              1.02x
16 threads           2503              1.02x
24 threads           2440              1.05x
32 threads           2473              1.03x
=====
```

High-Performance Decoder (12 Threads)

```
Running HIGH-PERFORMANCE QR decoder (12 threads)...
Successful QR decodes: 94/128
Average time per QR (total): 5.36 ms
Throughput: 186.4 QR/sec
Parallel efficiency: 629.8%
QR processing completed.
```

CUDA GPU Acceleration

```
=== High-Performance CUDA QR Processor Test ===
GPU Device: NVIDIA GeForce RTX 3050 Laptop GPU
Compute Capability: 8.6
...
Batch Size | Time (ms) | Throughput (QR/sec) | Memory Used (MB)
1           | 1.9       | 523.6                | 0
4           | 4.0       | 1001.8                | 1
8           | 7.3       | 1100.3                | 2
16          | 13.5      | 1186.5                | 4
32          | 25.7      | 1244.2                | 8
64          | 49.8      | 1284.5                | 16
```

128 | 117.8 | 1086.3 | 32

Best throughput: 1284.5 QR/sec (batch=64)
Sustained average over 5 runs: 1054.7 QR/sec

Decoder Comparison: ZBar vs WeChat (Representative Image)

Dataset: Cerabyte_Sample1_1000x_S0I4 (4).jpg

ZBar (Python):

Summary: 4/16 decoded successfully

Average time per QR: ~23 ms

WeChat (OpenCV Python variant):

Summary: 12/16 decoded successfully

Total time: 1409 ms (~88 ms/QR)

Benchmark Summary Table

Method	Decoded (16)	Avg Time / QR	Throughput
ZBar (CPU)	4 / 16	23 ms	~40 QR/sec
WeChat (CPU)	12 / 16	88 ms	~186 QR/sec
WeChat (12T)	94 / 128	5.3 ms	186 QR/sec
CUDA (Best)	—	0.78 ms	1284 QR/sec

Table 1: Decoder accuracy and performance comparison.

References

References

- [1] L. Kreuziger and A. Otto, “Revolutionizing optical data storage: high-speed, long-term solution with physically laser ablated matrices on ceramic-on-glass sheets,” 2024.
- [2] L. Kreuziger, *Sub-Micron Ablation of Chromium-based Thin Films for Optical Data Storage*, MSc Thesis, TU Wien, 2023.
- [3] D. Beer, “QUIRC: QR Code Recognition Library,” GitHub repository.
- [4] ZBar Bar Code Reader, “ZBar.” GitHub project and documentation.
- [5] WeChat QR Detector, OpenCV Contrib Module (dnn-based), documentation and source.
- [6] YOLOv8 Object Detection Models, Ultralytics, documentation and source.
- [7] KEYENCE VHX-7000 Digital Microscope, Product Datasheet.
- [8] NVIDIA GeForce RTX 3050 Laptop GPU, Product Specifications.