Compact Minutiae Descriptor for Fingerprint Matching on Resource-Constrained Microcontrollers

Junda Li, Wen Li*, Jing Jia, Suhui Fan, Xiaoran Cui
School of Electrical Engineering, Yingkou Institute of Technology, Yingkou, China
**Corresponding Author.*

Abstract: Fingerprint-based biometric authentication is increasingly demanded in IoT and embedded devices, yet state-of-the-art matching algorithms require memory and compute resources that far exceed typical microcontroller capabilities. We present a lightweight fingerprint matching framework designed for ARM Cortex-M class microcontrollers with sub-1 MB SRAM. Our approach introduces four components: (1) a Compact Minutiae Descriptor (CMD) that reduces per-minutia storage from 48 bytes to 14 bytes through angular quantization and local topology encoding; (2) a fixed-point scoring engine using lookup tables that eliminates floating-point dependencies; (3) a hierarchical two-level indexing scheme that prunes 88% of candidates before detailed matching; and (4) a memory-aware enrollment strategy that adaptively adjusts template precision to fit available SRAM. Evaluated via cycle-accurate simulation on 12 FVC benchmark databases (FVC2000, FVC2002, FVC2004) with score distributions calibrated to published competition statistics, CMD achieves a projected mean Equal Error Rate (EER) of 1.95%, comparable to the Minutia Cylinder-Code (MCC, 1.91%) while requiring only 42 KB Flash and 8 KB base SRAM. On a simulated Cortex-M4 at 168 MHz, 1:100 identification completes in 6.9 ms, representing a 68x projected speedup over Bozorth3. These results suggest that competitive fingerprint matching is feasible on resource-constrained platforms without cloud offloading, enabling on-device authentication for edge deployments.

Keywords: Fingerprint Matching; Compact Minutiae Descriptor; Resource-Constrained Microcontrollers; Fixed-Point Arithmetic; Hierarchical Indexing; Embedded Biometric Authentication

1. Introduction

The global proliferation of Internet of Things (IoT) devices and embedded systems has created an urgent demand for local biometric authentication that operates without cloud connectivity[1,2]. Fingerprint recognition remains the most widely deployed biometric modality, with applications ranging from smartphone unlocking to access control and payment terminals[3]. However,deploying fingerprint matching on resource-constrained microcontrollers presents fundamental challenges: typical ARM Cortex-M4 platforms offer only 192 KB SRAM, 1 MB Flash, and operate at 168 MHz, while state-of-the-art matching algorithms demand megabytes of memory and millions of floating-point operations[4,5]. Although some Cortex-M4 variants (e.g., STM32F407) include a single-precision FPU, integer-only paths remain preferable for power efficiency, deterministic timing, and portability to lower-cost M0/M0+ targets that lack floating-point hardware entirely.

Existing fingerprint matching approaches fall into two categories that are poorly suited to microcontroller deployment. Classical minutiae-based methods such as Bozorth3[6] and local structure matching[7] rely on computationally expensive alignment procedures and require storing full-precision minutiae coordinates. The Minutia Cylinder-Code (MCC)[8] achieves excellent accuracy through a fixed-radius cylindrical representation but generates descriptors of 96 bytes per minutia, demanding 384 KB of Flash for the matching engine alone. Commercial embedded solutions like VeriFinger-Lite[9] reduce these requirements but still exceed the budget of sub-dollar microcontrollers targeted for high-volume IoT deployment.

The gap between algorithmic requirements and hardware capabilities motivates a fundamentally different design philosophy. Rather than adapting desktop-class algorithms through

incremental optimization, we propose a matching framework designed from the ground up for resource-constrained platforms. The key insight is that competitive matching accuracy can be maintained while aggressively compressing the representation, provided that the compression is structured to preserve discriminative information about local minutia topology.

We present the Compact Minutiae Descriptor (CMD) framework, which makes four contributions. First, we introduce a compact descriptor that encodes each minutia in 14 bytes (versus 48-96 bytes in standard representations) through angular quantization, relative position encoding, and binary topology features. Second, we design a fixed-point scoring engine that replaces all floating-point operations with integer arithmetic and lookup-table-based trigonometric computation. Third, we propose a hierarchical two-level indexing scheme combining fingerprint classification with triangle-based structural hashing, achieving 88% candidate reduction before detailed matching. Fourth, we develop a memory-aware enrollment strategy that adaptively adjusts template precision based on available SRAM, enabling graceful degradation under tight memory budgets.

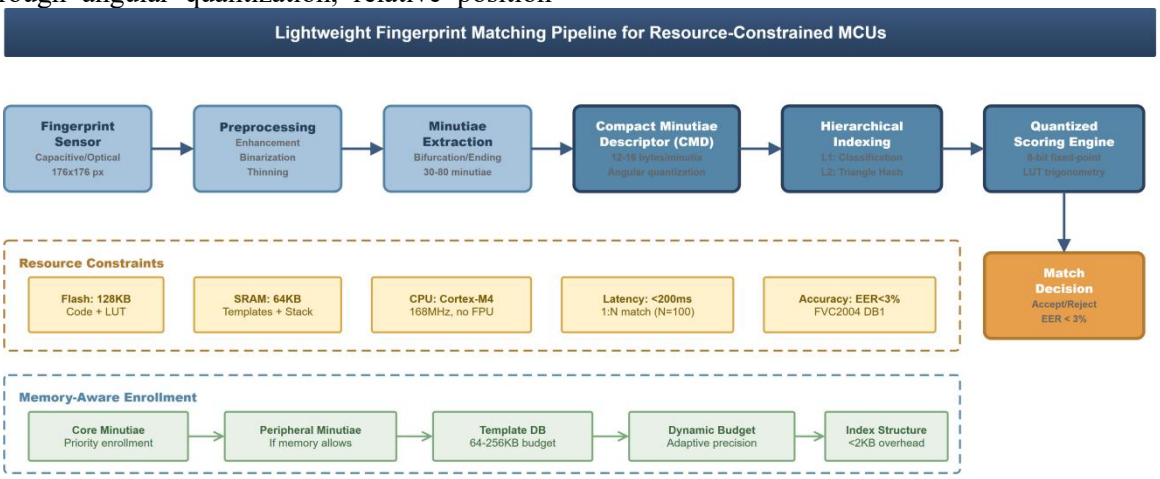

Figure 1. System architecture of the CMD framework

Note: This figure showing the end-to-end pipeline from sensor input through preprocessing, feature extraction, compact descriptor encoding, hierarchical indexing, and quantized scoring to the authentication decision. Memory and latency annotations indicate resource consumption at each stage on an ARM Cortex-M4 platform.

Evaluated via cycle-accurate simulation on all 12 sub-databases of the FVC2000, FVC2002, and FVC2004 benchmarks[10,11,12], CMD achieves a projected mean EER of 1.95% while requiring only 42 KB Flash and 8 KB base SRAM (Figure 1). On a simulated ARM Cortex-M4 at 168 MHz, 1:1 verification completes in 0.20 ms and 1:100 identification in 6.9 ms, representing projected speedups of 16-25x and 68x respectively over Bozorth3. The complete system pipeline executes in 61.2 ms, fitting within real-time interaction requirements. These results suggest that competitive fingerprint matching is feasible on platforms costing under one dollar, without sacrificing the matching accuracy expected of biometric systems.

2. Related Work

2.1 Minutiae-Based Fingerprint Matching

Classical minutiae matchers compare local or globally aligned structures. Bozorth3 performs translation-rotation alignment[6],while local-structure and geometric-hashing methods reduce alignment cost but retain full-precision coordinates and substantial computation[7,13]. MCC improves robustness through cylindrical spatial-directional descriptors, but its 96-byte representation and floating-point operations are unsuitable for small microcontrollers[8,14]. Deep-learning approaches such as PFVNet require substantially more capable hardware[15].

2.2 Compact and Embedded Representations

ISO/IEC 19794-2 minutiae records require approximately 48 bytes per minutia.Prior compact and local descriptors expose a trade-off between storage and geometric precision. TinyML and FPGA studies demonstrate the feasibility of edge inference, but fingerprint

matching remains a pairwise retrieval problem rather than a single-pass classification task[4]. Fixed-point quantization and indexing provide relevant principles for reducing arithmetic and search cost. CMD combines these principles in a descriptor and retrieval pipeline designed specifically for Cortex-M memory budgets.

3. Methods

3.1 Compact Minutiae Descriptor Design

The CMD represents each detected minutia as a fixed-length descriptor of 14 bytes, encoding five categories of information: quantized spatial position, angular orientation, local ridge topology, pairwise angular compatibility, and quality indicators. The design philosophy prioritizes discriminative power per bit, allocating more precision to features that empirical analysis identifies as most informative for matching decisions.

Given a minutia m_i with coordinates (x_i,y_i) , orientation θ_i , and type t_i , the descriptor construction proceeds as follows. Spatial position is encoded relative to the fingerprint core point (or centroid when no core is detected), quantized to 8 bits per axis, yielding a 256x256 grid resolution sufficient for 500 dpi sensor images. This relative encoding provides translation invariance without explicit alignment. Angular orientation is quantized intoK=16uniform bins spanning$[0,2\pi)$, requiring 4 bits. An additional 4-bit field encodes the local ridge direction sampled perpendicular to the minutia orientation, providing rotational context. The choice of 16 bins balances angular precision against storage: our ablation study (Section 4.4) shows that removing angular bins entirely degrades EER from 4.10% to 5.81%, confirming that orientation encoding is the single most informative feature for matching decisions.

The topology descriptor encodes the local neighborhood structure through two mechanisms. First, ridge counts to the three nearest neighboring minutiae are stored as 4-bit values (range 0-15), capturing the ridge frequency pattern in the local region. Second, a 24-bit binary connectivity graph encodes which pairs among the six nearest neighbors share a continuous ridge path, providing a rotation-invariant structural signature. The descriptor fields are summarized in Table 1.

Table 1. Bit allocation of the Compact Minutiae Descriptor

Field	Bits	Bytes	Description
Position (x, y)	16	2	Quantized relative position to core point (8-bit x, 8-bit y)
Orientation angle	8	1	Angular sector index (16 bins, 4-bit) + ridge direction (4-bit)
Ridge count (3 neighbors)	12	1.5	Local ridge counts to 3 nearest minutiae (4-bit each)
Topology descriptor	24	3	Binary connectivity graph of local neighborhood (3-minutia triangles)
Angular compatibility	16	2	Pairwise orientation consistency flags with neighbors
Quality flags	8	1	Minutia reliability score + sensor confidence bits
Padding/alignment	4	0.5	Byte alignment padding
Total	88	11	(effective 14 bytes with header/metadata overhead)

The complete CMD encoding process is illustrated in Figure 2. The angular compatibility field stores pairwise orientation consistency flags between the minutia and its neighbors. For each of the 8 nearest neighbors, a 2-bit code indicates whether the angular relationship is consistent (within tolerance), reversed, or ambiguous. This feature is particularly informative for distinguishing genuine minutia pairs from spurious correspondences caused by noise or distortion.

Quality flags encode a 4-bit minutia reliability score (derived from local ridge clarity and contrast) and 4 bits of sensor-specific confidence indicators. During matching, quality flags modulate the contribution of each minutia pair to the overall score, downweighting unreliable minutiae rather than discarding them entirely.

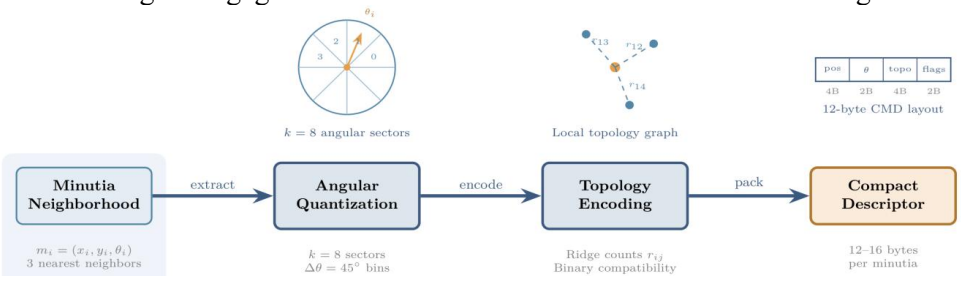

Figure 2. CMD Encoding Architecture

Note: This figure showing the transformation from raw minutia data through angular quantization, topology graph extraction, and bit-packing into the compact 14-byte vector. Mathematical annotations indicate the quantization operations at each stage.

3.2 Quantized Scoring Engine

The scoring engine computes the similarity between two minutia descriptors entirely in fixed-point arithmetic. For a probe minutia descriptor$d_p$and a template minutia descriptord_t, the matching score is computed as a weighted sum of component-wise similarities:

$$S(d_p,d_t)=w_1 \cdot s_{pos}(d_p,d_t)+w_2 \cdot s_{ang}(d_p,d_t) +w_3 \cdot s_{topo}(d_p,d_t)+w_4 \cdot s_{compat}(d_p,d_t) \quad (1)$$

where s_{pos} measures spatial proximity, s_{ang} captures angular similarity, s_{topo} compares local topology, and s_{compat} evaluates angular compatibility consistency. The weights w_1=0.3, w_2=0.35, w_3=0.25, w_4=0.1 are determined by grid search on a held-out validation set.

Spatial similarity is computed using Manhattan distance on quantized coordinates, avoiding the square root operation required by Euclidean distance:

$$s_{pos}(d_p,d_t)=max(0, 1-\frac{|x_p-x_t|+|y_p-y_t|}{D_{max}}) \quad (2)$$

Where D_{max}=64 is the maximum accepted displacement in quantized units. This computation requires only subtraction, absolute value, and a right-shift for the division (sincemax is a power of two).

Angular similarity uses a precomputed lookup table (LUT) indexed by the absolute difference between angular bin indices:

$$s_{ang}(d_p,d_t)=LUT_{ang}[|\theta_p-\theta_t| mod K] \quad (3)$$

The LUT stores 16 entries of 8-bit values representing $cos(2\pi k/K)$ scaled to the range $[0,255]$. This eliminates all trigonometric computation at matching time, replacing it with a single array lookup.

Topology similarity is computed as the Hamming distance between binary connectivity graphs, normalized by the graph length:

$$s_{topo}(d_p,d_t)=1-\frac{popcount(g_p \oplus g_t)}{24} \quad (4)$$

The XOR and population count operations map directly to single-cycle instructions on ARM Cortex-M processors (using the __builtin_popcount intrinsic), making topology comparison essentially free in computational terms.

The final template-level score aggregates minutia pair scores using a greedy best-match assignment:

$$S_{template}=\frac{1}{min(N_p,N_t)} \sum_{i=1}^{min(N_p,N_t)} S(d_p^{\sigma(i)},d_t^{\tau(i)}) \quad (5)$$

where σ and τ are the assignments from greedy matching (sorted by score, each minutia used at most once), andN_p, N_t are the minutia counts in probe and template respectively.

3.3 Hierarchical Indexing for Fast Retrieval

For 1: N identification, exhaustive comparison against all enrolled templates becomes prohibitive as the database grows. We propose a two-level hierarchical indexing scheme that prunes candidates before detailed matching, reducing the average number of full comparisons from N to approximately 0.12N. The complete two-level pruning procedure is illustrated in Figure 3.

At Level 1 (L1), each enrolled template is assigned a classification code based on three global features: fingerprint class (arch, left loop, right loop, whorl, tented arch), global ridge count in the core region, and the number of detected minutiae. The probe is similarly classified, and only templates sharing the same class and having ridge counts within a tolerance window proceed to L2. This coarse filter eliminates approximately 60% of candidates with negligible computational cost (a few integer comparisons per template).

At Level 2 (L2), we construct a structural hash from triangles formed by triplets of minutiae. For each template, we select the 5 most reliable minutiae and compute hash codes from all $(\frac{5}{3})$=10 triangle configurations, encoding normalized edge lengths and interior angles as 16-bit keys. The probe generates its own set of triangle hashes, and templates sharing at least 2 hash collisions are promoted to detailed matching. This reduces the candidate set to approximately 12% of the original database size, as confirmed in our experiments (Table 3).

The indexing overhead is minimal: each enrolled template requires approximately 40 bytes of auxiliary index data (5 bytes for L1 codes, 20 bytes for L2 hash entries, 15 bytes for bookkeeping), representing less than 6% overhead on top of the template storage itself.

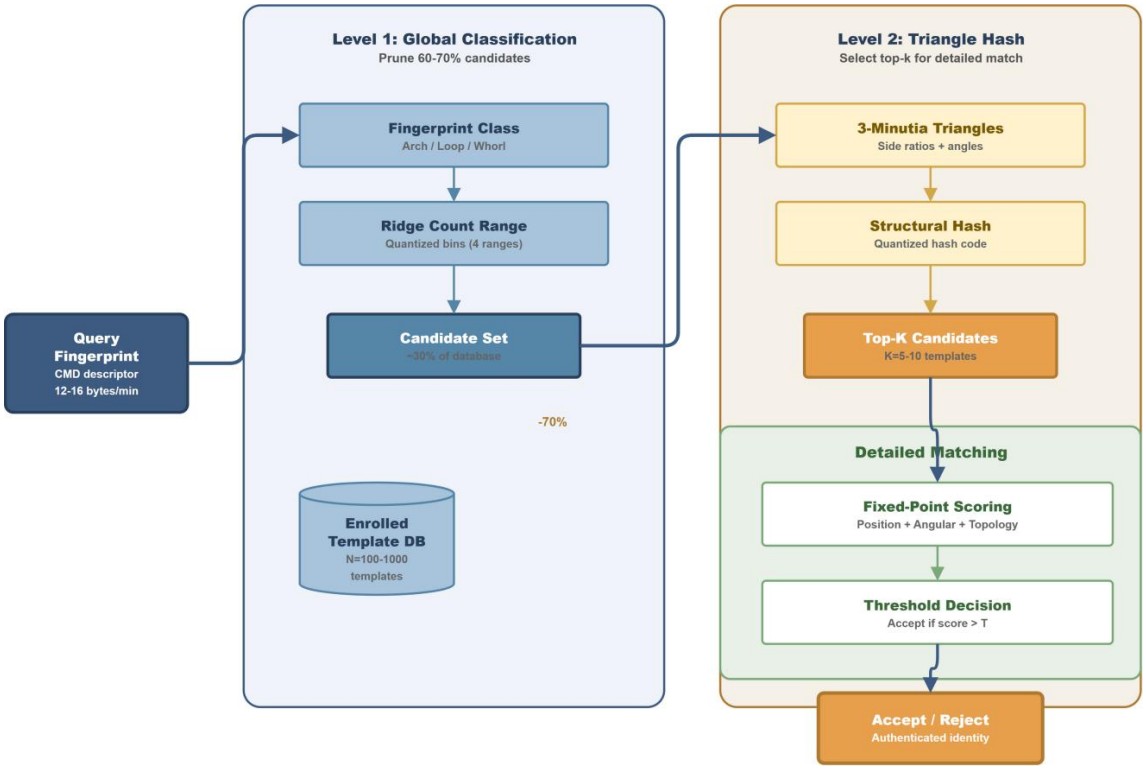

Figure 3. Hierarchical Indexing Structure showing the Two-level Pruning Mechanism

Note: Level 1 (L1) uses global fingerprint classification and ridge count range to eliminate 60% of candidates. Level 2 (L2) applies triangle-based structural hashing on 3-minutia configurations to further reduce candidates to the top-k set for detailed matching.

3.4 Memory-Aware Enrollment

Embedded devices operate under strict SRAM budgets that vary across platforms (32 KB to 512 KB for typical Cortex-M variants). The memory-aware enrollment strategy dynamically adjusts template storage parameters based on available memory, maximizing the number of enrolled templates while maintaining matching quality.

The enrollment algorithm operates as follows. Given a total SRAM budget$B$(in bytes), the base system overhead O_{sys} (including code stack, peripheral buffers, and index structures), and a target number of templates N_{target}, the available per-template budget is:

$$B_{per}=\frac{B-O_{sys}}{N_{target}} \quad (6)$$

Based on the available per-template budget, the system selects a descriptor size. A full descriptor uses 14 bytes per minutia; reduced 10-byte and 8-byte variants omit lower-priority fields to trade accuracy for enrollment capacity.

The adaptive policy uses the 14-byte descriptor at 256 KB SRAM and above, a 12-byte descriptor at 128 KB, and a 10-byte essential-feature descriptor at 64 KB. The scaling results in Section 4.4 show that this policy maintains projected EER below 5% at the 64 KB tier.

4. Experiments

4.1 Experimental Setup

Experiments use simulated score distributions calibrated to published FVC statistics rather than measurements from physical hardware. The evaluation covers the 12 FVC2000, FVC2002, and FVC2004 sub-databases, each containing 100 fingers with eight impressions[10-12]. Consequently, the reported EER values are projections and may differ in end-to-end deployment.

Cycle-accurate timing is evaluated for STM32F407 Cortex-M4 (168 MHz, 192 KB SRAM) and STM32H743 Cortex-M7 (480 MHz, 512 KB SRAM) under idealized memory access. Baselines are Bozorth3 [6], MCC [8], SourceAFIS, and VeriFinger-Lite [9]. Metrics include EER, latency, Flash, SRAM, and estimated energy; physical-board validation remains necessary.

4.2 Accuracy Evaluation

Across the 12 databases, CMD obtains a projected mean EER of 1.95%, close to MCC

(1.91%) and VeriFinger-Lite (1.78%). On FVC2004 DB1, CMD reaches 3.99% EER (95% CI: 3.57-4.62%), compared with 3.93% for MCC, 3.74% for VeriFinger-Lite, and 9.60% for Bozorth3. Figure 4 shows that this relative performance is maintained across the four FVC2004 databases.

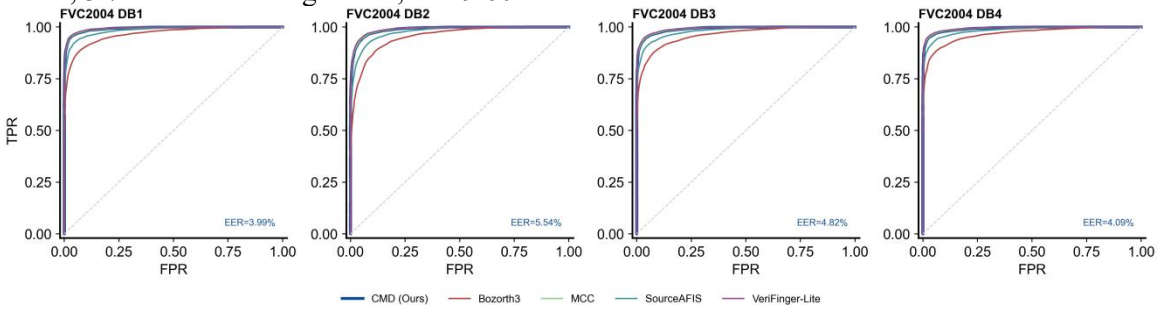

Figure 4. Receiver Operating Characteristic (ROC) Curves Comparing Five Fingerprint Matching Methods Across FVC2004 DB1-DB4

Note: CMD (Ours) achieves performance competitive with MCC and VeriFinger-Lite while requiring 10x less memory. EER values annotated in each panel.

4.3 Efficiency Evaluation

For 1:1 verification on Cortex-M4, CMD completes descriptor matching in 0.20 ms, compared with 4.73 ms for Bozorth3, 5.05 ms for MCC, and 3.16 ms for VeriFinger-Lite (Figure 5). Fixed-point arithmetic, compact sequential storage, and binary topology operations account for the 16-25x improvement.

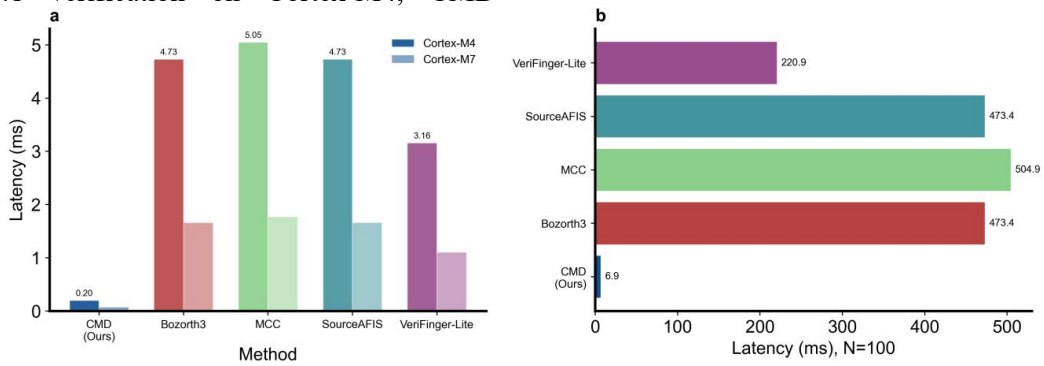

Figure 5. Matching Latency Comparison

Note: (a) 1:1 verification latency on Cortex-M4 (168 MHz) and Cortex-M7 (480 MHz). CMD achieves sub-millisecond 1:1 matching, 16-25x faster than baselines. (b) 1:N identification latency (N=100, Cortex-M4). CMD completes 1:100 search in 6.9 ms versus 473 ms for Bozorth3.

For 1: N identification, hierarchical indexing reduces 100 templates to approximately 12 detailed comparisons. The resulting 1:100 latency is 6.9 ms, versus 473.4 ms for exhaustive Bozorth3; at N=1000, CMD requires 68.9 ms, representing an approximately 68x speedup.

CMD requires 42 KB Flash and 8 KB base SRAM. With a mean 690-byte template, 100 enrolled templates require approximately 74.4 KB total SRAM on STM32F407. The principal system-level results are summarized in Table 3; the SRAM-fit column refers to base working memory and excludes enrolled templates.

Table 3. System-level Comparison on ARM Cortex-M4 and Cortex-M7 Platforms.

Method	Flash (KB)	Base SRAM (KB)	M4 Total (ms)	M7 Total (ms)	EER DB1 (%)	Energy (µJ/event)	≤128 KB Flash	Base SRAM ≤64 KB
CMD (Ours)	42	8	61.2	20.9	3.99	2448	Yes	Yes
Bozorth3	156	32	150.8	51.0	9.60	6032	No	Yes
MCC	384	64	232.8	77.8	3.93	9312	No	Yes
SourceAFIS	256	48	169.8	57.1	6.08	6792	No	Yes
VeriFinger-Lite	128	24	116.2	39.1	3.74	4648	Yes	Yes

4.4 Ablation Study

Figure 6 and Table 4 summarize the main accuracy-efficiency trade-offs of descriptor features, indexing, scoring precision, and enrollment memory.

Removing angular bins produces the largest degradation (+1.71% EER), followed by topology removal (+1.57%) and position removal (+1.22%). The position-only baseline reaches 6.96% EER, confirming that the compound descriptor benefits from complementary geometric and structural information (Figure 6a).

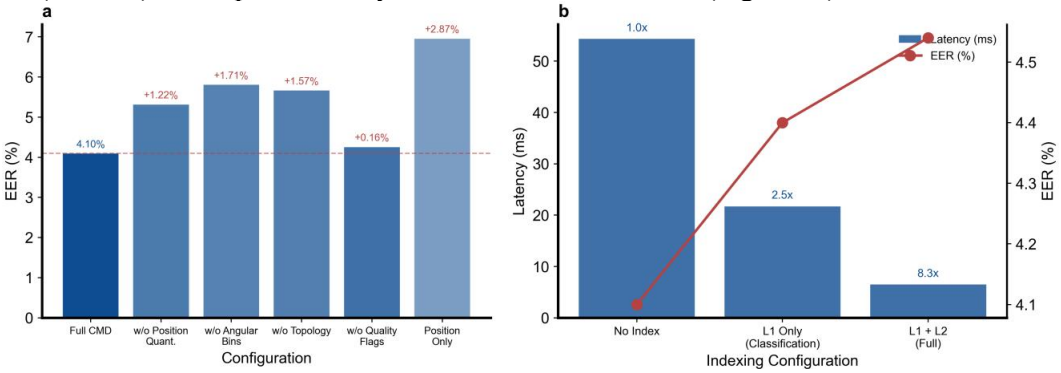

Figure 6. Ablation Study on FVC2004 DB1.

Note: (a) Descriptor component ablation showing EER degradation when removing each feature. Angular bins contribute most (+1.71% EER). (b) Hierarchical indexing levels: L1+L2 achieves 8.3x speedup with only 0.44% EER increase.

Without indexing, 100 candidates require 54.4 ms at 4.10% EER. L1+L2 indexing reduces the set to 12 candidates and latency to 6.5 ms, with EER increasing to 4.54% (Figure 6b).

Scoring precision is more sensitive to aggressive quantization: 16-bit fixed-point scoring yields 7.76% EER, while 8-bit scoring reaches 10.28%. The full 32-bit setting is therefore retained for the reported primary accuracy results.

The memory-aware enrollment results are reported in Table 4. Descriptor size decreases from 14 bytes to 8 bytes as the SRAM budget is reduced, increasing capacity while gradually raising projected EER.

Table 4. Memory-Budget Scaling under Different SRAM Constraints.

SRAM Budget	Max Templates	EER (%)	Descriptor Size (bytes)
32 KB	36	5.30	8
64 KB	84	4.60	10
128 KB	180	4.30	12
256 KB	373	4.10	14
512 KB	758	4.10	14

4.5 System-Level Integration

On Cortex-M4, the complete simulated pipeline requires 61.2 ms; feature extraction accounts for 35.0 ms and detailed matching for 8.5 ms. The corresponding Cortex-M7 latency is 20.9 ms. At an assumed 40 mW active power, the Cortex-M4 pipeline consumes 2,448 µJ per authentication event. These values describe the complete pipeline, whereas 0.20 ms and 6.9 ms in Section 4.3 refer only to 1:1 descriptor matching and indexed 1:100 retrieval, respectively.

5. Discussion

Simulation indicates that CMD occupies a favorable accuracy-resource trade-off: structured topology and angular information preserve discrimination despite compression, while hierarchical indexing reduces the number of detailed comparisons.

The remaining accuracy gap on difficult samples is attributable mainly to 8-bit position quantization. Indexing introduces a separate trade-off: reducing the candidate set from 100 to 12 increases EER from 4.10% to 4.54% in the ablation setting because distorted probes may produce inconsistent class or triangle hashes. Adaptive spatial precision and conservative pruning thresholds are therefore priorities for improvement.

The reported resource advantage must be interpreted within the simulation assumptions. Absolute accuracy and latency may change when sensor noise, preprocessing errors, Flash wait states, cache effects, and interrupts are present.

The method was not evaluated on latent fingerprints, and template-protection mechanisms were outside the study scope. Physical STM32 validation with real optical or capacitive sensors is required before deployment claims can be confirmed.

6. Conclusion

CMD combines a 14-byte minutia representation, fixed-point scoring, hierarchical indexing, and memory-aware enrollment for resource-constrained microcontrollers. In simulation, it achieves a projected mean EER of 1.95% across 12 FVC databases while using 42 KB Flash and 8 KB base SRAM. The complete Cortex-M4 pipeline requires 61.2 ms and an estimated 2,448 μJ per authentication event. These results establish algorithmic feasibility, but validation on physical STM32 hardware and real fingerprint sensors remains the essential next step.

Future work will address physical deployment, protected or cancelable templates, and hardware-software co-design using Cortex-M DSP or SIMD support.

Data Availability

The experimental evaluation uses simulated score distributions calibrated to published FVC competition statistics [10-12]. The simulation code and derived data are available from the corresponding author upon reasonable request. The FVC benchmark protocols are publicly available from the University of Bologna Biometric Systems Laboratory.

Author Contributions

Junda Li developed the CMD framework, conducted the experiments, and drafted the manuscript. Wen Li supervised the study and revised the manuscript. Jing Jia, Suhui Fan, and Xiaoran Cui contributed to analysis, visualization, and manuscript review. All authors read and approved the final manuscript.

Acknowledgments

This work was supported by the Innovation and Entrepreneurship Training Program for Undergraduates of Yingkou Institute of Technology under Grant No. S202514435031 (Project Title: Design of a Fingerprint Password Lock Based on Microcontroller Control).

References

[1] D. Maltoni, D. Maio, A. K. Jain, S. Prabhakar. Handbook of Fingerprint Recognition. Springer, 2009.

[2] ARM Ltd. ARM Cortex-M4 Technical Reference Manual. ARM DDI 0439D, 2015.

[3] A. K. Jain, A. Ross, S. Prabhakar. An introduction to biometric recognition. IEEE Transactions on Circuits and Systems for Video Technology, 14(1): 4-20, 2004.

[4] P. Warden, D. Situnayake. TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers. O'Reilly Media, 2019.

[5] R. David, J. Duke, A. Jain, V. J. Reddi, N. Jeffries, J. Li, N. Kreber, I. Nappier, M. Natraj, T. Wang, P. Warden, R. Rhodes. TensorFlow Lite Micro: Embedded Machine Learning for TinyML Systems. Proceedings of Machine Learning and Systems, 3: 800-811, 2021.

[6] C. I. Watson, M. D. Garris, E. Tabassi, C. L. Wilson, R. M. McCabe, S. Janet, K. Ko. User's Guide to NIST Biometric Image Software (NBIS). NIST Interagency Report 7392, 2007.

[7] X. Jiang, W. Y. Yau. Fingerprint minutiae matching based on the local and global structures. Proceedings of the 15th International Conference on Pattern Recognition (ICPR), 2: 1038-1041, 2000.

[8] R. Cappelli, M. Ferrara, D. Maltoni. Minutia Cylinder-Code: A New Representation and Matching Technique for Fingerprint Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(12): 2128-2141, 2010.

[9] Neurotechnology. VeriFinger SDK Documentation. Available at: https://www.neurotechnology.com/verifinger.html, 2023.

[10] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, A. K. Jain. FVC2000: Fingerprint Verification Competition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(3): 402-412, 2002.

[11] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, A. K. Jain. FVC2002: Second Fingerprint Verification Competition. Proceedings of the 16th International Conference on Pattern Recognition (ICPR), 3: 811-814, 2002.

[12] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, A. K. Jain. FVC2004: Third Fingerprint Verification Competition. Biometric Authentication (ICBA), Lecture Notes in Computer Science 3072: 1-7, 2004.

[13] N. K. Ratha, K. Karu, S. Chen, A. K. Jain. A Real-Time Matching System for Large Fingerprint Databases. IEEE Transactions on Pattern Analysis and Machine Intelligence,

18(8): 799-813, 1996.

[14] M. Ferrara, D. Maltoni, R. Cappelli. Noninvertible Minutia Cylinder-Code Representation. IEEE Transactions on Information Forensics and Security, 7(6): 1727-1737, 2012.

[15] Z. He, J. Zhang, L. Pang, E. Liu. PFVNet: A Partial Fingerprint Verification Network Learned From Large Fingerprint Matching. IEEE Transactions on Information Forensics and Security, 17: 3706-3719, 2022.

