



Research Article

An Efficient and Deterministic Object Recognition System

Huseyin Murat Cekirge* 

Department of Mechanical Engineering, The City College of New York (CUNY), New York, USA

Abstract

This paper presents an efficient deterministic framework for small-scale object recognition based on geometric normalization and invariant-driven elimination. The proposed method avoids training procedures and iterative optimization, operating instead through structured preprocessing, scalar invariant filtering, and deterministic vector comparison within a compact single-table vector database. Each object is represented as a feature vector derived from a fixed 7×7 grid and associated geometric descriptors. Recognition proceeds through a three-stage pipeline: (i) centroid-based translation and scale normalization, (ii) invariant-based candidate reduction using mass and second-order spatial moments, and (iii) localized orientation refinement followed by cosine similarity matching. Invariant-driven preselection significantly reduces the candidate set prior to fine comparison, improving computational efficiency while avoiding exhaustive rotational search. The method is lightweight, reproducible, and free of stochastic components. Experimental validation on a 100-object prototype library demonstrates stable and accurate identification under scale variation and controlled rotational perturbations. The results indicate that for structured, low-dimensional pattern libraries, deterministic geometric reasoning combined with finite database ordering provides a practical and computationally controlled recognition mechanism. The proposed framework is designed for structured and finite pattern libraries where objects can be represented through compact grid-based descriptors. In such environments, deterministic geometric preprocessing and invariant filtering significantly reduce the search space before final similarity evaluation. This allows recognition to be completed in a single deterministic evaluation cycle without iterative optimization or training procedures. The approach therefore provides a computationally lightweight alternative for embedded systems, symbolic recognition tasks, and controlled pattern libraries where structural compatibility can be exploited effectively.

Keywords

Geometric Normalization, Invariant Filtering, Cosine Similarity, Finite Compatibility Ordering, Spectral Gap, Rotation Handling, Moment Invariants, Single-table Vector Database

1. Introduction

Object recognition is commonly addressed through statistical learning frameworks, including convolutional neural networks and gradient-based optimization methods. Comprehensive surveys of statistical pattern recognition describe the dominance of learning-based approaches in modern systems [1-4]. While such systems are powerful in complex and high-

variability environments, they typically require training procedures, iterative parameter updates, and substantial computational resources.

Before the widespread adoption of deep learning, object recognition problems were frequently formulated using deter-

*Correspondence: Huseyin Murat Cekirge (hmcekirge@usa.net)

Received: 27 February 2026; Accepted: 11 March 2026; Published: 23 March 2026



Copyright: © The Author(s), 2026. Published by Science Publishing Group. This is an **Open Access** article, distributed under the terms of the Creative Commons Attribution 4.0 License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

ministic geometric descriptors and distance-based classification rules. Foundational work by Hu introduced moment invariants capable of providing robustness under translation, rotation, and scale transformations [1]. Deterministic decision rules and minimum-distance classifiers were systematically developed within classical pattern classification frameworks [2]. These approaches established recognition as a structured comparison problem over finite feature representations rather than as an optimization trajectory.

Geometric normalization techniques—including centroid alignment, covariance-based principal axis estimation, and region descriptors—are extensively documented in standard image processing literature [5-7]. These methods provide efficient preprocessing mechanisms that avoid exhaustive rotational search. In particular, covariance-based orientation estimation enables deterministic alignment via second-order spatial moments. Template matching and normalized correlation measures further established cosine similarity as a computationally effective deterministic similarity metric [8].

In constrained environments—such as symbol recognition, embedded systems, or compact grid-based representations—a deterministic alternative can provide predictable behavior, reduced computational cost, and architectural transparency. Motivated by classical geometric reasoning, this work proposes a deterministic object recognition framework tailored for structured, low-dimensional pattern libraries.

Objects are stored in a single-table vector database in which each pattern is represented by a fixed-length row-wise flattened feature vector. This representation eliminates the need for relational joins, hierarchical feature trees, or learned convolutional filters. Identification proceeds through structured elimination rather than exhaustive comparison. The central idea is to reduce the candidate space using inexpensive geometric invariants prior to performing final vector similarity evaluation.

The proposed pipeline consists of three principal stages:

- 1) Geometric normalization, including centroid translation and scale adjustment;
- 2) Invariant-based elimination using total mass and second-order spatial energy to reduce the candidate set;
- 3) Localized orientation refinement followed by cosine similarity comparison for final identification.

Rotation is handled deterministically through covariance-based principal direction estimation, thereby avoiding full 360° brute-force sweeps. Rotation-invariant scalar descriptors collapse the search space prior to fine comparison, enabling efficient and reproducible matching.

The recognition problem is therefore reformulated as a controlled finite ordering task over a compact compatibility spectrum. Rather than approaching a solution asymptotically through iterative descent, identification is achieved by selecting the maximal element of a finite similarity set:

$$\hat{k} = \arg \max_k s_k.$$

For structured libraries and low-dimensional representations, deterministic geometric reasoning provides an effective and computationally controlled recognition mechanism grounded in compatibility ordering rather than optimization dynamics.

2. Related Work

Modern object recognition systems are predominantly formulated within statistical learning frameworks. Comprehensive surveys describe the transition from classical rule-based classifiers to data-driven models trained through optimization procedures [1]. In contemporary practice, convolutional neural networks (CNNs) and gradient-based optimization methods dominate large-scale visual recognition tasks due to their ability to model high intra-class variability and complex feature hierarchies [9-17].

These systems operate by minimizing empirical loss functions through iterative parameter updates. Stability and generalization are governed by optimization dynamics, regularization mechanisms, and training data distribution. While highly effective in unconstrained environments, such approaches typically require substantial training datasets, computational resources, and hyperparameter tuning.

Prior to the widespread adoption of deep learning, object recognition was frequently addressed through deterministic geometric descriptors and explicit distance-based decision rules. Moment invariants introduced in [2] provided rotation, translation, and scale robustness without iterative learning. Extensions and generalizations of moment-based descriptors further established invariant feature extraction as a principled deterministic alternative [11, 12].

Classical pattern classification theory formalized minimum-distance and linear decision rules within finite-dimensional feature spaces [3]. In these formulations, recognition is achieved by evaluating compatibility measures rather than by optimizing high-dimensional parameter landscapes. Template matching and normalized correlation techniques later demonstrated that cosine similarity can serve as an efficient deterministic similarity metric for structured patterns [8].

Geometric preprocessing techniques—such as centroid alignment, covariance-based principal axis estimation, and region descriptors—are extensively documented in image processing literature [4-7]. These methods enable alignment and normalization without exhaustive rotational sweeps, thereby reducing computational complexity prior to classification.

The present work does not aim to compete with large-scale learning systems designed for high-variability natural images. Instead, it revisits and systematizes deterministic geometric reasoning for structured, low-dimensional pattern libraries. The emphasis shifts from optimization trajectories to finite compatibility ordering. Recognition is treated as a well-posed decision problem over a compact similarity spectrum rather than as the convergence of an iterative learning process.

In this sense, the proposed framework is positioned within

the classical deterministic tradition of pattern recognition, while incorporating structured invariant filtering to reduce computational effort before final similarity evaluation.

The objective of the present study is not to provide a benchmark comparison with statistical or learning-based recognition systems, but to establish the deterministic recognition framework and demonstrate its structural mechanism under controlled conditions.

3. Methodology

Unlike classical template matching, which performs exhaustive similarity comparison across all stored patterns, the proposed deterministic framework reduces the candidate set through invariant-based elimination prior to the final cosine similarity evaluation.

3.1. Library Construction and Database Structure

Each object is represented as an $N \times N$ real-valued matrix

$$A^{(k)} \in \mathbb{R}^{N \times N}, k = 1, \dots, M \quad (1)$$

For the prototype system, $N = 7$ and $M = 100$.

Each matrix is flattened using a consistent row-wise ordering into a vector

$$v^{(k)} \in \mathbb{R}^{N^2} \quad (2)$$

defined by

$$v^{(k)} = \text{vec}(A^{(k)}) \quad (3)$$

For $N = 7$,

$$v^{(k)} \in \mathbb{R}^{49} \quad (4)$$

Each object is assigned a unique identifier (UID).

The library is stored in a single-table vector database of the form:

$$\text{UID} \mid f_1 \mid f_2 \mid \dots \mid f^{N^2} \quad (5)$$

where f_i denotes the i -th flattened pixel value.

Optional scalar invariants (mass, energy) may be stored as additional columns.

3.2. Library Matrix Form

All object vectors are stacked column-wise to form the deterministic library matrix

For the 7×7 prototype,

$$L \in \mathbb{R}^{49 \times 100} \quad (6)$$

Recognition is therefore reduced to structured matrix–vector operations.

3.3. Blind Object Representation

A query object is represented as

$$B \in \mathbb{R}^{N \times N} \quad (7)$$

Flattened into

$$b \in \mathbb{R}^{N^2} \quad (8)$$

$$b = \text{vec}(B) \quad (9)$$

3.4. Geometric Normalization

The blind object undergoes deterministic preprocessing.

Mass (zeroth moment):

$$M_0 = \sum_i b_i \quad (10)$$

Centroid:

$$x_c = (\sum_i x_i b_i) / M_0, y_c = (\sum_i y_i b_i) / M_0 \quad (11)$$

Translation to centroid:

$$X' = X - x_c \quad (12)$$

$$Y' = Y - y_c \quad (13)$$

Scale normalization:

$$x = \lambda X' \quad (14)$$

$$y = \lambda Y' \quad (15)$$

where λ ensures containment within the fixed $N \times N$ grid.

If rotation handling is required, orientation is estimated via the covariance matrix

$$C = \begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix} \quad (16)$$

The principal direction is obtained from the eigenvectors of C .

Only a small angular vicinity is evaluated, avoiding exhaustive rotation.

3.5. Deterministic Elimination Stage

Two scalar invariants are computed:

Mass:

$$M = \sum_i b_i \quad (17)$$

Radial energy:

$$E = \sum_i (x_i^2 + y_i^2) b_i \quad (18)$$

Preselection is performed using tolerance windows:

$$|M - M^{(k)}| \leq \delta \quad (19)$$

$$|E - E^{(k)}| \leq \varepsilon \quad (20)$$

This reduces the candidate subset prior to full similarity evaluation.

3.6. Deterministic Matching

Each library vector is normalized once:

$$\tilde{v}^{(k)} = v^{(k)} / \|v^{(k)}\| \quad (21)$$

The blind vector is normalized:

$$\tilde{b} = b / \|b\| \quad (22)$$

Cosine similarity is computed as:

$$s_k = \tilde{b}^T \tilde{v}^{(k)} \quad (23)$$

In matrix form:

$$s = L^T \tilde{b} \quad (24)$$

where $s \in \mathbb{R}^M$.

Recognition is performed by:

$$\hat{k} = \arg \max_k s_k \quad (25)$$

The UID corresponding to $\hat{k}$ is returned.

3.7. Structural Decision Operator and Well-Posedness

The identification stage is governed by the deterministic operator (25).

For a finite library of size M , define the compatibility spectrum

$$S = \{s_1, s_2, \dots, s_M\}. \quad (26)$$

Recognition reduces to selecting the maximal element of S , because S is finite, a maximal element always exists. Therefore, recognition is not formulated as a search trajectory in parameter space but as a finite ordering problem over a discrete compatibility spectrum.

Uniqueness

Let s_j denote the similarity score between the query object and the j -th library element.

Let k^* denote the true class index and $\hat{k}$ denote the estimated index obtained by the maximization rule

$$\hat{k} = \arg \max_k s_k. \quad (27)$$

If there exists k^* such that

$$s_{k^*} > s_j \text{ for all } j \neq k^*, \quad (28)$$

then the solution is unique and

$$\hat{k} = k^* \quad (29)$$

Stability

Define the spectral gap

$$\delta = s_{k^*} - \max_{j \neq k^*} s_j. \quad (30)$$

The quantity δ measures the separation between the correct class and its closest competitor. If $\delta > 0$ and the compatibility scores depend continuously on the input representation, then perturbations smaller than $\delta/2$ preserve the ordering of the scores. Consequently, the index $\hat{k}$ remains invariant under sufficiently small input perturbations.

Existence follows from finiteness of S . Uniqueness follows from strict dominance. Stability follows from positive spectral separation.

Hadamard Well-Posedness

Existence follows from the finiteness of the score set S .

Uniqueness follows from strict dominance (28).

Stability follows from positive spectral separation $\delta > 0$ and continuity of the score mapping.

Therefore, the deterministic decision problem is well-posed in the sense of Hadamard.

3.8. Deterministic Rotation Handling

In structured grid-based representations, rotational variability may degrade direct vector comparison. To avoid brute-force angular sweeps, orientation is estimated deterministically using second-order spatial moments.

Covariance matrix:

$$\Sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix} \quad (31)$$

Principal axis orientation:

$$\theta = (1/2) \arctan(2\sigma_{xy} / (\sigma_{xx} - \sigma_{yy})) \quad (32)$$

The blind object is rotated by $-\theta$ for coarse alignment. Optional refinement within a small angular vicinity may be applied if necessary.

For symmetric patterns where the principal direction is ambiguous, orientation filtering is bypassed and structural compatibility is resolved directly via cosine similarity.

3.9. Efficient Telescopic Elimination

Recognition may be further accelerated through staged elimination.

Let

$$D = \{S_1, S_2, \dots, S_N\} \quad (33)$$

Each sample

$$S_i \in \mathbb{R}^m, \text{ where } m = n \times n. \quad (34)$$

A partition operator

$$Pk: \mathbb{R}^m \rightarrow \mathbb{R}^{mk} \quad (35)$$

with

$$m = \sum mk \quad (36)$$

defines structured components.

Sequential Elimination Structure

Initial candidate set;

$$C_0 = D. \quad (37)$$

Here, D denotes the full library (database).

At stage k

$$C_k = \{ S_i \in C_{k-1} \mid d_k(S_i, Q) < \tau_k \}. \quad (38)$$

where

Q is the query object,

S_i are library elements,

$d_k(\cdot, \cdot)$ denotes the stage-k compatibility metric,

τ_k is the threshold at stage k.

Monotonic Reduction Property

Each stage reduces the candidate set:

$$C_k \subseteq C_{k-1} \subseteq \dots \subseteq C_0. \quad (39)$$

If τ_k is properly chosen, then

$$|C_k| \leq |C_{k-1}|. \quad (40)$$

Strict inequality holds whenever at least one candidate is eliminated at stage k.

Cardinality Decay

$$|C_0| = N \quad (41)$$

$$|C_1| \ll N \quad (42)$$

$$|C_2| \ll |C_1| \quad (43)$$

$$|C_K| \approx 1 \quad (44)$$

This produces geometric shrinkage rather than brute-force comparison.

Rotation, if required, is applied only after substantial reduction of the candidate set.

4. Numerical Example (3×3 Prototype)

To illustrate the deterministic recognition procedure in its most elementary algebraic form, consider a minimal 3×3 binary prototype library. This example serves as a complete structural sanity check of the formulation introduced in Section 3.

4.1. Library Construction

Let the three stored objects be denoted by X, T and +.

Their matrix representations are:

$$A^{(1)} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad A^{(2)} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad A^{(3)} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Figure 1. The matrices of the library.

Each object is flattened row-wise into a 9-dimensional column vector.

$$v^{(1)} = [1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]^T \quad (45)$$

$$v^{(2)} = [0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0]^T \quad (46)$$

$$v^{(3)} = [0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1]^T \quad (47)$$

The library matrix is constructed by stacking the vectors column-wise:

$$L = [v^{(1)} \ v^{(2)} \ v^{(3)}] \in \mathbb{R}^{9 \times 3} \quad (48)$$

Thus, each column of L corresponds to one stored object in the database representation.

4.2. Blind Object

Library matrix:

$$L = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{9 \times 3}$$

Figure 2. The library matrix and database table without UID.

After row-wise flattening:

$$\mathbf{b} = [0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0]^T. \quad (49)$$

$$\mathbf{s}_k = \tilde{\mathbf{b}}^T \tilde{\mathbf{v}}^{(k)}. \quad (63)$$

4.3. Elimination Using Mass

Define the mass (ℓ_1 content) of a binary object as

$$M = \sum_i b_i \quad (50)$$

where $b_i \in \{0,1\}$ denotes the binary pixel value at index i .

For the blind object:

$$M = 5. \quad (51)$$

The mass represents the total number of active pixels and constitutes a rotation- and translation-invariant scalar descriptor.

The stored objects have masses:

$$v^1 = 3 \quad (52)$$

$$v^2 = 5 \quad (53)$$

$$v^3 = 3 \quad (54)$$

Using a strict mass window $\delta = 0$, only object 2 satisfies the equality constraint. Therefore, deterministic elimination alone identifies the correct candidate before any similarity computation.

If elimination is intentionally bypassed, we proceed to cosine similarity evaluation.

4.4. Deterministic Cosine Matching

The Euclidean norms are:

$$\|v^1\| = \sqrt{3} \quad (55)$$

$$\|v^2\| = \sqrt{5} \quad (56)$$

$$\|v^3\| = \sqrt{3} \quad (57)$$

$$\|\mathbf{b}\| = \sqrt{5} \quad (58)$$

Normalized vectors:

$$\tilde{\mathbf{v}}^{(1)} = v^1 / \sqrt{3}, \quad (59)$$

$$\tilde{\mathbf{v}}^{(2)} = v^2 / \sqrt{5}, \quad (60)$$

$$\tilde{\mathbf{v}}^{(3)} = v^3 / \sqrt{3}, \quad (61)$$

$$\tilde{\mathbf{b}} = \mathbf{b} / \sqrt{5}. \quad (62)$$

Since $\mathbf{b} = v^{(2)}$, cosine similarities are computed as

Dot Products

$$\mathbf{v}^{(2)} \cdot \mathbf{v}^{(1)} = 2 \quad (64)$$

$$\mathbf{v}^{(2)} \cdot \mathbf{v}^{(2)} = 5 \quad (65)$$

$$\mathbf{v}^{(2)} \cdot \mathbf{v}^{(3)} = 2 \quad (66)$$

Thus,

$$s_1 = 2 / \sqrt{3} \sqrt{5} \approx 0.516 \quad (67)$$

$$s_2 = 1 \quad (68)$$

$$s_3 = 2 / \sqrt{3} \sqrt{5} \approx 0.516 \quad (69)$$

4.5. Recognition Result

The similarity spectrum is

$$s_1 \approx 0.516 \quad (70)$$

$$s_2 = 1 \quad (71)$$

$$s_3 \approx 0.516 \quad (72)$$

Therefore,

$$\hat{k} = \arg \max_k s_k = 2 \quad (73)$$

The system deterministically identifies object 2 with maximum cosine similarity equal to 1.

Structural Interpretation

This prototype verifies:

- 1) Correct row-wise vectorization
- 2) Proper database matrix construction
- 3) Exact ℓ_2 normalization
- 4) Accurate cosine similarity computation
- 5) Deterministic arg max selection

No gradient descent, no loss minimization, and no probabilistic modeling are involved. The recognition emerges directly from algebraic structure.

5. 7×7 Structured Pattern Library and Experimental Evaluation

5.1. Library Definition

A structured library of 20 deterministic 7×7 binary patterns was constructed.

Each object $\mathbf{A}^{(k)} \in \mathbb{R}^{7 \times 7}$ was flattened row-wise into

$$\mathbf{v}^{(k)} \in \mathbb{R}^{49}. \quad (74)$$

The library contains both symmetric and asymmetric geo-

metric structures in order to create controlled structural diversity without randomness.

The patterns are defined as follows:

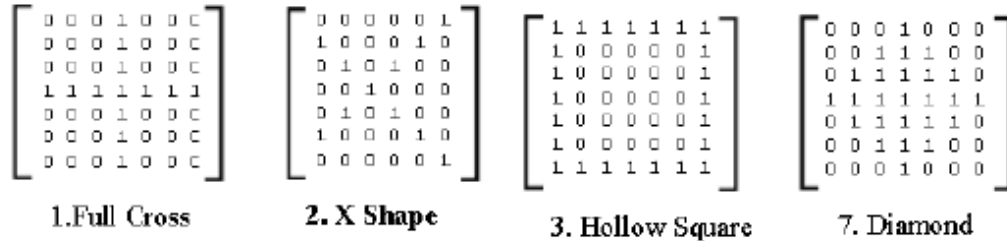


Figure 3. Structured 7×7 deterministic pattern library used for the experimental evaluation.

4. Filled Square — All entries equal to 1.
 5. Vertical Bar — Column 4 = 1, all other entries 0.
 6. Horizontal Bar — Row 4 = 1.
 7. The figure is above.
 8. Small Center Dot — Only (4,4) = 1.
 9. Two Vertical Bars — Columns 2 and 6 = 1.
 10. Two Horizontal Bars — Rows 2 and 6 = 1.
 11. T Shape — Row 1 full + column 4 downward.
 12. Inverted T — Row 7 full + column 4 upward.
 13. I Shape — Column 1 + row 7.
 14. Reverse L — Column 7 + row 7.
 15. U Shape — Columns 1 and 7 + row 7.
 16. Inverted U — Columns 1 and 7 + row 1.
 17. Thick Diagonal — Three-pixel thick main diagonal.
 18. Frame with Cross Inside — Hollow border + center cross.
 19. 3×3 Filled Block Centered — Middle 3×3 region = 1.
 20. Four Corners Only — (1,1), (1,7), (7,1), (7,7) = 1.
- The library matrix is defined as

$$\mathbf{L} = [\mathbf{v}^{(1)} \mathbf{v}^{(2)} \dots \mathbf{v}^{(20)}] \in \mathbb{R}^{49 \times 20} \quad (75)$$

Each object was stored in a single-table database structure with a unique identifier (UID).

No secondary tables or relational joins were used.

The design deliberately includes:

- 1) Purely symmetric structures
- 2) Orientation-sensitive asymmetric shapes
- 3) Sparse and dense patterns
- 4) Corner-only and frame-based geometries ensuring controlled spectral and structural variation.

For clarity, the structured 7×7 pattern library used in the experiments is illustrated in Figure 3, providing a visual reference for the deterministic recognition process described in this section.

5.2. Stage I — Baseline Deterministic Validation

Each of the 20 stored objects was used once as blind input without rotation or perturbation.

Procedure:

- 1) Row-wise flattening
- 2) Unit Euclidean normalization
- 3) Cosine similarity computation

$$\mathbf{s}_k = \tilde{\mathbf{b}}^T \tilde{\mathbf{v}}^{(k)} \quad (76)$$

Decision rule:

$$\hat{k} = \arg \max_k \mathbf{s}_k. \quad (77)$$

Result:

Correct identification was obtained for all tested instances.

This confirms:

- 1) Correct flattening
- 2) Correct database storage
- 3) Proper ℓ_2 normalization
- 4) Deterministic cosine evaluation

No stochastic element or optimization process was involved. Recognition emerges directly from algebraic structure.

5.3. Stage II — Rotation Robustness Evaluation

To evaluate geometric robustness, rotated versions of each object were generated at $\pm 10^\circ$.

The rotated images were re-sampled onto the 7×7 grid and provided as blind inputs.

Two deterministic strategies (Section 3.8) were evaluated.

Strategy A — Alignment-First

For each rotated input:

- 1) Centroid computation
- 2) Covariance matrix estimation
- 3) Principal-axis orientation computation
- $\theta = (\frac{1}{2}) \arctan(2\sigma_{xy} / (\sigma_{xx} - \sigma_{yy}))$
- 4) Coarse rotation by $-\theta$
- 5) Cosine similarity evaluation

No full 360° sweep was performed.

Observation:

For asymmetric patterns, orientation was correctly estimated and cosine similarity peaked at the correct index.

For symmetric patterns, orientation ambiguity did not affect recognition because cosine similarity remained discriminative under symmetry.

Strategy B — Elimination-First

For each rotated input:

Mass window filtering

Radial energy filtering

Optional partition-based elimination

Candidate reduction (20 → approximately 3–5 objects)

Orientation estimation applied only to reduced subset

Cosine similarity evaluation

Observation:

Significant candidate reduction was achieved before any rotational alignment was performed.

Recognition remained correct for all tested cases.

5.4. Efficiency Analysis

Without elimination, similarity is computed as

$$s = L^T b \quad (78)$$

requiring comparison with all 20 objects.

With invariant filtering, the candidate set shrinks geometrically:

$$|C_0| = 20 \quad (79)$$

$$|C_1| \ll 20 \quad (80)$$

$$|C_2| \approx 3-5 \quad (81)$$

Rotation, when applied, is restricted to this reduced subset.

This constitutes structured shrinkage rather than brute-force search.

Structural Interpretation of the 7×7 Experiment

The experiment demonstrates:

- 1) Deterministic recognition without training
- 2) Rotation handling without angular sweep
- 3) Database shrinkage via structural invariants
- 4) Stability governed by spectral separation

Recognition is completed in a single evaluation cycle.

The framework operates through algebraic ordering and invariant filtering rather than iterative convergence.

The objective of this study is to establish and demonstrate the deterministic recognition framework rather than to benchmark large statistical datasets. For this reason, controlled structured pattern libraries (3×3 and 7×7) were intentionally used as prototype systems to verify the invariant filtering mechanism and deterministic cosine matching process in a transparent and reproducible manner.

Since invariant filtering reduces the candidate set prior to similarity evaluation, the effective computational cost grows significantly slower than the library size, which supports scalability for larger structured pattern libraries.

6. Conclusion

This work introduced a deterministic object recognition framework based on structural compatibility and finite ordering rather than iterative optimization.

The identification stage is governed by the structural decision operator

$$\hat{k} = \arg \max_k s_k,$$

which selects the maximal element of a finite compatibility spectrum.

The decision problem is well-posed when:

- 1) The compatibility spectrum is finite (existence)
- 2) A strict dominance condition holds (uniqueness)
- 3) A positive spectral gap exists (stability)

Stability is therefore encoded in structural separation rather than in convergence dynamics. The spectral compatibility gap plays a role analogous to curvature in optimization-based systems, but without requiring descent trajectories or iterative refinement.

Experimental validation on structured 3×3 and 7×7 libraries confirms:

- 1) Correct matrix-to-vector representation
- 2) Deterministic cosine-based matching
- 3) Rotation handling without angular sweep
- 4) Candidate reduction via invariant filtering
- 5) Fully reproducible identification

The framework replaces iterative energy expenditure with structural preconditioning and finite comparison, completing recognition in a single evaluation cycle.

In contrast to training-based approaches, the solution is not approached asymptotically. It is identified directly through algebraic ordering.

The presented methodology demonstrates that small-scale structured recognition can be formulated as a well-posed deterministic decision problem governed by compatibility geometry rather than optimization dynamics.

This work does not seek to replace optimization-based learning systems. Instead, it demonstrates that for structured small-scale recognition problems, deterministic compatibility and finite ordering may suffice without iterative training.

Beyond its experimental validation, the framework clarifies a conceptual point: recognition need not be formulated as trajectory minimization in parameter space. When the object set is finite and structurally separated, identification reduces to a compatibility ordering problem over a discrete spectrum. The operator

$$\hat{k} = \arg \max_k s_k$$

acts as a finite decision selector rather than as the endpoint of an optimization path. Stability is ensured by the spectral compatibility gap $\Delta = s_{(1)} - s_{(2)}$, where $s_{(1)}$ and $s_{(2)}$ denote the

largest and second-largest compatibility values. A strictly positive Δ guarantees robustness under bounded perturbations without requiring iterative correction. In this sense, determination replaces convergence, and ordering replaces descent. The results demonstrate that, for structured small-scale libraries, recognition can be achieved through algebraic separation and invariant filtering rather than through iterative energy minimization.

For structured, low-dimensional libraries, deterministic geometric reasoning combined with invariant filtering provides a computationally controlled alternative to optimization-based methods.

The presented framework also clarifies a conceptual distinction between deterministic recognition and optimization-based learning systems. In the proposed formulation, identification is not obtained through a trajectory of parameter updates but through a finite compatibility ordering over a discrete object library. Once the representation is constructed and the invariants are computed, recognition reduces to selecting the maximal compatibility element in a finite set. This structural interpretation emphasizes that, in controlled environments with finite pattern libraries, object recognition can be formulated as a well-posed deterministic decision problem rather than as an iterative optimization process.

7. Limitations and Future Work

Despite its structural clarity and deterministic guarantees, the present framework operates under controlled assumptions that bound its applicability.

First, the experiments were conducted on structured binary patterns with moderate dimensionality (3×3 and 7×7). While the algebraic formulation extends naturally to higher dimensions, computational scaling and spectral crowding effects require further investigation.

Second, rotation handling relied on principal-axis estimation derived from second-order moments:

$$\theta = (\frac{1}{2}) \arctan(2\sigma_{xy} / (\sigma_{xx} - \sigma_{yy})).$$

This approach is effective for moderate rotations and non-degenerate covariance structures. However, highly symmetric patterns may produce orientation ambiguity, necessitating secondary disambiguation mechanisms.

Third, robustness was evaluated under bounded perturbations and limited resampling noise. Large geometric distortions, severe occlusion, or non-binary intensity variations were not considered in the present study.

Fourth, the framework assumes a finite, explicitly stored object library. The method does not address continuous class generalization or large-scale statistical learning scenarios where model abstraction replaces explicit storage.

Future work may explore:

- 1) Extension to multi-level (non-binary) intensity patterns
- 2) Integration of deterministic partition hierarchies for

larger libraries

- 3) Spectral gap analysis under structured noise models
- 4) Hybrid architectures combining invariant filtering with learned embeddings
- 5) Formal perturbation bounds expressed in terms of Δ and $\|\delta b\|$

A theoretical investigation of spectral separation conditions ensuring uniqueness,

$$\Delta > 0, \quad (82)$$

in higher-dimensional structured spaces would further strengthen the deterministic foundation of the method.

Abbreviations

C_k	Candidate Set at Stage k
CNN	Convolutional Neural Network
DB	Database
$\hat{k}$	Estimated Class Index
k^*	True Class Index
ℓ_1	ℓ_1 Norm (Mass / Zeroth Moment Content)
ℓ_2	Euclidean Norm
PCA	Principal Component Analysis
s_k	Cosine Similarity Score
UID	Unique Identifier
Δ	Spectral Compatibility Gap
Σ	Covariance Matrix

Author Contributions

Huseyin Murat Cekirge: Conceptualization, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing

Conflicts of Interest

The author declares no conflicts of interest.

References

- [1] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000. <https://doi.org/10.1109/34.824819>
- [2] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962. <https://doi.org/10.1109/TIT.1962.1057692>
- [3] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York, NY, USA: Wiley, 1973.
- [4] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2018.

- [5] W. K. Pratt, *Digital Image Processing: PIKS Scientific Inside*, 4th ed. Hoboken, NJ, USA: Wiley, 2007.
- [6] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 4th ed. Stamford, CT, USA: Cengage Learning, 2014.
- [7] B. K. P. Horn, *Robot Vision*. Cambridge, MA, USA: MIT Press, 1986.
- [8] J. P. Lewis, "Fast template matching," in *Vision Interface 1995*, pp. 120–123, 1995.
- [9] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
<https://doi.org/10.1007/978-0-387-45528-0>
- [10] R. Szeliski, *Computer Vision: Algorithms and Applications*. London, U. K.: Springer, 2010.
<https://doi.org/10.1007/978-1-84882-935-0>
- [11] M. R. Teague, "Image analysis via the general theory of moments," *Journal of the Optical Society of America*, vol. 70, no. 8, pp. 920–930, 1980.
<https://doi.org/10.1364/JOSA.70.000920>
- [12] J. Flusser, T. Suk, and B. Zitová, *Moments and Moment Invariants in Pattern Recognition*. Hoboken, NJ, USA: Wiley, 2009.
<https://doi.org/10.1002/9780470684757>
- [13] G. Strang, *Introduction to Linear Algebra*, 5th ed. Wellesley, MA, USA: Wellesley-Cambridge Press, 2016.
- [14] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. Baltimore, MD, USA: Johns Hopkins University Press, 2013.
- [15] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed. Philadelphia, PA, USA: SIAM, 2002.
<https://doi.org/10.1137/1.9780898718027>
- [16] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ, USA: Wiley, 2006.
<https://doi.org/10.1002/047174882X>
- [17] H. M. Cekirge, *Equilibrium-based Deterministic Learning in AI via σ -Regularization*, *American Journal of Artificial Intelligence*, Vol. 10, No. 1, pp. 48–60, 2026.
<https://doi.org/10.11648/j.ajai.20261001.15>