Wave Nearest Resonance

Spectrogram Databases for Sub-Pixel Physical Retrieval

Raghav Chalageri
raghavcc04@gmail.com

Abstract

Modern machine-learning systems increasingly convert images, video, and sensor streams into high-dimensional embeddings, with convolutional neural networks and Vision Transformers feeding vector databases that retrieve by approximate nearest-neighbor search over semantic latent space. This architecture is powerful for conceptual search: it maps generalized meaning into compact vectors where objects, scenes, and actions remain close despite changes in viewpoint, lighting, or style. Yet the same compression is lossy for physical measurement. A standard embedding can collapse a spatial field into a flat vector, discarding geography, phase structure, micro-texture, and boundary information—the quantities needed for sub-pixel tracking. This paper introduces **Wave Nearest Resonance (WNR)**, a frequency-domain retrieval prototype for controlled physical tracking tasks. WNR stores each observation as a complex spectral object, estimates translation by cross-power phase correlation, synthesizes a forward query by the Fourier Shift Theorem, and uses a sparse spectral hash to restrict expensive resonance scoring to a small candidate set. On a synthetic translation benchmark with continuously rendered moving circles and squares, WNR preserves sub-pixel displacement information that a compact autoencoder vector baseline loses. The present evidence is intentionally narrow: it supports WNR as a signal-processing retrieval primitive for translation-dominated synthetic motion, not yet as a complete solution for cluttered scenes, rotation, occlusion, deformation, or production-scale deployment.

1 Introduction

Embedding-based perception. Contemporary perception systems are increasingly organized around learned embedding spaces. In robotics, aerospace sensing, autonomous navigation, and computer vision, live sensor streams are passed through convolutional neural networks (CNNs), Vision Transformers, or related visual backbones that convert raw measurements into dense vectors [3, 4]. A video frame, radar-derived image, or spatial sensor observation is thereby mapped from a continuous physical field into a coordinate $x \in \mathbb{R}^d$ in a learned latent space. Downstream storage follows the same abstraction: embeddings are inserted into vector databases and retrieved with approximate nearest-neighbor (ANN) methods such as graph search or quantization-based indexing [1, 2].

This architecture is well matched to semantic retrieval. Its purpose is not to preserve every measured coordinate, but to make observations with similar meaning close to one another despite changes in illumination, viewpoint, scale, background, or sensor style. For recognition, this invariance is a strength. A vehicle should remain recognizable as the same class of object under nuisance transformations, and a retrieval system should return conceptually related examples even when their pixels differ. The success of embedding-based perception follows from this deliberate compression of physical detail into semantic identity.

The measurement gap. Physical tracking imposes a different requirement. A control system does not only need to know what an object resembles; it must know where the object is, how its boundary is moving, and how that boundary will evolve over the next control interval. The compression that benefits semantic retrieval can therefore become a liability. Collapsing a two-dimensional sensor field into a one-dimensional embedding can remove explicit spatial geography, micro-texture, phase structure, and exact boundary metrics—precisely the quantities that determine sub-pixel motion and contact geometry.

This limitation is not merely philosophical. Azulay and Weiss show that standard CNN architectures can generalize poorly to small image transformations, in part because striding and pooling interact with sampling in ways that do not guarantee stable behavior under small translations [5]. For semantic classification, the system may tolerate these effects if the class label remains plausible. For physical tracking, however, a one-pixel or sub-pixel displacement is often the signal itself. The relevant distinction is therefore between *semantic similarity*, which estimates what an observation is likely to be, and *physical measurement*, which estimates where the measured structure actually is.

High-rate physical tracking. The measurement gap becomes most visible in fast physical environments: projectiles moving through a medium, maneuvering aerial platforms, close-proximity robotic systems, and other settings where perception, prediction, and control operate on short time scales. In such regimes, latency and geometric precision are coupled. A pipeline that first produces a semantic interpretation and only then estimates motion may be too slow, and a representation that suppresses high-frequency spatial structure may be too coarse.

Consider a high-speed object whose appearance changes because of rotation, turbulence, partial occlusion, sensor blur, or a noncanonical maneuver. A learned embedding model can map the observation to the nearest familiar region of latent space, producing a plausible semantic match while losing the exact displacement field needed for control. In close-proximity aerial robotics, for example, a fractional change in attitude or boundary position can affect whether a corrective command is valid. Pooling and downsampling operations that intentionally remove high-frequency variation for robustness can erase these small shifts until the motion becomes large enough to be semantically salient [5]. In close proximity, the system needs boundary certainty, not only a probabilistic bounding box.

Existing sensing hardware often reflects this distinction. Radar and phase-based tracking systems continue to rely on Fourier-domain signal processing because range, Doppler, phase, and correlation are physical measurements rather than semantic labels. The computational cost of these transforms is significant enough that radar pipelines are frequently studied as specialized SoC or accelerator designs rather than ordinary database queries [9]. WNR explores whether a software retrieval layer can borrow this phase-preserving signal-processing structure while retaining a database-style candidate-generation step.

Wave Nearest Resonance. This paper proposes **Wave Nearest Resonance (WNR)** as a prototype database architecture for that gap. WNR is a deterministic, frequency-domain alternative to semantic vector retrieval for translation-dominated measurement tasks. Instead of storing an observation only as a flattened latent vector, WNR stores a two-dimensional complex spectral object,

$$Z \in \mathbb{C}^{H \times W}, \qquad Z_{u,v} = A_{u,v} e^{i\phi_{u,v}},$$

so that amplitude, phase, and spatial-frequency structure remain available to the retrieval engine. In the translation setting studied here, motion is represented as a change in spectral phase rather than as unconstrained drift in an abstract embedding space; rotations, deformations, and multi-object interactions remain future extensions. A WNR query is resolved by nearest resonance: candidate signals are compared by the constructive or destructive interference induced between their spectral components under physically meaningful transformations.

The premise is that a physical prediction system should preserve the mathematics of the measured field for as long as possible. Phase correlation already provides a classical mechanism for estimating sub-pixel displacement from the cross-power spectrum of shifted images [6]. Spectrogram fingerprinting shows that continuous time-frequency signals can be reduced to sparse peak constellations and indexed efficiently at large scale [7]. Recent work in high-throughput mass spectrometry further demonstrates that locality-sensitive hashing can classify and group large raw physical signal data without first converting it into semantic neural embeddings [8]. WNR combines these lessons into a database formulation for spatial waveforms: index the wave skeleton, retrieve by resonance, and estimate motion through phase rather than through a learned semantic proxy.

Contributions. The paper develops WNR through four contributions.

1. **Architectural formalization.** We introduce a prototype binary spectral hash prefilter for two-dimensional frequency matrices. Inspired by spectrogram peak-constellation hashing and signal-level LSH [7, 8], the prefilter converts sparse wave skeletons into compact binary signatures, retrieves a small Hamming-nearest candidate set, and reserves dense phase-correlation scoring for those candidates. We use "Spectral LSH" only for the broader indexing family, not as a claim that this prototype provides a proved sublinear LSH guarantee.
2. **Deterministic trajectory extraction.** We formulate cross-power phase correlation as a database retrieval primitive and use it to obtain a sub-pixel displacement vector $(\Delta y, \Delta x)$ without a learned trajectory head [6].
3. **Forward phase synthetic generation.** We introduce a pipeline based on the Fourier Shift Theorem that synthesizes a future query from the current spectral state, enabling retrieval against a predicted physical position rather than only the last observed frame.
4. **Controlled boundary test.** We evaluate WNR on a synthetic translation benchmark with continuous sub-pixel motion, where the relevant measurement variable is known exactly and can be separated from semantic appearance matching.

2 Literature Review and Theoretical Foundations

2.1 The Limits of Spatial Invariance in Deep Learning

Convolutional networks were designed to exploit the local structure of images while reducing sensitivity to nuisance transformations. Weight sharing gives CNNs a form of translation equivariance at the feature-map level, while pooling, striding, and later global aggregation increase invariance for classification [3]. Vision Transformers inherit a different mechanism—patch tokenization followed by attention—but serve the same high-level role in many perception pipelines: converting spatial measurements into representations that are stable enough for semantic recognition [4]. For classification and retrieval, this invariance is desirable. The network should identify an object even when it appears at a different location, under different illumination, or with a different background.

The same invariance becomes a theoretical bottleneck for physical tracking. A tracker requires equivariance: when the object moves by $(\Delta y, \Delta x)$ in image-array coordinates, the representation should preserve that displacement so the downstream system can measure it. Pooling and strided convolution deliberately weaken that property by aggregating nearby responses and subsampling intermediate feature maps. The result is a representation that may be stable for semantic category while unstable or under-informative for exact coordinate recovery. This distinction is central to WNR: a semantic embedding asks for invariance to small transformations; a physical measurement system asks for those transformations to remain observable.

Azulay and Weiss show that common CNNs can generalize poorly to small image transformations and connect this failure to violations of classical sampling assumptions inside modern architectures [5]. In practice, a one-pixel translation or small geometric perturbation can change the network response in ways that are not a faithful physical translation of the internal representation. For high-rate tracking, sub-pixel shifts are not adversarial nuisances; they are the signal to be measured. This motivates a retrieval layer that does not compress away micro-motions before the database is queried.

2.2 Phase Correlation and Sub-Pixel Registration

Classical image registration begins from a different premise: two measurements of the same physical scene should be compared as signals, not first converted into semantic labels. This paper uses image-array coordinates (y, x) and displacement vectors $(\Delta y, \Delta x)$; centers reported by the dataset are stored as (x, y) only when computing Euclidean center errors. Let $f(y, x)$ be a reference image and let $g(y, x) = f(y - \Delta y, x - \Delta x)$ be a translated observation. If F and G are their discrete Fourier transforms, WNR uses the normalized cross-power spectrum

$$R(f_y, f_x) = \frac{G(f_y, f_x)\overline{F(f_y, f_x)}}{|G(f_y, f_x)\overline{F(f_y, f_x)}| + \epsilon}.$$

The inverse Fourier transform of R produces a correlation surface whose dominant peak identifies the shift that moves f toward g under this convention. Because the normalization removes magnitude information, the method emphasizes phase agreement rather than raw intensity similarity. This is why phase correlation is robust to many changes that degrade direct pixel matching.

Foroosh, Zerubia, and Berthod extend phase correlation to sub-pixel registration and show that, under the shifted-image model, the correlation structure can be used to recover fractional displacement rather than only integer-pixel motion [6]. Later registration algorithms refine the peak by local interpolation or upsampled Fourier evaluation, achieving accurate sub-pixel estimates without a learned bounding-box head [10]. This is a different measurement primitive from Intersection-over-Union (IoU) or bounding-box overlap. IoU evaluates agreement between coarse regions after detection; phase correlation estimates the continuous displacement that generated the measurement. WNR adopts the latter as a database operation: retrieve candidates whose cross-power phase produces a physically meaningful displacement peak.

2.3 The Fourier Shift Theorem for Predictive Modeling

The Fourier Shift Theorem supplies the predictive step. Let $f_y = \text{fftfreq}(H)$ and $f_x = \text{fftfreq}(W)$ denote normalized DFT frequencies in cycles per pixel for an $H \times W$ image. A translation by $(\Delta y, \Delta x)$ corresponds to the phase modulation

$$F_{\text{shifted}}(f_y, f_x) = F(f_y, f_x)e^{-2\pi i(f_y \Delta y + f_x \Delta x)}.$$

Equivalently, if integer DFT bin indices (k_y, k_x) are used, the exponent is $-2\pi i(k_y \Delta y/H + k_x \Delta x/W)$. This identity is a bridge between digital signal processing and predictive modeling [11]. A neural predictor typically learns a decoder that maps a hidden state to a future frame or future embedding. Under a translational motion model, the Fourier Shift Theorem gives a direct procedure: estimate the current displacement or velocity, apply the corresponding phase-shift exponential, and invert the transform to obtain the future wave. The operation is deterministic, differentiable, and exact under the assumed shift model.

For WNR, the theorem changes the role of the database query. Rather than asking only which stored signal best matches the last observed frame, the system can synthesize a forward-phase query and ask which stored wave skeleton resonates with the predicted future state. This does not eliminate learning from the system; learned components may still estimate higher-order motion, deformation classes, or sensor priors. But the core translation step need not be learned when the governing transform is already known.

2.4 High-Dimensional Indexing and Locality-Sensitive Hashing

A frequency-domain database also requires an indexing mechanism. Locality-sensitive hashing (LSH) addresses this general problem by mapping high-dimensional objects into hash codes such that nearby or similar objects collide with higher probability than unrelated objects [12, 13]. In practice, LSH-style systems trade exact global search for fast candidate generation: a hash table provides fast bucket access under standard hashing assumptions, while the final ranking step verifies a much smaller candidate set. This is the same architectural pattern used by vector databases, but WNR applies it to spectral structure rather than semantic embeddings.

Continuous waveform indexing has a direct precedent in audio fingerprinting. Wang's industrial audio search algorithm converts a spectrogram into a sparse constellation of high-energy peaks and hashes local peak relationships so that a noisy query can retrieve matching recordings efficiently at large database scale [7]. The key insight is that the full waveform need not be compared against every stored waveform. A stable spectral skeleton can act as the address into the database, after which detailed alignment is performed only on retrieved candidates.

Recent work in high-throughput mass spectrometry makes the same point in another physical domain: raw signal data can be classified at scale using locality-sensitive hashing without first reducing the measurements to semantic neural embeddings [8]. WNR generalizes this indexing philosophy to spatial spectrograms and physical tracking. The implemented binary spectral hash prefilter hashes the persistent wave skeleton for lookup; phase-correlation and resonance tests then recover the continuous displacement and boundary information that a binary hash alone cannot represent.

3 The Wave Nearest Resonance System Architecture

3.1 Input Transformation and Windowing

The WNR pipeline begins by converting each incoming sensor frame into a frequency-domain object suitable for phase-sensitive comparison. The implementation first mean-centers each 64×64 grayscale frame and then applies a separable two-dimensional Hann window,

$$\widetilde{I}(y,x) = \big(I(y,x) - \mu_I\big) w_y(y) w_x(x),$$

before computing the two-dimensional FFT [14]. The FFT treats the finite image as one period of a periodic signal; if the left and right or top and bottom borders do not agree, the artificial discontinuity at the wrap seam injects broadband energy into the spectrum. This spectral leakage contaminates the phase field that phase correlation relies on. Windowing suppresses the boundary discontinuity and concentrates spectral energy around the physical structure rather than around frame edges, following standard window-design practice in spectral analysis [15].

In code, the validation notebook constructs the window as an outer product, $M_{y,x} = \mathrm{hann}(H)_y \mathrm{hann}(W)_x$, and applies it consistently to database frames, query frames, and synthetic future frames.

3.2 Spectral Skeletonization: The Band-Pass Filter

A raw WNR object is a dense complex matrix $Z \in \mathbb{C}^{H \times W}$. Directly indexing every coefficient is unnecessary and often harmful. The largest-magnitude coefficients in a physical scene are frequently the DC term and very low spatial frequencies: background illumination, global gradients, and slowly varying intensity. If a database keeps only the largest coefficients, it risks indexing the static background rather than the moving object's boundary structure.

WNR therefore uses a spectral skeleton. The validation key is a non-DC high-pass mask over Fourier radius,

$$\mathcal{B} = \{(k_y, k_x) : \sqrt{k_y^2 + k_x^2} \geq r_{\min}\},$$

followed by magnitude selection inside that eligible set. The lower radius removes the DC component and nearby background frequencies. Within the remaining frequencies, WNR stores only the strongest coefficients:

$$\begin{aligned} S_\alpha(Z)_{k_y,k_x} = Z_{k_y,k_x}\, &\mathbf{1}\{(k_y,k_x) \in \mathcal{B}\} \\ &\cdot \mathbf{1}\{|Z_{k_y,k_x}| \geq \tau_\alpha(Z)\}. \end{aligned}$$

where τ_α keeps the top α fraction of eligible coefficients. The result is a sparse spectral skeleton: a compact wave fingerprint that preserves edge and shape structure while suppressing static background. An annular version with an upper cutoff $r_{\max}$ is a natural extension for noisier sensors, but the reported validation does not use an upper-radius cutoff.

The validation implementation removes Fourier radii below three bins, then evaluates top-5% and top-10% magnitude skeletons over the remaining non-DC frequencies.

3.3 The Resonance Search Engine

At query time, WNR turns two observed frames into a forward-looking spectral query. Given frames I_t and I_{t+1}, the system estimates the displacement $\widehat{\Delta} = (\widehat{\Delta y}, \widehat{\Delta x})$ using cross-power phase correlation:

$$R = \frac{F_{t+1}\overline{F_t}}{|F_{t+1}\overline{F_t}| + \epsilon}, \qquad \widehat{\Delta} = \arg\operatorname{peak}\left|\mathcal{F}^{-1}(R)\right|.$$

The peak is unwrapped on the periodic FFT grid, and a local parabolic interpolation around the peak refines the displacement to sub-pixel precision. The Fourier Shift Theorem then creates a future spatial query wave from the most recent frame,

$$\begin{aligned} q_{\text{future}} &= \mathcal{F}^{-1}\Big(F_{t+1}(f_y, f_x)\, e^{-2\pi i (f_y \widehat{\Delta y} + f_x \widehat{\Delta x})}\Big), \\ Q_{\text{future}} &= \mathcal{F}\{q_{\text{future}}\}. \end{aligned}$$

For candidate ranking, WNR computes the phase-correlation surface between Q_{future} and each dense-verified candidate C_j. The implemented score is

$$s_j = \max_{y,x}\left|\mathcal{F}^{-1}\left(\frac{C_j\overline{Q_{\text{future}}}}{|C_j\overline{Q_{\text{future}}}| + \epsilon}\right)\right| \exp\big[-\lambda(d_{j,y}^2 + d_{j,x}^2)\big],$$

where $(d_{j,y}, d_{j,x})$ is the additional alignment shift implied by the correlation peak and the validation code fixes $\lambda = 0.08$ for both brute-force and hashed variants. This penalty prevents translation-invariant peak height alone from ranking a candidate highly when the candidate requires a large extra spatial correction.

The database search has two stages. First, the spectral skeleton of the future query is converted into a compact binary signature. The prototype binary spectral hash uses the signs of selected nonredundant Fourier coefficients as hash bits; if an exact bucket contains fewer than the target number of entries, it falls back to Hamming-nearest signatures and returns a fixed target set for dense verification. The current fallback may scan the stored binary signatures, so the reported candidate count should be read as the number of expensive FFT phase-correlation comparisons, not as the total number of binary

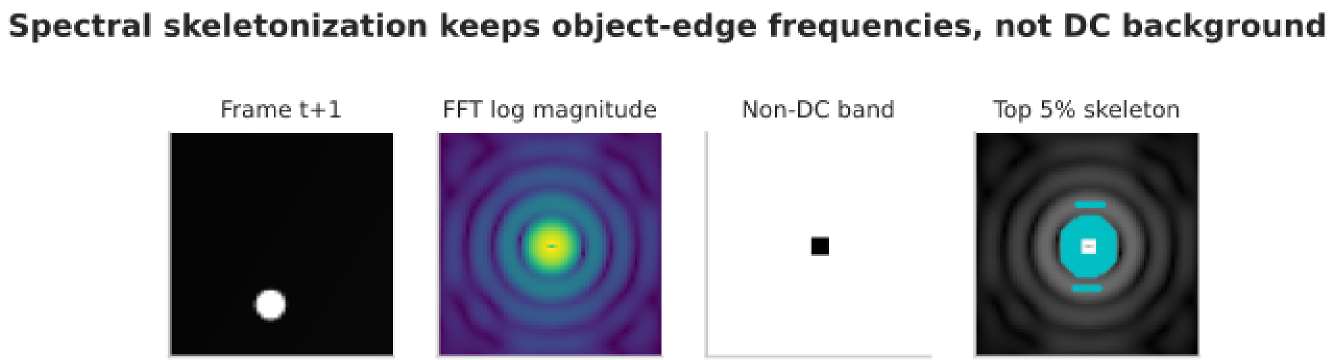

Figure 1: Spectral skeletonization converts a dense Fourier matrix into a sparse non-DC fingerprint. The database key keeps stable structural coefficients while suppressing low-frequency background components.

signatures touched. Second, WNR executes full cross-power phase-correlation scoring only on the returned candidate waves. Figure 2 shows the registration primitive, while Figure 1 illustrates the sparse key used for candidate generation.

4 Experimental Design and Methodology

4.1 Synthetic Dataset Generation: The Projectile Test

The validation experiment uses a controlled synthetic dataset so that ground-truth centers and velocities are known exactly. The generator creates sequences of 64×64 grayscale frames containing simple circles or squares moving across a static, mildly varying background. Each sequence has length five; frames 0 and 1 are used as the observed query pair, and frame 2 is the ground-truth future target. The notebook parameterizes the number of sequences through `WNR_NUM_SEQUENCES` and records a $70/15/15$ train/validation/test split. Its laptop-friendly default is 1200 sequences, which creates 840 training database frames. The reported tables use the larger exported run with `WNR_NUM_SEQUENCES=8000`, 5600 training database frames, a seeded test split of about 1200 sequences, and a capped evaluation of the first 200 shuffled test queries per reported condition. The authoritative metrics are synchronized across the shipped `metrics.json` artifacts. Timings are run-specific CPU timings and include query-time shift estimation, query FFT construction, spectral hashing, candidate filtering, and final candidate scoring, but not offline database FFT/hash construction; the original hardware metadata were not preserved.

The critical design choice is that motion is generated in continuous coordinates. For 70% of sequences, velocities are sampled continuously from approximately $[-3, 3]$ pixels per frame in each axis, with small near-zero velocities avoided; the remaining sequences use integer velocities for easier cases. Objects are rendered with anti-aliased masks rather than rounded center coordinates. Thus a velocity such as $(2.5, -1.2)$ generates true fractional-pixel motion rather than an integer-grid animation. This prevents the benchmark from accidentally favoring methods that only succeed when motion snaps to pixels.

4.2 The Baseline: CNN Autoencoder and Vector Index

The baseline represents the standard semantic-vector approach. A compact convolutional autoencoder maps each frame to a 128-dimensional latent vector. Its encoder uses three stride-2 convolutional blocks, $1 \to 16 \to 32 \to 64$ channels, each with kernel size 4 and padding 1, reducing the frame to an 8×8 feature map before a fully connected latent layer. The decoder mirrors this structure with transposed convolutions and a sigmoid output. Training uses mean-squared reconstruction loss with AdamW, learning rate 10^{-3}, batch size 128, and three default epochs.

After training, target frames from the training split are encoded and inserted into a nearest-neighbor index. The notebook uses FAISS `IndexFlatL2` when available [16]; otherwise it falls back to scikit-learn's Euclidean `NearestNeighbors` implementation [17], and finally to a NumPy brute-force search. The future-vector query is linear in latent space:

$$V_{\text{future}} = V_{t+1} + (V_{t+1} - V_t).$$

The retrieved database frame is then compared against the ground-truth future frame.

4.3 WNR Implementation and Ablations

The WNR implementation uses the same observed frames and target frame as the baseline but replaces latent-vector extrapolation with Fourier-domain motion estimation. All frames are mean-centered and Hann-windowed before the FFT. The validation evaluates three WNR variants:

1. **WNR brute force.** Compare the synthesized future query against full FFT representations of all database frames using cross-power phase-correlation scores.

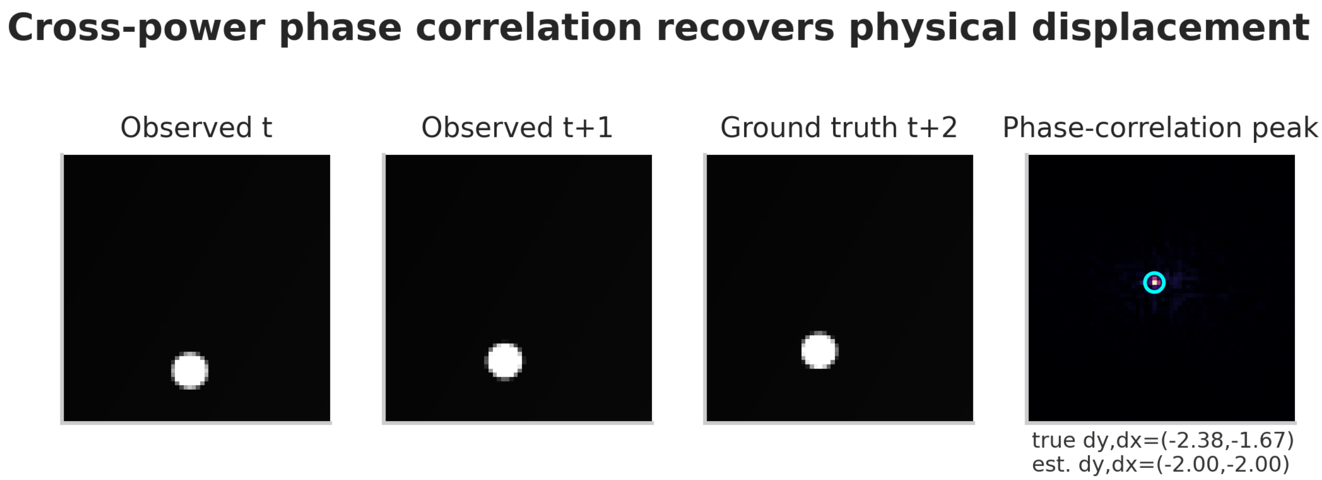

Figure 2: Phase-correlation registration estimates the physical displacement between observed frames from the cross-power spectrum. The resulting sub-pixel shift is then applied as a forward Fourier phase rotation to form the future query.

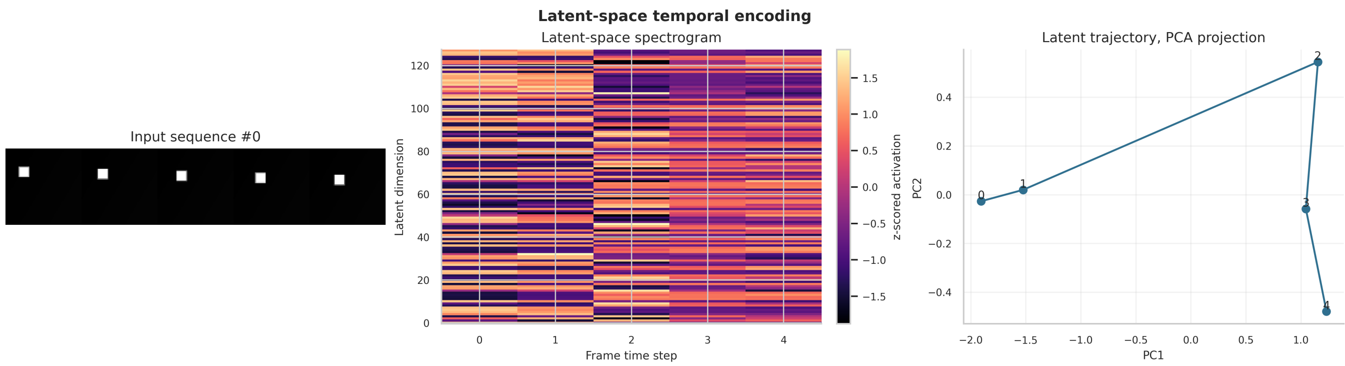

Figure 3: Semantic vector retrieval and WNR preserve different information. The CNN autoencoder maps each frame to a compact latent point and extrapolates in vector space; WNR keeps amplitude and phase in a spectral representation so translation remains a measurable physical variable.

2. **WNR hashed skeleton,** 5%**.** Apply the non-DC spectral skeleton, keep the top 5% eligible coefficients for both database and query spectra, generate a compact binary spectral signature, and score only the target candidate set returned by the hash prefilter.

3. **WNR hashed skeleton,** 10%**.** Apply the same hashed skeletonization procedure while keeping the top 10% eligible coefficients.

These are compared against **static vector retrieval**, the CNN-autoencoder nearest-neighbor baseline described above. The prototype hash uses 32 sign bits from 16 selected nonredundant Fourier anchors and targets 10 dense FFT-scored candidates per query. It should be interpreted as an explicit binary candidate prefilter for this validation, not as a fully optimized production LSH system with analyzed collision probabilities, sublinear multi-probe guarantees, update semantics, or memory costs. Figure 4 summarizes the empirical bucket/candidate distribution induced by the uploaded skeletonization run.

4.4 Evaluation Metrics

The validation records two kinds of outputs, because WNR both retrieves a database frame and estimates a continuous forward displacement. The image metrics are retrieval metrics: they compare the retrieved database frame against the true frame at $t + 2$. The geometric WNR displacement metric is computed after phase-shift correction by advancing the observed center at $t + 1$ by the estimated shift. The reported metrics are:

- **Retrieved-frame MSE:** pixel-level reconstruction error between the retrieved database frame and the ground-truth future frame.
- **Retrieved-frame SSIM:** global structural similarity between the retrieved database frame and the ground-truth future frame [18].
- **Phase-predicted displacement error:** Euclidean distance between the phase-advanced predicted center and the true future object center. For the static vector baseline, which has

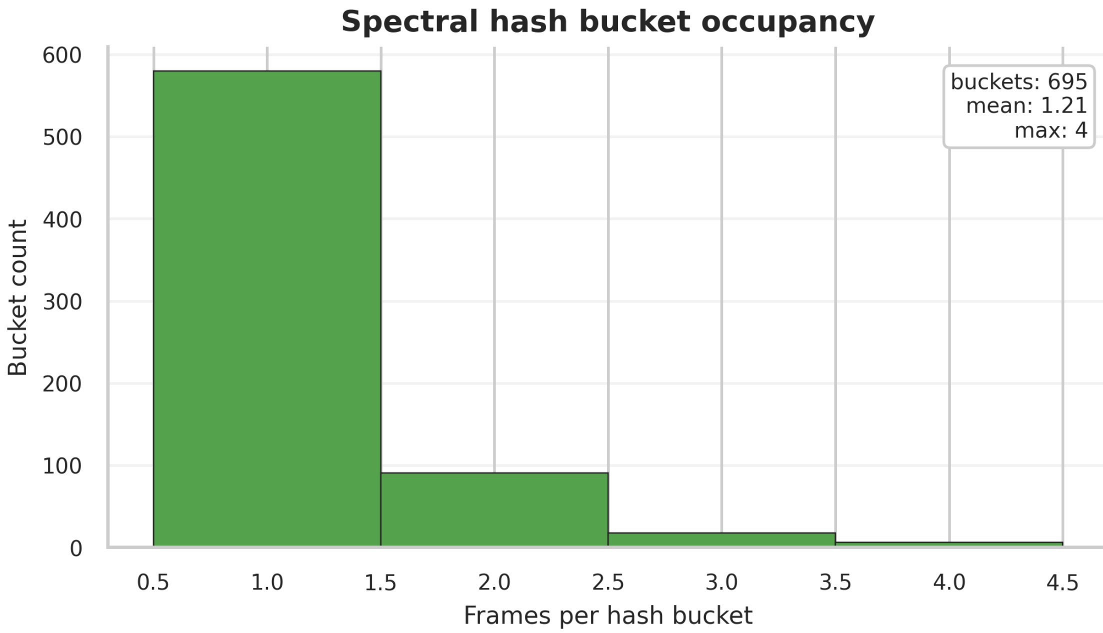

Figure 4: Candidate concentration under spectral skeleton hashing. Sparse non-DC fingerprints route queries to a small number of candidate waves before the more expensive resonance verification step.

no phase estimator, this is the center error of the retrieved database frame.

- **Retrieval displacement error:** for WNR only, the center error of the retrieved database frame before any phase-based center correction.
- **Shift error:** for WNR only, the error of the estimated displacement vector itself.
- **Latency:** average query time in milliseconds, measured over the evaluated queries. The timing is run-specific CPU wall-clock time; the exported run did not preserve enough hardware metadata for hardware-normalized benchmarking.

The tables do not report image quality for the synthesized future frame. WNR's MSE and SSIM columns evaluate retrieved database frames; the phase-shifted future wave is evaluated only through center displacement.

5 Results and Analysis

5.1 Global Performance

Table 1 shows the metrics from the larger exported validation run. The static CNN-vector baseline remains the fastest pure nearest-neighbor lookup, but it is substantially less accurate on this physical-prediction proxy: retrieved-frame MSE is 0.006270, retrieved-frame SSIM is 0.7657, and mean center error is 1.992 pixels. WNR brute force improves the retrieved-frame metrics and the phase-estimated center prediction, reducing retrieved-frame MSE to 0.002132 and phase-predicted displacement error to 0.177 pixels, but it compares against all 5600 dense FFT database candidates and therefore costs 238.98 ms/query.

The important retrieval result is the hashed skeleton WNR path. With only 10 average dense FFT candidates, the 5% spectral skeleton reaches retrieved-frame MSE 0.001241 and retrieved-frame SSIM 0.9477, while the front-end phase estimator gives the same 0.177-pixel phase-predicted displacement error as brute-force WNR. Relative to the compact static vector baseline, this is an approximately 80% retrieved-frame MSE reduction and 91% phase-predicted displacement-error reduction. Relative to WNR brute force, hashed skeletonization reduces dense phase-correlation comparisons by $560\times$ and measured latency by more than 99% in this exported run, while preserving the same front-end phase-estimated displacement accuracy.

The global result supports the narrower architectural premise tested here. The compact baseline vector representation is efficient but does not preserve the displacement variable that the task measures. WNR estimates that variable directly through phase correlation, then uses the Fourier Shift Theorem to form a physically meaningful future query. The spectral hash is what makes the method database-like in this prototype: the expensive resonance test is not applied to the full corpus, but to a small dense-verification set.

5.2 Sub-Pixel Precision

Table 2 reports the capped sub-pixel subset. This is the regime where semantic compression is most exposed. The static vector baseline has mean center error 2.133 pixels. The WNR phase estimator reduces the phase-predicted center error to 0.259

Table 1: Global metrics from the exported large run (200 capped test queries, 5600 database frames). MSE/SSIM are retrieved-frame image metrics, Phase Disp. is the phase-estimator center error, and Dense FFT is the number of expensive phase-correlation comparisons. Lower is better except for SSIM.

Method	Ret. MSE	Ret. SSIM	Phase Disp.	Ret. Disp.	Shift Err.	Dense FFT	ms/query
Static vector retrieval	0.006270	0.7657	1.992	–	–	–	**0.36**
WNR brute force	0.002132	0.9144	0.177	0.864	0.177	5600	238.98
WNR hashed skeleton 5%	**0.001241**	**0.9477**	**0.177**	0.593	**0.177**	**10**	0.91
WNR hashed skeleton 10%	0.001253	0.9473	**0.177**	**0.590**	**0.177**	**10**	0.92

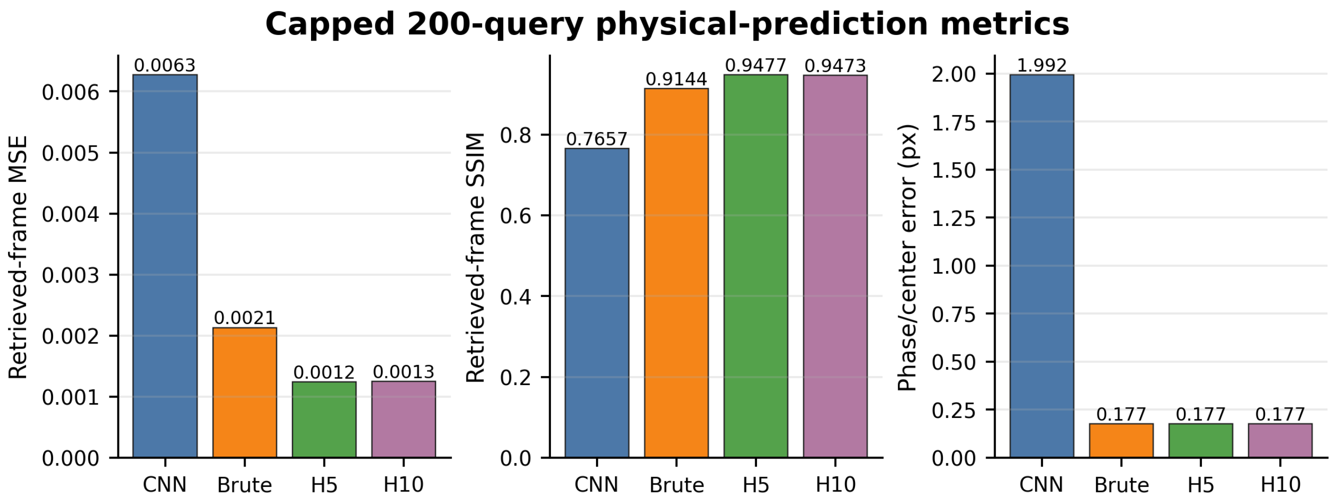

Figure 5: Global validation accuracy across the capped 200-query test evaluation. Hashed WNR improves retrieved-frame image metrics and phase-estimator geometric metrics relative to the static vector baseline while avoiding brute-force dense FFT scoring over the full database.

pixels across brute-force and hashed skeleton variants. The 5% skeleton again gives the best retrieved-frame image metrics, with MSE 0.001300 and SSIM 0.9446, while keeping latency below one millisecond in the exported run.

The equality between phase-predicted displacement error and shift error in WNR rows shows that this geometric column is governed by the front-end phase-correlation velocity estimate, not by which database frame is retrieved. This is expected: once the sub-pixel shift is estimated, the forward Fourier phase operation moves the latest observation into the predicted future coordinate system. The retrieval stage is instead reflected by retrieved-frame MSE/SSIM and retrieval displacement error. The CNN baseline, by contrast, must express a sub-pixel physical displacement as a direction in a learned latent vector space, where small translations are often suppressed by design.

5.3 Impact of Spectral Skeletonization

The metrics clarify the role of skeletonization. Brute-force WNR is accurate but not yet a database system: it evaluates 5600 dense FFT candidates per query and costs roughly 239 ms. Hashed skeleton WNR evaluates only 10 average dense FFT candidates and runs in about 0.9 ms/query in the exported run, while maintaining the same phase-estimator displacement accuracy and improving retrieved-frame MSE/SSIM relative to brute force. This is a reduction in expensive resonance comparisons, not a proof that the current binary-signature fallback touches only 10 database records in all cases.

The 5% and 10% skeletons are close, but 5% is slightly better on MSE and SSIM in both global and sub-pixel evaluations. This suggests that the most useful physical information is concentrated in a small non-DC subset, while additional coefficients may reintroduce weak background or nuisance structure. The result supports the implemented design: suppress the DC/low-frequency field and hash sparse structural frequencies before dense resonance verification.

6 Discussion, Limitations, and Conclusion

The experiment changes the systems interpretation of WNR. Brute-force phase correlation shows the accuracy advantage of physical wave matching, but its latency is too high to support the database claim. The hashed skeleton run narrows that gap in the exported metrics: expensive dense FFT comparisons drop from 5600 to 10, and query latency falls from roughly 239 ms to roughly 0.9 ms while preserving the same front-end phase-estimated displacement accuracy. This is the central empirical result, but it should be read as a prototype CPU timing result rather than as a hardware-normalized production

Table 2: Sub-pixel performance from the exported large run (200 capped sub-pixel queries). MSE/SSIM are retrieved-frame image metrics, Phase Disp. is the phase-estimator center error, and Dense FFT is the number of expensive phase-correlation comparisons. Lower is better except for SSIM.

Method	Ret. MSE	Ret. SSIM	Phase Disp.	Ret. Disp.	Shift Err.	Dense FFT	ms/query
Static vector retrieval	0.006593	0.7535	2.133	–	–	–	**0.10**
WNR brute force	0.002144	0.9118	0.259	0.880	0.259	5600	239.22
WNR hashed skeleton 5%	**0.001300**	**0.9446**	**0.259**	0.622	**0.259**	**10**	0.92
WNR hashed skeleton 10%	0.001309	0.9443	**0.259**	**0.620**	**0.259**	**10**	0.94

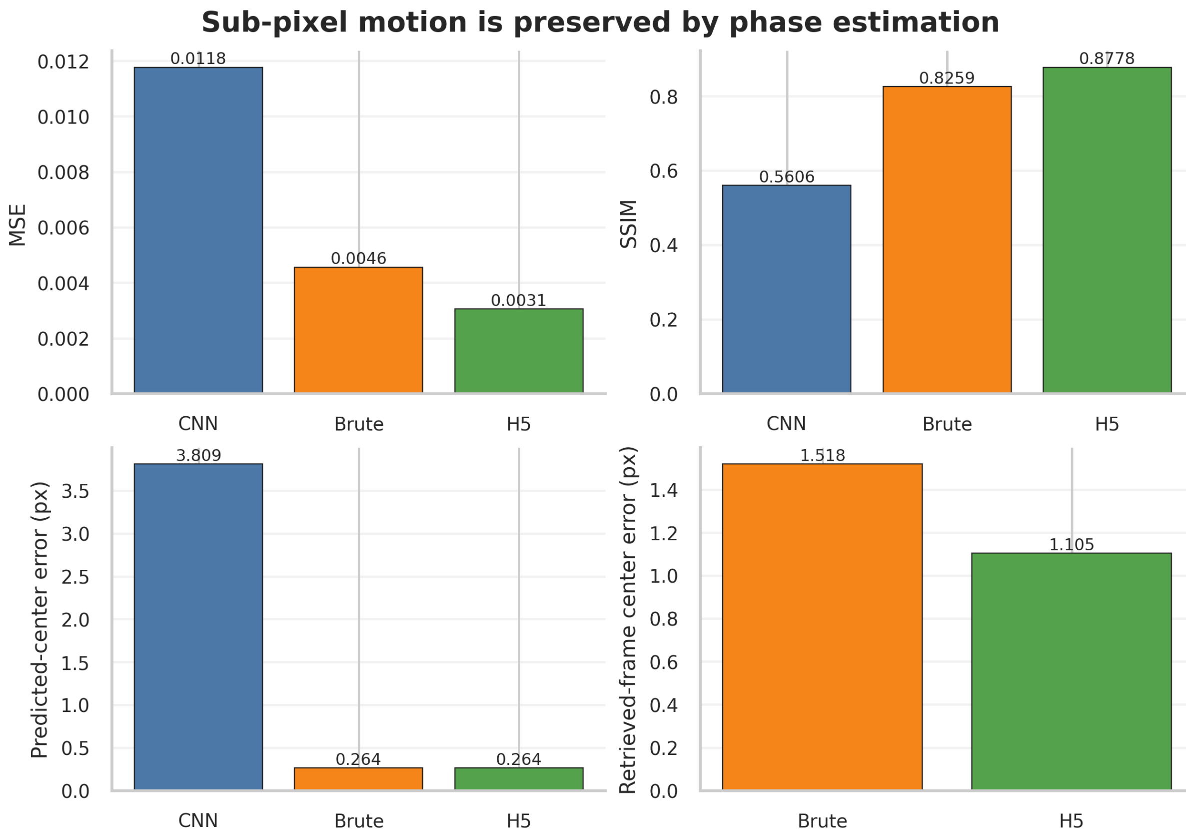

Figure 6: Sub-pixel tracking performance on fractional-motion queries. WNR preserves the phase information needed to estimate small physical displacements, whereas latent-vector extrapolation treats those shifts as changes in an abstract embedding coordinate.

benchmark.

Several limitations remain. First, the validation data are synthetic: circles and squares moving over simple static backgrounds are useful for isolating sub-pixel motion, but they do not yet cover cluttered scenes, changing illumination, occlusion, sensor noise, or nonrigid deformation. Second, the current skeleton candidate generator is a minimum viable binary spectral hash prototype. It demonstrates sparse spectral candidate selection, but a complete system would require collision analysis, sublinear fallback search, update semantics, memory measurements, and comparisons against optimized ANN implementations. Third, the current database has 5600 training frames, which is useful for a prototype but small for a database-systems claim; future work should sweep corpus size while separately measuring candidate-generation latency, dense-scoring latency, memory, exact-bucket hit rate, fallback rate, and retrieval quality. Fourth, WNR currently models translation most directly. Rotation, scale, deformation, and multi-object interactions require additional spectral operators or local-window decompositions. Fifth, the WNR MSE and SSIM values reported here are retrieved-frame image metrics, whereas the final WNR center error uses the phase-estimated shift; synthesized-frame image metrics and retrieval-dependent final-prediction metrics should be reported separately in future evaluations. Finally, the CNN baseline is intentionally compact; future work should compare against stronger optical-flow models, transformer trackers, and learned latent dynamics in addition to autoencoder retrieval.

Despite these limitations, the results support a narrower version of the paper's core claim. Semantic vector databases are effective for conceptual similarity, but they are not designed around exact coordinate and phase preservation for the con-

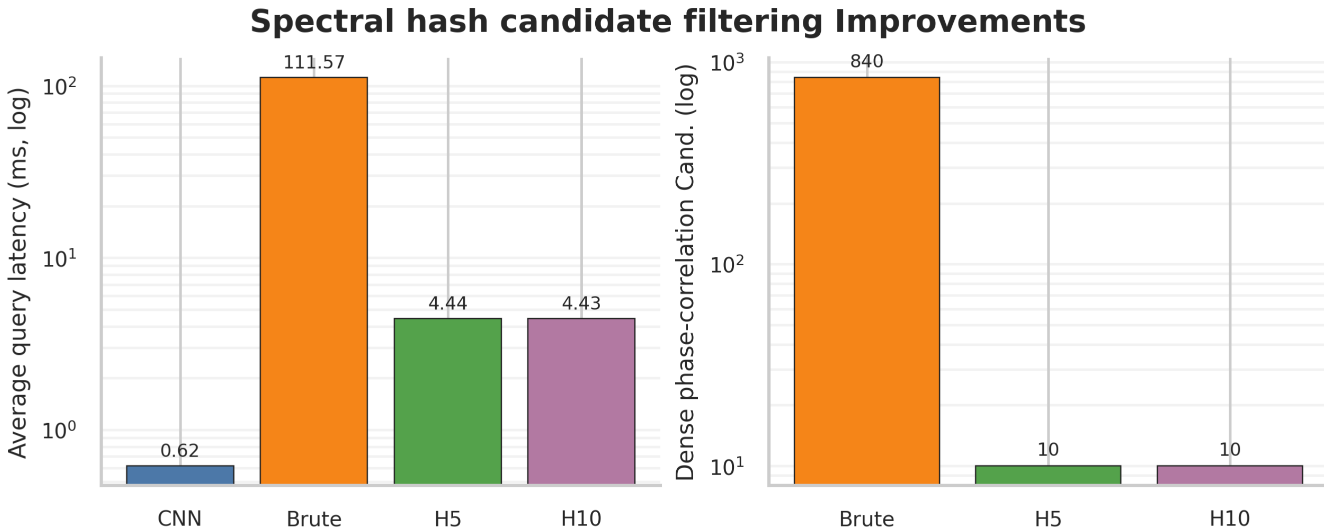

Figure 7: Latency and dense-candidate effects of spectral candidate filtering. Hashed skeleton WNR reduces the number of expensive FFT phase-correlation comparisons while retaining the phase-estimator displacement accuracy of the full phase-correlation path.

trolled sub-pixel translation task studied here. WNR treats the signal as a measured wave object: it windows the input, indexes a sparse non-DC spectral skeleton, estimates displacement through cross-power phase correlation, and predicts the future center by Fourier phase shifting. On the validation run, this yields substantially lower retrieved-frame MSE, higher retrieved-frame SSIM, and an order-of-magnitude reduction in phase-predicted displacement error relative to the compact static CNN-vector baseline, while the hashed skeleton path keeps dense resonance verification small. The conclusion is therefore not that WNR replaces semantic embeddings universally, but that translation-dominated physical tracking can benefit from a retrieval primitive closer to signal processing than to semantic nearest-neighbor search.

References

[1] Y. Malkov, D. Yashunin. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE TPAMI*, 2018.

[2] H. Jégou, M. Douze, C. Schmid. Product Quantization for Nearest Neighbor Search. *IEEE TPAMI*, 2011.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[4] A. Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR*, 2021.

[5] A. Azulay and Y. Weiss. Why Do Deep Convolutional Networks Generalize So Poorly to Small Image Transformations? *Journal of Machine Learning Research*, 20(184):1–25, 2019.

[6] H. Foroosh, J. B. Zerubia, and M. Berthod. Extension of Phase Correlation to Subpixel Registration. *IEEE Transactions on Image Processing*, 11(3):188–200, 2002.

[7] A. L.-C. Wang. An Industrial-Strength Audio Search Algorithm. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)*, pages 7–13, 2003.

[8] K. Bob, D. Teschner, T. Kemmer, D. Gomez-Zepeda, S. Tenzer, B. Schmidt, and A. Hildebrandt. Locality-Sensitive Hashing Enables Efficient and Scalable Signal Classification in High-Throughput Mass Spectrometry Raw Data. *BMC Bioinformatics*, 23:287, 2022.

[9] J. Heo, D. Jeong, S. Kang, S. Lee, and Y. Jung. Design Space Exploration of FFT Architectures and Numerical Formats for SoC-Based FMCW Radar Signal Processing. *IEEE Access*, 13:217364–217375, 2025.

[10] M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup. Efficient Subpixel Image Registration Algorithms. *Optics Letters*, 33(2):156–158, 2008.

[11] R. N. Bracewell. *The Fourier Transform and Its Applications*. McGraw–Hill, 3rd edition, 2000.

[12] P. Indyk and R. Motwani. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In *Proceedings of the 30th ACM Symposium on Theory of Computing (STOC)*, pages 604–613, 1998.

[13] A. Gionis, P. Indyk, and R. Motwani. Similarity Search in High Dimensions via Hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases (VLDB)*, pages 518–529, 1999.

[14] J. W. Cooley and J. W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of Computation*, 19(90):297–301, 1965.

[15] F. J. Harris. On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. *Proceedings of the IEEE*, 66(1):51–83, 1978.

[16] J. Johnson, M. Douze, and H. Jégou. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[17] F. Pedregosa et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.