

Scientific Project

Digital-Optical Implementation of Quantum-Inspired and Classical Classifiers with a Joint Transform Correlator

João Guilherme

Supervisors:

Emmanuel Cruzeiro

Hugo Terças

May 17, 2025

abc

I. INTRODUCTION

Quantum-inspired algorithms are classical routines that use mathematical structures and physical intuitions from quantum mechanics — such as superposition-style probability amplitudes, interference or quantum state discrimination — yet run entirely on conventional hardware. These algorithms have multiple applications, including machine learning, where they promise faster convergence or reduced dimensional dependence compared with traditional methods.

Some machine-learning algorithms are based on distance. Here, the decision rule depends on the similarity between a query sample and a set of prototypes, typically evaluated through dot products or euclidean distance calculation. Although straightforward in software, these operations become a computational bottleneck for high-dimensional data streams or large datasets. There have been optical implementations of classical classification algorithms ([5], [3]). These implementations offer a compelling alternative: coherent light can perform calculations at the speed of propagation, exploiting spatial parallelism while consuming only milliwatts of power.

Among the various optical architectures, the *Joint Transform Correlator* (JTC) ([6], [2]) stands out for its simplicity and adaptability. Unlike classical VanderLugt systems ([8]), the JTC does not require a pre-fabricated filter; instead, the reference and query patterns are placed side by side in the input plane. A single Fourier transform—implemented with a lens—produces an output intensity whose off-axis terms encode the cross-correlation between the two patterns. When feature vectors are encoded as two-dimensional phase distributions, the correlation peak height provides a direct measure of their similarity, which can be mapped to a distance-based decision rule.

In this work we propose optical implementations of the Quantum-Inspired and Classical Nearest Mean Classifier ([1], [7]), and Radial Basis Function Classifier ([5], [3]), that use a single SLM and a 1-f lens system ([4]). We begin by reviewing the theoretical connection between optical correlation and amplitude-based similarity measures. We then describe our experimental set-up, which integrates a single reflective, phase only, spatial light modulator and a CMOS camera to implement a proof-of-concept optical classifier (Section ??). Finally, we benchmark the models with the MNIST dataset and compare their performances with purely electronic implementations, highlighting the trade-offs in speed, energy, and classification accuracy (Section ??).

i. Machine learning and supervised learning

Machine learning (ML) provides algorithmic tools that *learn* directly from data rather than relying on hand-crafted rules. In the *supervised* setting, learning is guided by *examples* whose correct answers are known in advance. We review the key ingredients used throughout this work.

Input data. We start from N raw samples $\mathbf{u}_i \in \mathbb{R}^p$, $i = 1, \dots, N$, each accompanied by a *label* $y_i \in \{1, \dots, K\}$. The labels partition the data into

$$C_k = \{\mathbf{u}_i : y_i = k\}, \quad N_k = |C_k|, \quad p_k = \frac{N_k}{N},$$

where N_k and p_k are the size and empirical probability of class k , respectively. It is convenient to organise the entire data set as a matrix $\mathbf{M} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_N] \in \mathbb{R}^{p \times N}$ and the labels as a vector $\mathbf{y} = [y_1, \dots, y_N]^T$.

Training and test sets. A *training set* ($\mathbf{M}_{\text{train}}, \mathbf{y}_{\text{train}}$) is used to tune the parameters of a classifier, while a disjoint *test set* ($\mathbf{M}_{\text{test}}, \mathbf{y}_{\text{test}}$) gauges its ability to generalise to unseen data. The algorithm is trained on the training set and evaluated on the test set. In this work we will be using the MNIST dataset (Fig.1), with 60000 training samples and 10000 testing samples.

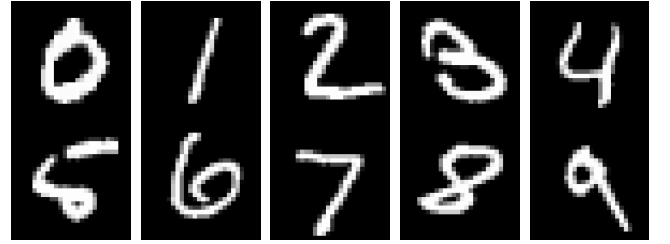


Figure 1: One representative example per class from the MNIST dataset. Each 28×28 grayscale image is flattened to a $p = 784$ -dimensional feature vector before the pre-processing pipeline (centering, standardisation, PCA).

Pre-processing. Real-world data rarely arrive in a form that is optimal for learning. We apply three light transformations that are standard in pattern recognition:

- Centring* removes global offsets, $\mathbf{u}_i \mapsto \mathbf{u}_i - \bar{\mathbf{u}}$, where $\bar{\mathbf{u}} = N^{-1} \sum_{j=1}^N \mathbf{u}_j$.
- Standardisation* rescales each feature so that every dimension has unit variance, preventing attributes with large numerical ranges from dominating the learning process.
- Principal-component analysis* (PCA) projects the centred, standardised vectors onto the $p' (< p)$ directions of greatest variance, $\mathbf{x}_i = \Pi_{\text{PCA}} \mathbf{u}_i \in \mathbb{R}^{p'}$, thereby discarding redundant noise and shrinking the effective dimensionality.

Balanced accuracy. Performance is summarised through the *confusion matrix* $V_{k'k}$, whose entry counts how many test samples belonging to class k are classified as k' . The *balanced accuracy* (BA)

$$\text{BA} = \frac{1}{K} \sum_{k=1}^K \frac{V_{kk}}{N_k}$$

computes the mean of the per-class accuracies and is robust to class imbalance. When classes are homogeneous in size, BA reduces to the usual accuracy.

Classical Nearest-Mean Classifier (CNM-C). The nearest-mean classifier (NMC) is a simple, interpretable classifier that assigns a label to a test sample based on the *centroid* of the training samples in each class (Fig. 2). The *centroid* of class k is defined as the mean of the training samples in that class:

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \mathbf{u}_i, \quad \text{where } \mathbf{u}_i \in C_k.$$

The CNM classifier assigns a test sample \mathbf{x} to the class j if:

$$d(\mathbf{x}, \mu_j) = \min_{k=1, \dots, K} d(\mathbf{x}, \mu_k),$$

where the distance $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$ is the Euclidean distance. The CNM classifier is simple to implement, as it only requires the computation of the centroids and a single distance calculation for each test sample.

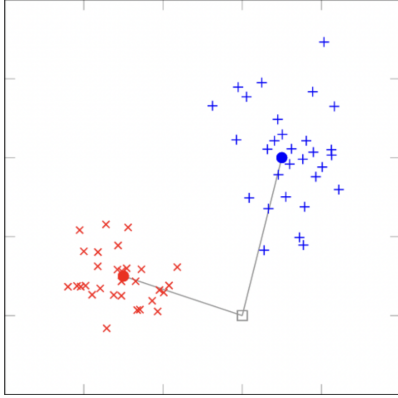


Figure 2: Illustration of the CNM-C decision rule. The training set is split into two classes and the centroids are computed. The nearest centroid to the test sample is selected as the predicted class.

ii. Quantum-inspired classification

Quantum-inspired learning borrows mathematical objects native to quantum theory—*density operators*, *trace distance*, *superposition*—but applies them on ordinary CPUs/GPUs. By embedding classical feature vectors in a Hilbert space, one can exploit the geometry of quantum states to obtain alternative similarity measures that sometimes improve accuracy or robustness compared with purely Euclidean methods [7].

Density-pattern encoding. Given a pre-processed feature vector $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$, map it to a $(d+1)$ -dimensional *density pattern* $\rho_{\mathbf{x}}$ (a positive, unit-trace matrix) via the *stereographic projection*:

$$\tilde{\mathbf{x}} = \frac{1}{1 + \|\mathbf{x}\|^2} (2x_1, \dots, 2x_d, \|\mathbf{x}\|^2 - 1), \quad (1)$$

$$\rho_{\mathbf{x}} = \tilde{\mathbf{x}} \tilde{\mathbf{x}}^\top, \quad \rho_{\mathbf{x}}^2 = \rho_{\mathbf{x}}. \quad (2)$$

The outer product guarantees $\rho_{\mathbf{x}}$ is a *pure* quantum state. Other encodings (e.g. the *informative* map that appends the norm itself as an extra coordinate) are possible; the best choice is often dataset-dependent.

Quantum centroid. For each class k with training subset $\mathcal{S}_k^{\text{tr}}$, define its *quantum centroid* as the uniform mixture

$$q_k = \frac{1}{|\mathcal{S}_k^{\text{tr}}|} \sum_{(\mathbf{x}_i, k) \in \mathcal{S}_k^{\text{tr}}} \rho_{\mathbf{x}_i}.$$

Unlike the classical centroid, q_k is usually a *mixed* state and has no direct counterpart in the original feature space.

Distance measure. Similarity between two density patterns is quantified by the *trace distance*

$$d_{\text{tr}}(\rho, \sigma) = \frac{1}{2} \text{Tr}|\rho - \sigma|,$$

a bona-fide metric that reduces to the Euclidean norm in two-dimensional Bloch space.

Quantum-inspired nearest-mean classifier (QNMC). The prediction rule mirrors the classical CNM-C but replaces vectors, means and Euclidean distance by density patterns, quantum centroids and trace distance:

$$\text{QCl}(\mathbf{x}) = \arg \min_k d_{\text{tr}}(\rho_{\mathbf{x}}, q_k).$$

Properties and practical notes.

- **No optimisation:** training is an $O(nd^2)$ centroid accumulation; prediction costs $O(nd^3)$ if eigen-decomposition is used to evaluate the trace distance.
- **Rescaling sensitivity:** unlike the classical NMC, the QNMC is *not* scale-invariant. Feature-wise weights therefore act as tunable hyper-parameters that can boost performance when chosen to reflect feature importance.
- **Encoding choice:** encodings that preserve the original vector norm (e.g. the informative map) often outperform purely directional ones, because the norm carries class information in many datasets.

The next section adapts this framework to an optical joint-transform-correlator implementation and benchmarks it on MNIST.

Table 1: Classification accuracy (mean \pm std.)

Classifier	Distance metric	Encoding	Accuracy (%)
RBF-C	Euclidean	-	-
RBF-C	JTC	-	-
CNM-C	Euclidean	—	80.38 ± 0.38
CNM-C	JTC	—	72.42 ± 0.55
QNM-C	Trace	Standard	85.84 ± 0.38
QNM-C	Fidelity	Standard	78.77 ± 0.34
QNM-C	Trace	Informative	81.26 ± 0.37
QNM-C	Fidelity	Informative	82.39 ± 0.40

iii. Joint Transform Correlator

III. RESULTS

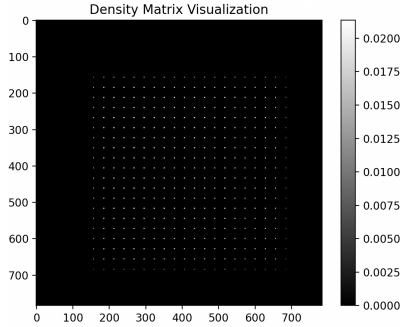
IV. DATA ENCODING

i. Gram Matrix Encoding

We consider a column vector u representing an image sample. To embed this into a higher-order representation, we construct the normalized outer product:

$$\rho = \frac{uu^T}{\text{Tr}(uu^T)}$$

This yields a density-like matrix ρ with trace 1, encoding pairwise correlations between pixels. Such an encoding is inspired by quantum mechanical density operators and preserves structural information in the image.

Figure 3: Gram matrix encoding ρ of the digit 1 image

V. RADIAL BASIS FUNCTION NETWORK

RBF networks model the hypothesis function $h(x)$ using a radial function centered around key points (called *centers*). Each center influences the prediction based on its proximity to the input x . The standard form is:

$$h(x) = \sum_{n=1}^N w_n \exp(-\gamma \|x - x_n\|^2)$$

where γ determines the width of the radial functions, and w_n are the learned weights.

i. Exact Interpolation

Given training data $D = \{(x_1, y_1), \dots, (x_N, y_N)\}$, we can formulate the interpolation condition:

$$\sum_{m=1}^N w_m \exp(-\gamma \|x_n - x_m\|^2) = y_n, \quad \forall n$$

In matrix notation:

$$\Phi \mathbf{w} = \mathbf{y}, \quad \text{where } \Phi_{n,m} = \exp(-\gamma \|x_n - x_m\|^2)$$

If Φ is invertible, we can directly solve:

$$\mathbf{w} = \Phi^{-1} \mathbf{y}$$

This yields an exact interpolating function.

ii. Classification

For classification, we apply a decision rule such as:

$$h(x) = \text{sign} \left(\sum_{n=1}^N w_n \exp(-\gamma \|x - x_n\|^2) \right)$$

In multi-class settings, this can be extended to a softmax output over class scores.

iii. Training with Least Squares

Instead of enforcing exact interpolation, we may use a least squares approach to minimize:

$$E = \sum_{n=1}^N (h(x_n) - y_n)^2$$

This leads to the ridge-regularized solution:

$$\mathbf{w} = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T \mathbf{y}$$

where λ is a regularization parameter.

VI. CHOOSING RBF CENTERS

Using every data point as a center is computationally expensive. Instead, we reduce the number of centers to $K \ll N$ by clustering the dataset using **K-Means**:

We define:

$$J = \sum_{k=1}^K \sum_{x_n \in S_k} \|x_n - \mu_k\|^2$$

and use Lloyd's algorithm to iteratively minimize J :

$$\mu_k = \frac{1}{|S_k|} \sum_{x_n \in S_k} x_n$$

$$S_k = \{x_n : \|x_n - \mu_k\| \leq \|x_n - \mu_j\|, \forall j \neq k\}$$

Repeat these steps until convergence. Since the process is sensitive to initialization, we typically run K-Means multiple times and choose the best clustering.

Figure 4: Example of a Gaussian RBF centered at μ

VII. IMPLEMENTATION NOTES

The RBF network was implemented in Python using NumPy and scikit-learn. To avoid memory issues, we:

- Use MiniBatchKMeans to find centers efficiently.
- Use perceptron-based or regularized least squares learning.
- Avoid computing large $\Phi^T \Phi$ matrices when possible.

VIII. OPTICAL IM

IX. CONCLUSION

RBF networks provide an elegant and powerful framework for classification. Through Gaussian kernel construction and data-driven center selection, they can adapt to local structures in the data. When combined with efficient encodings such as Gram matrices and practical training algorithms, they scale to real-world tasks like digit recognition.

REFERENCES

- [1] E.Z. Cruzeiro, C. De Mol, S. Massar, et al. Quantum-inspired classification based on quantum state discrimination. *Quantum Machine Intelligence*, 6:79, 2024.
- [2] Andrei Drăgulescu. Optical correlators for cryptosystems and image recognition: A review. *Sensors*, 23(2):907, 2023.
- [3] Wesley E. Foor and Mark A. Neifeld. Adaptive, optical, radial basis function neural network for handwritten digit recognition. *Applied Optics*, 34(32):7545–7555, 1995.
- [4] Bahram Javidi and Joseph L. Horner. Single spatial light modulator joint transform correlator. *Applied Optics*, 28(5):1027–1032, 1989.
- [5] M. A. Neifeld and D. Psaltis. Optical implementations of radial basis classifiers. *Applied Optics*, 32:1370–1379, 1993.
- [6] D. Psaltis, E. G. Paek, and S. S. Venkatesh. Optical image correlation with a binary spatial light modulator. *Optical Engineering*, 23:698–704, 1984.
- [7] G. Sergioli, G. Russo, E. Santucci, A. Stefano, S.E. Torrisi, S. Palmucci, C. Vancheri, and R. Giuntini. Quantum-inspired minimum distance classification in biomedical context. *quant-ph*, 2018.
- [8] A. Vander Lugt. Signal detection by complex spatial filtering. *IEEE Transactions on Information Theory*, 10(2):139–145, Apr 1964.