www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 17(3), 2026, 287-301. ISSN: 2007-1558.
https://doi.org/10.61467/2007.1558.2026.v17i3.1457

Spectral analysis of distorted and autoencoder-reconstructed MNIST images in a kernel-induced feature space

Rafael Castaneda-Diaz[1,2], Daniela López-Betancur[1], Carlos Guerrero-Méndez[1], Efrén González-Ramírez[1], Flossi Puma-Ttito[1], V.H. Carrera-Escobedo [2]
[1] Unidad Académica de Ciencia y Tecnología de la Luz y la Materia, Universidad Autónoma de Zacatecas, Campus Siglo XXI, Zacatecas C.P. 98160, Mexico.
[2] Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria de Ingeniería Campus Zacatecas (UPIIZ), Zacatecas 98160, Mexico.
Email address(es): rcastanedad@ipn.mx, danielalopez106@uaz.edu.mx,, guerrero_mendez@uaz.edu.mx, gonzalezefren@uaz.edu.mx, fdayana.puma@uaz.edu.mx, vcarrerae@ipn.mx
✉Corresponding author: Rafael Castaneda-Diaz

Abstract. In this paper, the unsupervised learning spectral methods Principal Component Analysis (PCA) and Kernel Principal Component Analysis (KPCA) were explored. PCA captured the maximum variance of MNIST images, including their distorted and autoencoder-reconstructed versions, and KPCA extracted the maximum variance and nonlinear structures of the feature space where the same images were mapped. Accordingly, for KPCA the kernel functions: a Polynomial of degree 5, a Radial Basis Function (RBF), and a Cosine kernel, were evaluated. A correlation analysis for distortions level (DL), digit inclination (DI) and anisotropic (AN) of pixel values of the same images and the PCA-derived components from KPCA were calculated. DL and AN exhibit strong correlation with the first KPCA component for the polynomial and RBF kernels, indicating substantial explanatory strength. In contrast, correlations involving DI, although statistically significant due to the large sample size, show weak effect sizes and therefore limited explanatory values.

Keywords: Convolutional, autoencoder, geometric factors, PCA, KPCA, RKHS.

Article information
Received: May 30, 2026
Accepted: Jun 2, 2026

1 Introduction

In machine learning problems, extracting meaningful features from a dataset with high dimensionality is an important task. A high-dimensional dataset is a dataset that contains many variables (features) relative to the number of samples. In many cases, one of the main goals is to visualize relationships between them, for instance: biological variables (Gorban et al., 2008), gene expression (Reverter et al., 2014), in digital medical image (Yun et al., 2023), digital histology (Jansen & Niyogi, 2013), face recognition (He et al., 2005), in microarray data (Hira & Gillies, 2015), text classification (Shah & Patel, 2016). If the number of dimensions is extremely high, then different problems associated with it take place, for instance: curse of dimensionality and empty space problem. In this sense, many other problems take place: sparsity due to expansion of the space occupied by the dataset, overfitting due models learn noise, high computational cost, understanding the data and directly visualizing spaces beyond three-dimension. Such problems require efficient automated methods that can reduce the dimensionality of a dataset in an intelligent way. Basically, dimensionality reduction seeks to transform the high-dimensional data into a lower dimensional space that retains certain properties of the input data (Strange & Zwiggelaar, 2014a). In general, dimensionality reduction is separated into two broad approaches. -feature selection and feature extraction. However, both approaches are used to compress the input data. Authors in Ghojogh et al. (2023) considered that dimensionality reduction methods can be grouped into three categories: spectral dimensionality reduction, probabilistic reduction, and (artificial) neural-network dimensionality. In this article, we explored the applications of two unsupervised learning spectral methods: the linear method PCA and the nonlinear method KPCA, which are described in the following sections.

The first step is to embed the data in a suitable feature space, and then use an algorithm based on linear algebra, geometry or statistics, to discover patterns in the embedded data. In this paper, from the input space $\mathbb{R}^{784}$, the linear method PCA directly captures maximum global variance from the image dataset. In contrast, KPCA extracts nonlinear structures from a vector space, called feature space F, into which image dataset is mapped. Additionally, linear relations are sought among the projections of the data items in F (Saul et al., 2006; Scholkopf et al., 1999; Shawe-Taylor & Cristianini, 2004). According to authors in Ghojogh et al. (2023) unsupervised dimensionality reduction methods have one of the following three approaches: 1. Similar points in the input space are embedded close to one another in the embedding space. 2. The dissimilar points in input space are embedded far away from one another in the embedding space. 3. The case where there is the combination of both, the first and the second approaches together. In this sense, the applications of three different kernels in KPCA, nonlinearities of information were revealed visually through two-dimensional plots. Kernel functions induce different RKHS structures; fact that can be appreciated in the two-dimensional plots. Furthermore, a kernel function can be considered as a nonlinear similarity measure between input data (Saul et al., 2006). The potential of KPCA for capturing a global structure but preserving the local structure of data is undoubted, but many questions are raised. Authors in (Castaneda-Diaz et al., 2025) reported that MNIST images of 28x28 pixels were altered with different levels of noise and then reconstructed by a convolutional autoencoder (CAE). In this paper, PCA and KPCA are applied to the materials described above. On the other hand, to find possible correlations between the variables: DL, DI and AN from images and PC1 and PC2 obtained from applying PCA on RKHS, the Pearson correlation coefficient (PCC) r and the Spearman correlation coefficient (SCC) r_s were calculated and compared. Additionally, to support the obtained correlations, several hypotheses were formulated and tested at a 99% confidence level, and the corresponding p-values were calculated.

The outline of this paper is as follows. In section 2, a general setting for the unsupervised spectral reduction methods PCA and KPCA are described. In section 3, the underlying theory for both methods, dataset descriptions and preprocessing are presented. In section 4, normality test and correlation analysis for parameters of the used images with PCA-derived components from RKHS are presented. Finally, section 5 contains concluding arguments and suggestions for future work. **Table** 1 lists the notations used throughout the article.

Table 1. Notations used in this article

$\mathbb{R}^d$	d-dimensional Eculidean space	k	Kernel Function k defined on $\mathbb{R}^d \times \mathbb{R}^d$
$x, \boldsymbol{x}, \boldsymbol{X}$	Scalar, vector, matrix	$k(\boldsymbol{x}, \cdot) \in F$	Elements of F
$Col(\boldsymbol{U})$	Column space of the matrix $\boldsymbol{U}$	ϕ	Nonlinear transformation defined on $\mathbb{R}^d$
F	Feature space	λ_k, v_k	Eigenvalue λ_k, eigenvector v_k
$\boldsymbol{u}_j$	j-the vector $\boldsymbol{u}$	$\boldsymbol{K}(\boldsymbol{i}, \boldsymbol{j})$	Gram matrix $\boldsymbol{K}(\boldsymbol{i}, \boldsymbol{j}) = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$
$\|\cdot\|$	Euclidean distance	$< \boldsymbol{\phi}(\boldsymbol{x}_i), \boldsymbol{\phi}(\boldsymbol{x}_j) >$	Inner product in F

2 State of the art

2.1 General framework

In mathematical words, an image can be described as a data point or as one observation in a d-dimensional vector space, i.e, the Euclidean space $\mathbb{R}^d$. In this context, $\mathbb{R}^d$ is called input space. The d values of one observation are considered as d features. In general, a data set of n points with d features each one, occupies some embedded region of $\mathbb{R}^d$ with local dimensionality less than d. This fact is called the manifold hypothesis (Fefferman et al., 2016; Ghojogh et al., 2023). To have as much information as possible from a dataset, there are two approaches, one for selecting, and one for extracting the most informative features from all data points in the space $\mathbb{R}^d$, respectively. Selecting methods basically take the p most informative features from the d-dimensional vector, where $p \leq d$ and $p \in (0, d]$. Consequently, the new p-dimensional vector in $\mathbb{R}^p$, discards those coordinates that do not provide important information. On the other hand, extracting methods transform the d-dimensional data vector into a p-dimensional data vector, where the new p features are completely different from the original features (Ghojogh et al., 2019). Particularly, feature extraction is referred in different ways, including: dimensionality reduction, manifold learning, subspace learning, submanifold learning, manifold unfolding, embedding encoding, and representation learning (Bengio et al., 2013; Cayton, 2008; LeCun et al., 2015). In this paper, we are particularly interested in the application of extracting methods which can be used to reduce high dimensionality, i.e., to produce a compact low dimensional encoding and to generate a two-dimensional data visualization (Ghojogh et al., 2023). A visualization refers to techniques that can represent a dataset in a manner that reveals some underlying structure of the data in a way that is easily understood or appreciated by the user (Shawe-Taylor & Cristianini, 2004) .

According to Ghojogh (2021), dimensionality reduction methods can be grouped into categories: spectral dimensionality reduction, probabilistic dimensionality reduction and (artificial) neural network-based dimensionality reduction. In this case, we focused on the application of spectral dimensionality reduction methods. It is remarkable that spectral methods provide a geometric perspective, and they are useful for finding linear or nonlinear submanifolds of the data, and are often reduced to a generalized eigenvalue problem (Saul et al., 2006; Strange & Zwiggelaar, 2014b).

Historically, the first unsupervised spectral linear method was the PCA proposed by Pearson 1901 (Jolliffe, 2014). PCA seeks to find the low-dimensional subspace within the data that maximally preserves the covariances up to rotation. This maximum covariance subspace encapsulates the direction along which the data varies the most (Strange & Zwiggelaar, 2014b). A similar supervised method is called Fisher Discriminant Analysis (FDA) proposed by Fisher in 1919 (Fisher, 1919). This supervised linear method, like PCA, is based on scatter, i.e., the dispersion of the data. FDA is used to find the proper subspace which preserves either the relative similarity or relative dissimilarity of the points after transforming the data from the input space onto a subspace. In a similar manner, a method called Multidimensional Scaling (MDS) tries to preserve similarities after transforming linearly the input data and also seeking to preserve pairwise Euclidean distances (Ghojogh et al., 2023). The linear methods operate under the assumption that data lies on or near a linear subspace, however, spaces globally may be highly nonlinear. Under these circumstances, it is possible to lead to misunderstanding about geometric structures of the data. On the other hand, whereas linear dimensionality reduction methods measure inter-point Euclidean distances, the nonlinear methods measure the inter-point manifold distances by approximating geodesics. In this sense, the first nonlinear method is called Sammon map, which implements an improved version of the metric in MDS (Strange & Zwiggelaar, 2014b). Recently, the nonlinear methods called Stochastic Neighbor Embedding (t-SNE) (Maaten & Hinton, 2008) and Uniform Manifold Approximation and Projection (UMAP) are getting popularity. In both cases, these methods are competitive for generating low-dimensional visualization and preserving more of the local and the global structure (McInnes et al., 2020; Mittal et al., 2024). In view of Ghojogh et al. (2023), linear algorithms cannot perform well on nonlinear data. Consequently, there are two approaches that can be used: the first is to design a nonlinear algorithm to manage nonlinear data; and the second is that nonlinear data should be transformed into linear. Then, the transformed data, which now has a linear pattern, will be able to use a linear algorithm. This approach is known as kernelization. From this point of view, the nonlinear method called KPCA has the capacity for extracting complicated nonlinear structures from a dataset (Bishop & Bishop, 2024), (Wang, 2014). Basically, KPCA implies the application of PCA on the feature space which is a nonlinear transformation of the input dataset. Additionally, PCA and KPCA can be extremely sensitive to outlying observations and conclusions drawn based on contaminated principal components can be misleading (Deng et al., 2007). In the following section, PCA and KPCA are described. We adopted most of the mathematical notations provided by Ghojogh et al. (2023) in Part II (Spectral Dimensionality Reduction), specifically from Chapters 5 and 6. For additional details not discussed in this paper, readers are referred to this excellent book.

3 Material and methods

3.1 Principal Component Analysis

In PCA, only n data points $\{\boldsymbol{x}\}_{i=1}^{n}$ are required, where $\boldsymbol{x}_i = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$. Therefore, it is necessary to stack in a column-wise matrix data $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}$. The core idea behind PCA is to construct a linear transformation $T: \mathbb{R}^d \to \mathbb{R}^p$. The n vectors $\boldsymbol{x}_i \in \mathbb{R}^d$ under the function T are orthogonally projected onto the vector space spanned by p vectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_p\}$, where each vector is a d-dimensional and usually $p << d$. The p vectors are stacked column-wise in a matrix $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_p] \in \mathbb{R}^{d \times p}$. In this sense, the aim is to project onto the column space of $\boldsymbol{U}$, denoted by $Col(\boldsymbol{U})$. $Col(\boldsymbol{U})$ is called PCA *subspace*. Based on the matrix $\boldsymbol{U}$, if the p vectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_p\}$ are orthonormal, then $\boldsymbol{U}$ is orthogonal and $\boldsymbol{U}^T = \boldsymbol{U}^{-1}$ and $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the $d \times d$ identity matrix. Furthermore, it is possible to define the $d \times d$ projection matrix $\boldsymbol{U}$, which projects forward $\boldsymbol{x}_i \in \mathbb{R}^d$ as $\boldsymbol{U}^T\boldsymbol{x}_i$ onto $Col(U) \subset \mathbb{R}^p$ and backwards $\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{x}_i$ to $\mathbb{R}^d$. If $\boldsymbol{U}^T\boldsymbol{x}_i$ is denoted as $\widetilde{\boldsymbol{x}_i} = \boldsymbol{U}^T\boldsymbol{x}_i$ and $\widehat{\boldsymbol{x}_i} = \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{x}_i$ as the reconstruction back of $\boldsymbol{x}_i$ onto $\mathbb{R}^d$, then $\widehat{\boldsymbol{x}_i} \approx \boldsymbol{x}_i$. In this sense, if $\boldsymbol{x}_t \in \mathbb{R}^d$ is an out-of-sample data point, its linear transformation can be calculated as $\widetilde{\boldsymbol{x}_t} = \boldsymbol{U}^T\boldsymbol{x}_t$ onto $Col(\boldsymbol{U}) \subset \mathbb{R}^p$, and similarly, it is reconstructed back as $\widehat{\boldsymbol{x}_t} = \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{x}_t$ onto $\mathbb{R}^d$. PCA is a multivariate statistical technique that tries to find vectors $\boldsymbol{u}$ which direction represent the maximum variance of the dataset in $\mathbb{R}^d$ space. According to the previous description, $\boldsymbol{u}^T\boldsymbol{X}$ is the projection of the matrix data $\boldsymbol{X}$ onto PCA subspace in the direction of $\boldsymbol{u}$.

Initially, PCA requires the data to be centered at the origin $\boldsymbol{0}$ in $\mathbb{R}^d$. To be centered, every data point $\boldsymbol{x}_i$ in $\boldsymbol{X}$ takes the form $\breve{\boldsymbol{x}_i} = \boldsymbol{x}_i - \mu$, where $\mu = (\mu_1, \mu_2, \dots, \mu_d)$ and $\mu_i = \frac{1}{n}\sum_{k=1}^{n} x_{ki}$ for $x_{ki} \in (x_{1i}, x_{2i}, \dots, x_{ni})^T$, the i-th feature and $i = 1, \dots, d$. Therefore, the projection of $\breve{\boldsymbol{x}_i}$ onto PCA subspace is $\widetilde{\boldsymbol{x}_i} = \boldsymbol{U}^T\breve{\boldsymbol{x}_i}$ and its reconstruction back $\widehat{\boldsymbol{x}_i} = \boldsymbol{U}\boldsymbol{U}^T\breve{\boldsymbol{x}_i} + \mu = \boldsymbol{U}\widetilde{\boldsymbol{x}_i} + \mu$, if μ is considered.

In a similar manner, for an out-of-sample data point, $\hat{x}_t = UU^T\breve{x}_t + \mu = U\widetilde{x_t} + \mu$. If the centered data are stacked in a column-wise matrix $\breve{X} = [\breve{x}_1, \breve{x}_2, \dots, \breve{x}_n] \in \mathbb{R}^{d\times n}$, then the reconstruction back of the centered data matrix $\breve{X}$ is $\hat{X} = UU^T\breve{X}$. In the general case, for the n data points, the correct U minimizes the residual matrix $\breve{X} - \hat{X}$ in one or more directions $u_1, u_2, \dots, u_p$. In any case, each vector u_j for $j = 1, \dots, p$, satisfies $u_j^T u_j = \| u_j \|_2^2 = 1$ and $\breve{x}_i^T u_j = u_j^T \breve{x}_i \in \mathbb{R}$. Notice that $\breve{x}_i\breve{x}_i^T$ is the covariance matrix of the features in $\breve{x}_i$. Consequently, the sum of all dot products $\breve{x}_i\breve{x}_i^T$ is equal to a $d \times d$ *covariance matrix* of the centered dataset, denoted by S, which $S = \sum_{i=1}^{n} \breve{x}_i\breve{x}_i^T = \breve{X}\breve{X}^T$.

To find the matrix U, we can find the eigen-decomposition of S, i.e., $SU = U\Lambda$. The eigen-decomposition implies to find $SU = U\Lambda$, where columns of U, i.e., $\{u_1, u_2, \dots, u_p\}$ are the eigenvectors of S and Λ with the eigenvalues of S, respectively. Optionally, one can use the matrix factorization technique known as Singular Value Decomposition (SVD) of $\breve{X}$. The SVD of $\breve{X}$ takes the form $\breve{X} = U\Sigma V^T$ where the columns of $U \in \mathbb{R}^{d\times p}$ (called the *left singular vectors*) are the eigenvectors of $\breve{X}\breve{X}^T$ and the columns of $V \in \mathbb{R}^{n\times n}$ (called the *right singular vectors*) are the eigenvectors of $\breve{X}^T\breve{X}$, and $\Sigma \in \mathbb{R}^{d\times n}$ is a rectangular diagonal matrix whose diagonal entries (called *singular values*) are the squared root of eigenvalues of $\breve{X}\breve{X}^T$ and/or $\breve{X}^T\breve{X}$. In this path, the *left singular vectors,* columns of U are consider the principal directions in the PCA subspace.

3.2 Kernel Principal Component Analysis

KPCA was proposed as a nonlinear form of PCA. KPCA is based on the use of integral operators defined by kernel functions, which allow the efficient computation of principal components in a high-dimensional feature space related to the input space by a nonlinear map. First, suppose a transformation $\phi: \mathbb{R}^d \to F$ is defined, where F denotes the feature space, which is t-dimensional. t may be infinite or finite. Despite the high-dimensionality of F, for certain choices of ϕ, it is possible that PCA performs in F (Schölkopf et al., 1998).

Consider n data points $\{x_i\}_{i=1}^n$, where $x_i = (x_{i1}, x_{i2}, \dots, x_{id}) \in \mathbb{R}^d$. Suppose that $\{\phi(x_1), \phi(x_2), \dots, \phi(x_n)\}$ is the mapped data in the feature space F, and it is centered

$$\frac{1}{n}\sum_{i=1}^{n} \phi(x_i) = 0. \quad \textbf{(1)}$$

As it was described for PCA, we must find t eigenvalues $\lambda \geq 0$ and their corresponding eigenvectors v of the $t \times t$ covariance matrix of the mapped data into the feature space given by

$$C = \frac{1}{n}\sum_{i=1}^{n} \phi(x_i) \quad \textbf{(2)}$$

which satisfies $\lambda_k v_k = Cv_k$ for $k = 1,2, \dots, t$. Notice that v_k is a linear combination of images $\phi(x_i)$ in F i.e., $v_k = \sum_{i=1}^{n} \alpha_{ki}\phi(x_i)$. Consequently, combining $\lambda_k v_k = Cv_k$ with Eq. (2), then

$$\frac{1}{n}\sum_{i=1}^{n} \phi(x_i) \cdot \phi(x_i)^T \sum_{j=1}^{n} \alpha_{kj}\phi(x_j) = \lambda_k \sum_{i=1}^{n} \alpha_{ki}\phi(x_i). \quad \textbf{(3)}$$

If Eq. (3) is multiplied by $\phi(x_l)^T$ in both sides

$$\frac{1}{n}\sum_{i,l=1}^{n} \phi(x_l)^T \cdot \phi(x_i) \sum_{i,j=1}^{n} \alpha_{kj}\phi(x_i)^T \cdot \phi(x_j) = \lambda_k \sum_{i,l=1}^{n} \alpha_{ki}\phi(x_l)^T \cdot \phi(x_i), \quad \textbf{(4)}$$

then, we have

$$\frac{1}{n}\sum_{i,l=1}^{n} \phi(x_l)^T \cdot \phi(x_i) \sum_{i,j=1}^{n} \alpha_{kj}\phi(x_i)^T \cdot \phi(x_j) = \lambda_k \sum_{i,l=1}^{n} \alpha_{ki}\phi(x_l)^T \cdot \phi(x_i). \quad \textbf{(5)}$$

According to Wang (2014), if a kernel function is defined as $k(x_i, x_j) = < \phi(x_i)^T, \phi(x_j) >$, then Eq. (5) can be rewritten in the matrix form $\frac{1}{n}K^2 v_k = \lambda_k K v_k$, where $K(i,j) = k(x_i, x_j)$ and v_k is a n-dimensional column vector $v_k = [\alpha_{k1}, \alpha_{k2}, \dots, \alpha_{kn}]^T$. K is known as Gram matrix. The matrix K is positive semidefinite, it is often nonsingular and for nonzero eigenvalues, it is possible to multiply as follows $K^{-1}(\frac{1}{n}K^2 v_k) = K^{-1}(\lambda_k K v_k)$, such that $Kv_k = n\lambda_k v_k$. The kernel function is defined as $k: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, which is either continuous or has a countable domain, and can be decomposed $k(x, y) = < \phi(x), \phi(y) >$, where ϕ is the map into a feature Hilbert space F (Shawe-Taylor & Cristianini, 2004a). In this context, it is

important to understand that F is a space of functions f. As indicated by Scholkopf et al. (1998), for principal component extraction one computes projections of the image of a test point $\phi(\boldsymbol{x})$ onto eigenvectors v_k in F according to

$$< v_k, \phi(\boldsymbol{x}) >= \sum_{i=1}^{n} \alpha_{ki} < \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x})^T > . \quad (6)$$

However, $\phi(\boldsymbol{x})$ in Eq. (6) requires explicit form in dot products, but most of the time we do not have the explicit form of the functions ϕ in F. Therefore, the kernel function k basically plays a fundamental role: it defines the inner product without explicitly computing $\phi(\boldsymbol{x})$ in F. In general, for all functions $f \in F$, f is defined on $\mathbb{R}^d$, $f \longmapsto f(\boldsymbol{x})$ $(\boldsymbol{x} \in \mathbb{R}^d)$ are continuous and $f(\boldsymbol{x}) = < f, k(\boldsymbol{x}, \cdot) >$ where $k(\boldsymbol{x}, \cdot) \in F$. For this property, k is called a reproducing kernel. In this sense, the set of functions $\{k(\boldsymbol{x}, \cdot): \boldsymbol{x} \in \mathbb{R}^d\}$ span a subset of functions f is referred to as Reproducing Kernel Hilbert Space (RKHS). See (Scholkopf et al., 1999) for details. For this work, we implemented three well-known kernels to compare their capacities:

1. Radial Basis Function or Gaussian Kernel:

$$k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma \parallel \boldsymbol{x} - \boldsymbol{y} \parallel_2^2) , \quad (7)$$

where $\gamma = \frac{1}{\sigma^2}$ and σ is the variance of the kernel. A proper value suggested for $\gamma = 1/d$ where d is the dimensionality of input data. In this case, $\gamma = \frac{1}{784} = 0.00123$.

2. Polynomial Kernel:

$$k(\boldsymbol{x}, \boldsymbol{y}) = (\gamma \boldsymbol{x}^T \boldsymbol{y} + c)^d, \quad (8)$$

where $\gamma = 00.00123$, $c = 0$ and $d = 5$.

3. Cosine Kernel:

$$k(\boldsymbol{x}, \boldsymbol{y}) = \cos(\boldsymbol{x}, \boldsymbol{y}) = \frac{x^T y}{\|x\|_2 \|y\|_2}, \quad (9)$$

Under these conditions, PCA can be applied to the resulting RKHS subspaces induced by a function k.

3.3 Dataset description and preprocessing

In this article, MNIST images are used. MNIST dataset —MNIST for Modified National Institute of Standard, and it is available on the website keras.io. This dataset contains 70000 handwritten digit grayscale images with pixels values from 0 (completely white) to 1 (completely black), each with size 28×28 pixels, that is, 784 pixels. On this matter, MNIST is a common standard dataset that has been implemented to show the capacities, particularly for generating low-dimensional visualization with linear and nonlinear methods (Maaten & Hinton, 2008; McInnes et al., 2020; Bishop & Bishop, 2024) . In this context, each pixel value was considered as a feature, i.e., 784 features per image. In this sense, an image is an object contained within a 784-dimensional space. The experimental work described in the following sections was based on the distorted MNIST images implemented in (Castaneda-Diaz et al., 2025). Authors reported the implementation of a CAE for restoring distorted MNIST images. Distortions within images were induced randomly by corrupting separately the whole dataset in three different proportions of the 784 pixels: 10% (79 pixels), 20% (157 pixels), and 30% (235 pixels), respectively. The distortion process involved replacing the original pixels with completely black pixels, i.e., to set the value to 1. To this point, authors reported that the CAE model was cloned three times to be trained with the three corresponding distorted training datasets. The three resulting models were tested by cleaning the corresponding distorted testing images. The cleaning process generated three new data sets of previously unseen images. Consequently, the 10000 original MNIST testing images, and the corresponding distorted 30000 versions and cleaned 30000 versions were used as out-of-sample data points to project onto PCA subspace. Some of them are depicted in **Fig.** 1.

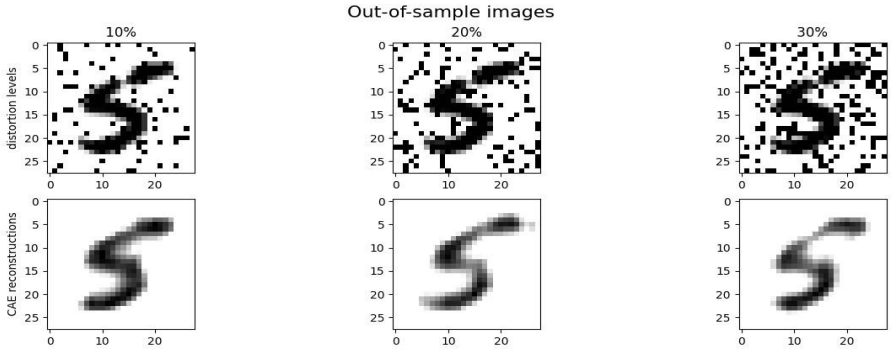

Fig. 1. Random distorted images and their CAE-reconstructed versions.

3.4 Two-dimensional visualization of PCA and KPCA projections

As a part of the preprocessing stage, a sample of 7000 random images was used to generate PCA and KPCA two-dimensional projections. As previously described, PCA has the capacity for reducing dimensionality while capturing the maximum global variance from a dataset in one principal direction $\boldsymbol{u}_k$ from the basis vectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_p\}$. **Table** 2 describes the composition of the sampled images used to generate the p vectors. **Fig. 2**. in (a) shows the PCA projections from input space $\mathbb{R}^{784}$ onto $\mathbb{R}^p$ for $p = 2$. According to authors in Castaneda-Diaz et al. (2026), the same information was used to compare the functionality of PCA and FDA methods. Particularly, in this reference, PCA and FDA spaces were generated with exclusively clean original training MNIST images, whereas their respective distorted versions and their autoencoder-reconstructed versions were used as out-of-samples. However, the reported projections for PCA are very similar to these in plot (a) in **Fig. 2,** which shows that the distorted images and their autoencoder-cleaned versions fall projected into the same region but following the direction of the first two principal components. Notably, no clear separation is observed, as most samples overlap. On the other hand, as previously described for KPCA implementation, one selects a reproducing kernel function k which generates the respective RKHS subspace of F, i.e., $\mathbb{R}^{7000}$. It is remarkable that each RKHS is a subset of the corresponding feature space F that is structured by k and where PCA is applied. **Fig. 2** (b) shows that the cosine kernel structures F in a radial form, whereas (c) and (d) show similar projections generated by the polynomial kernel of degree 5 and RBF kernel, respectively.

Table 2. A combined 7000 sample size was selected using random stratified sampling from MNIST dataset, their corresponding distorted versions in the 10%, 20% and 30% of their pixels and the corresponding CAE-reconstructed versions

MNIST	10% dist.	CAE-recon.	20% dist.	CAE-recon.	30% dist.	CAE-recon.	Total
1000	1000	1000	1000	1000	1000	1000	7000

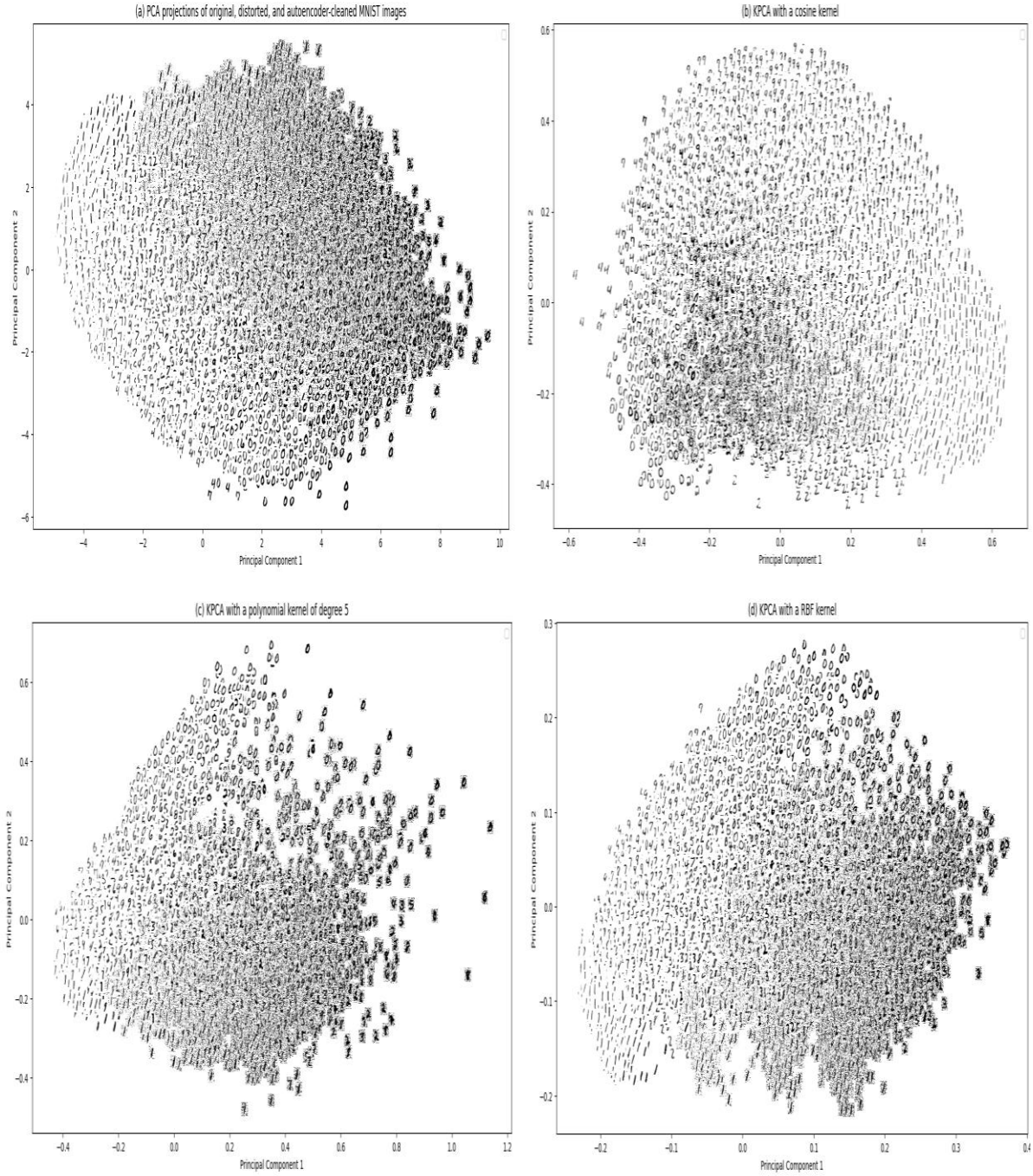

Fig. 2. First row, left to right: (a) shows PCA projections directly from $\mathbb{R}^{784}$; (b) shows KPCA projections with a cosine kernel. Second row, left to right: (c) shows KPCA projections with a polynomial kernel of degree 5 and (d) shows KPCA projections with an RBF kernel. In (b), (c) and (d) PCA was applied to the subspace RKHS in F, i.e., on the corresponding $\mathbb{R}^{7000}$.

Each two-dimensional plot in **Fig**. 2, shows the way the kernel function gives structure to the feature space F and visually reveals clustering structure of the data. The described preprocessing stage was developed using sklearn.decomposition.KernelPCA methos contained in the library from the website https://scikit-learn.org/ . The executed code used a random seed equal to 42. The referenced website suggests that $\gamma = \frac{1}{d}$ (d is the number of features in data) for Eq. (7) and (8). Intuitively, and according to subjective criteria based on the two-dimensional plots shown in **Fig.** 2, in this work we hypothesize that DL, DI and AN in

images appear to be determinant factors in the PCA on RKHS. In this sense, we explore potential correlations between PCA-derived principal components applied on RKHS spaces and those parameters of images. Furthermore, the strength of these associations depends on the kernel used. Basically, the nonlinear kernels capture nonlinear properties, which are apparently visually revealed by the two-dimensional projections.

3.5 Description of variables

So far, we have implemented PCA to obtain two-dimensional projections, one directly from input space $\mathbb{R}^{784}$ and the other three from a 7000-dimnsional feature space induced by the three kernel functions k, see **Fig. 2**. In the following sections, distortion level, digit inclination, and anisotropy are defined.

3.6 Distortion level

With respect to the distortion levels (DL), recall that the noise was induced randomly by corrupting separately all the images of the whole dataset in three different proportions of the 784 pixels: 10% (79 pixels), 20% (157 pixels), and 30% (235 pixels), respectively. In this sense, each image was associated with the numbers 0.0, 0.1, 0.2 or 0.3 according to its distortion level.

3.7 Digit inclination

The stroke direction (DI) for each digit image apparently defines the variance of the projections in **Fig. 2**. This conjecture is based on visual observations. It appears that DI is playing a critical role; however, it could be a misinterpretation. To test this conjecture, a possible approach is to calculate the DI for each image and then evaluate its correlation with the values of the first or second principal components generated by PCA on RKHS. Furthermore, seeing that an image can be represented as a dataset of points with nonzero pixel value scattered on the xy-plane, then PCA can be used for finding the digit inclination. In this regard, we denoted with $PCAI_i$ where i refers to the i-th image. Basically, $PCAI_i$ finds the direction of maximum variances of the nonzero pixel values for i-th image, which is pointed by the eigenvalue associated to the principal eigenvector.

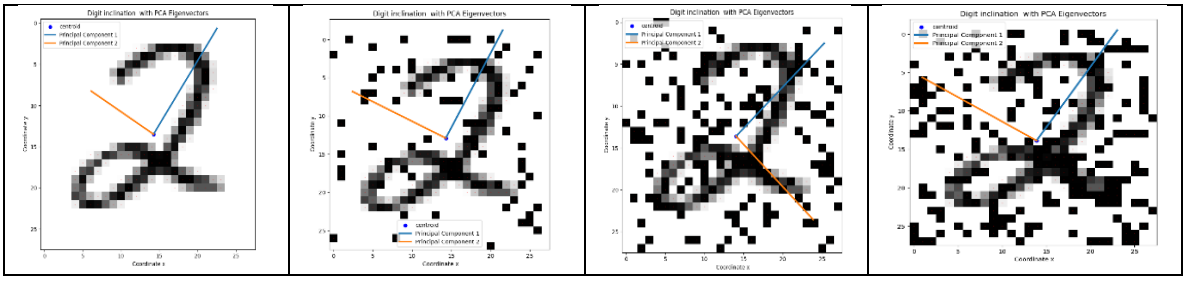

Fig. 3. Digit 2 with various levels of distortion and the indicated stroke direction (blue eigenvector).

Fig. 3. shows four digits 2 whit different levels of distortion. In the plots, the blue line indicates the direction of the stroke, which follows the first principal component direction of $PCAI_i$. To calculate $PCAI_i$ we adopted some ideas from (Burger & Burge, 2022) and (Gonzalez & Woods, 2018). In this sense, we follow the next algorithm of five steps:

1. Identify the coordinates (x_j, y_j) for nonzero pixel value.
2. Compute the centroid: $\overline{\boldsymbol{x}} = \frac{1}{n}\sum_{j=1}^{n} x_j$ and $\overline{\boldsymbol{y}} = \frac{1}{n}\sum_{j=1}^{n} y_j$.
3. Computing the covariance matrix associated to the i-th image:

$$\boldsymbol{C}_i = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{pmatrix}, \tag{10}$$

where $\sigma_{xx} = \frac{1}{n}\sum_{j=1}^{n}(x_j - \overline{\boldsymbol{x}})^2$, $\sigma_{yy} = \frac{1}{n}\sum_{j=1}^{n}(y_j - \overline{\boldsymbol{y}})^2$ and $\sigma_{xy} = \frac{1}{n}\sum_{j=1}^{n}(y_j - \overline{\boldsymbol{y}})^2\left(x_j - \overline{\boldsymbol{x}}\right)^2$.

4. Compute eigenvalues λ_1, λ_2 and the corresponding eigenvectors $v_1 = [v_{1x}, v_{1y}]$ and $v_2 = [v_{2x}, v_{2y}]$ associated to $\boldsymbol{C}_i$. If λ_1 is the eigenvalue associated to v_1 and $\lambda_1 \geq \lambda_2$, then v_1 is the principal component for the i-the image.
5. Consequently, the digit orientation can be computed with the inclination angle $\theta = \tan^{-1}\left(\frac{v_{1y}}{v_{1x}}\right)$ of the principal component $v_1 = [v_{1x}, v_{1y}]$. In this case, the angle is measured in degrees.

3.8 The concept of anisotropy in MNIST images

In general, the concept of anisotropy (AN) has some different connotations. For example, in digital image processing, it is studied as a type of diffusion filter that smooths an image as a physical diffusion process in certain direction. In this context, we adopted it to the situation where variability of the nonzero pixel values are distributed markedly in a certain direction (Burger & Burge, 2022). Alternatively, authors in (Vliet & Verbeek, 1995) defined anisotropy as $1 - \frac{\lambda_2}{\lambda_1}$ to estimate local orientation and anisotropy in noisy images. Basically, in this experimental work, we considered that it is affordable to find correlations between PC1 and PC2 scores with the corresponding measured anisotropy in image. In **Fig. 3.**, DI varied according to the digit stroke that is indicated by the eigenvector (blue vector) corresponding to the larger eigenvalue calculated by $PCAI_i$, so AN is defined in terms of the eigenvalues as follows $\Delta = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}$. Therefore, Δ takes values between 0 and 1. If $\lambda_1 \approx \lambda_2$, then $\Delta \approx 0$ and it reveals an isotropic distribution of nonzero pixel values. In contrast, if $\lambda_1 \gg \lambda_2$, then $\Delta \to 1$ and it reveals an anisotropic distribution of nonzero pixels values along v_1 direction, i.e., data is elongated along the principal component in $PCAI_i$.

4 Results

4.1 Assessment of normality

So far, we have suggested examining possible correlations between DL, DI and AN and the PC1 and the PC2 scores obtained from PCA applied on RKHS space induced by three different kernel functions, respectively. Therefore, it is necessary to calculate the values for DL, AN and DI for each image after projecting it onto RKHS and then perform a normality test to the corresponding values. In this case, a histogram and a Quantile-Quantile plot (Q-Q) are used as a visual normality test, either before using Pearson correlation coefficient (PCC) or Spearman correlation coefficient (SCC) to identify correlations. In this sense, **Fig.** 4. shows the histograms (left) and the corresponding Q-Q plot (right) for DL, AN and DI. In (a) plots indicate that approximately 57.1% of images exhibit 0% of distortion. In contrast, the remaining images are 10%, 20% and 30% distorted. In general, distortion values do not have a normality distribution. On the other hand, for AN and DI, (b) and (c) show positive skewness and long right tails. This observation is supported by Q-Q plots, where the data deviates from the reference line, particularly in the tails, indicating departures from normality. Therefore, AN and DI cannot be assumed to follow a normal distribution.

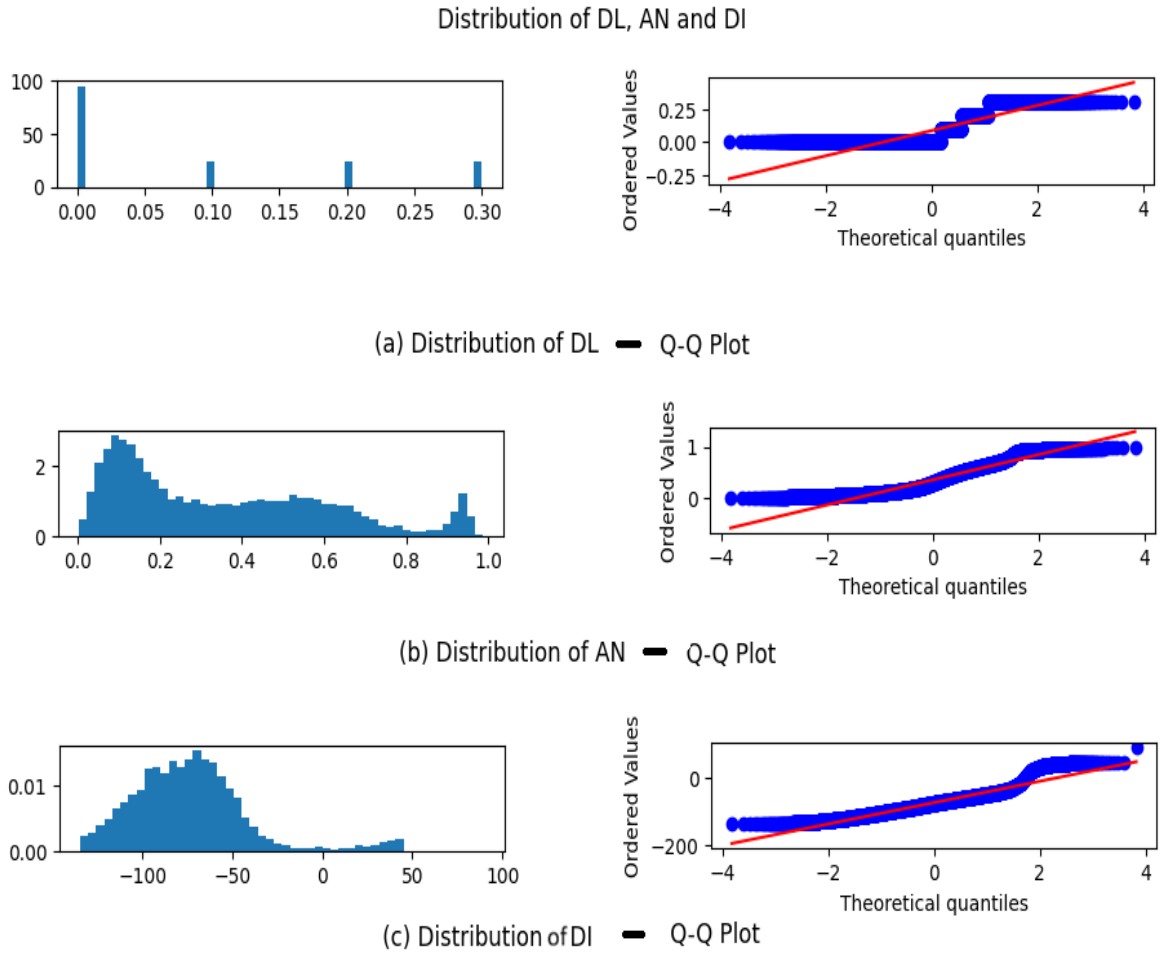

Fig. 4. Visual normality test with a histogram (left) and a Q-Q plot (right): (a) distortion level (DL); (b) anisotropy (AN) and (c) digit inclination (DI).

Similarly, **Fig. 5**. shows the visual normality test for the PC1 and PC2 scores according to the corresponding RKHS space induced by the kernel functions. The histograms and the corresponding Q-Q plots for PC1 and PC2 in (a) and (b) induced by a degree 5 polynomial kernel. In this case, PC1 shows a right-skewed distribution, while PC2 shows a more symmetric bell-shaped distribution. The middle row plots show similar bell-shaped distribution for both PC1 and PC2 induced by the cosine kernel. Finally, the plots in the bottom row show similar distributions induced by the RBF kernel to that induced by the degree-5 polynomial kernel.

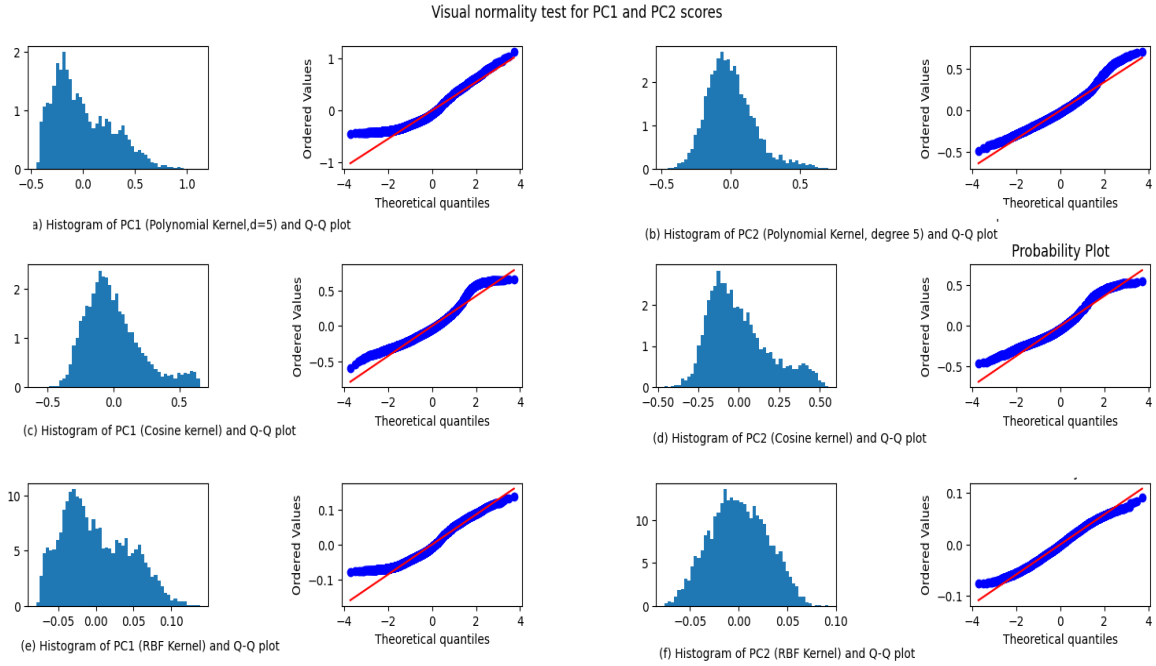

Fig. 5. Visual normality test using histogram (left) and a Q-Q plot (right): Plots in (a) and (b) in the top row show the histograms and the corresponding Q-Q plots for PC1 and PC2 scores, induced by a degree 5 polynomial kernel. Plots (c) and (d), in the middle row, show the histograms and the corresponding Q-Q plots for PC1 and PC2 scores, induced by the cosine kernel. Finally, plots (e) and (f) in the bottom row show the histograms and the corresponding Q-Q plots for PC1 and PC2 scores, induced by the RBF kernel.

4.2 Correlations coefficients PCC and SCC

In the previous section, normality distribution was visually tested, and the results suggested that PC2 scores exhibit consistent with normality for the three kernels. In contrast, PC1 scores exhibit non-normal distribution, noticeable for polynomial and RBF kernels. Even under non-normality conditions, it is possible to identify correlations between DL, AN, and DI and the KPCA scores. So, to measure relationships, the first concept used is the Pearson correlation coefficient (PCC), which measures linear correlations. Additionally, in contrast with PCC, a second concept is the Spearman correlations coefficient (SCC), which measures monotonic relationships. In this case, PCC and SCC are defined in the following paragraphs.
PCC, denoted with r, is defined as follows:

$$r = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}, \tag{11}$$

where $\bar{x}$ and $\bar{y}$ are the means of the variables x and y, and n is the number of components for x and y . r is a measure of the linear relationship between x and y, and it ranges from -1 to 1. The PCC is interpreted according to the following conditions:

- If $r = 1$, there is a perfect linear correlation. As one variable increases, the other increases proportionally.
- If $0 < r < 1$, there is a positive correlation. If r is higher, it indicates a higher linear correlation.
- If $r = 0$, there is no linear correlation.
- If $-1 < r < 0$, there is a negative linear correlation. As one variable increases, the other decreases proportionally.
- If $r = -1$, there is a perfect negative linear correlation.

Additionally, to support the results regarding possible significant correlations revealed by Eq. (11), a null hypothesis H_0 was tested, which states that no relationship exists between two normal distributed variables, i.e., the population correlation coefficient $\rho = 0$. Fundamentally, the interpretation for ρ is like that for r. The parameter ρ is defined as follows:

$$\rho = E\left[\left(\frac{x-\mu_1}{\sigma_x}\right)\left(\right.\right.$$

where ρ is the product-moment coefficient correlation between the random standardized variables $\left(\frac{x-\mu_1}{\sigma_x}\right)$ and $\left(\frac{y-\mu_2}{\sigma_y}\right)$. Furthermore, recall that the sample correlation coefficient r is an unbiased estimator for population parameter ρ. In this sense, a confidence interval for ρ was calculated. So, it is necessary to use the Fisher $z-$transformation that converts r into a variable that is approximately normally distributed. The Fisher-transform correlation is defined as follows:

$$z = \frac{1}{2}\ln$$

which has a mean $\mu_z = \frac{1}{2}\ln\left(\frac{1+\rho}{1-\rho}\right)$ and variance $\frac{1}{n-3}$. Furthermore, the inference about z can be based on the statistic

$$z =$$

$$\frac{\sqrt{n-3}}{2}\, ln\frac{(1+}{(1-}$$

which is a random variable that is approximately the standard normal distribution. Under the assumption of no correlation, i.e., the null hypothesis that $\rho = 0$, the variable in Eq. (14) takes the from

$$z = \frac{\sqrt{n-3}}{2}\, ln\frac{(1+r)}{(1-r)}.$$

In this regard, to calculate the confidence interval for ρ, it is necessary to calculate the confidence interval for μ_z, the mean distribution of z,

$$z - \frac{\frac{z\alpha}{2}}{\sqrt{n-3}} < \mu_z < z$$

and then write the confidence interval for ρ recovering r with the inverse transformation $r = \frac{e^z - e^{-z}}{e^z + e^{-z}}$. Additionally, once the confidence interval for ρ was calculated, it was complemented with the p-value, which is the probability of obtaining results at least as extreme as the observed data, assuming that the null hypothesis is true. **Table** 3 presents the results of calculating the corresponding correlation values for r with DL, AN and DI with PC1 and PC2 scores generated by PCA on RKHS subspaces, the 99% confidence interval for ρ, and the p-value with a significance level of 0.01 for testing H_0 (Myers et al., 2012).

Table 3. Pearson correlation coefficients (PCCs) between DL, AN, and DI and the PC1 and PC2 scores, including p-values, null-hypothesis (H_0) test results, and confidence intervals

	PC1				PC2			
	r	p-value	H_0	CI	r	p-value	H_0	CI
Polynomial with $d = 5$								
DL	**0.834**	0.0	Yes	(0.825, 0.844)	**-0.453**	1.98e-138	Yes	(-0.477, -0.4282)
DI	0.125	4.5e-31	Yes	(0.106, 0.167)	0.319	1.06e-10	Yes	(0.291, 0.3463)
AN	**-0.766**	0.0	Yes	(-0.778, -0.752)	-0.010	0.3605	No	(-0.030, 0.024)
Cosine								
DL	-0.240	1.64e-98	Yes	(-0.313, -0.256)	**-0.355**	0.0	Yes	(-0.381, -0.327)
DI	-0.139	1.34e-37	Yes	(-0.144, -0.083)	-0.148	7.38e-36	Yes	(-0.201, -0.141)
AN	**0.731**	0.0	Yes	(0.703, 0.733)	0.194	2.574e-79	Yes	(0.164, 0.223)
Radio Basis Functions								
DL	**0.840**	0.0	Yes	(0.831, 0.849)	**-0.342**	1.165e-191	Yes	(-0.448, -0.398)
DI	0.160	1.15e-37	Yes	(0.129, 0.189)	0.075	7.54e-06	Yes	(0.044, 0.105)

AN	**-0.812**	0.0	Yes	(-0.815, -0.793)	-0.064	6.820e-19	Yes	(-0.094,-0.033)

On the other hand, SCC is denoted by r_s and it is a non-parametric measure of relation between variables. r_s uses ranks to calculate correlation. It is defined as follows:

$$r_s = 1 - \frac{6\sum d_i}{n(n^2-1)}, \tag{17}$$

where $d_i = rank(x_i) - rank(y_i)$ is the difference between the i −ranks of x_i and y_i, and n is the number of paired observations (samples). Likewise, r_s is interpreted in the same manner as r (Croatia et al., 2015). According to (de Winter et al., 2016), r is more suitable for light-tailed distributions, whereas r_s is preferable when variables feature heavy-tailed distributions or when outliers are present, as is often the case in psychological research. The application of SCC is described in **Table** 4.

Table 4. SCC calculated for DL, AN and DI with PC1 and PC2 scores

Kernel	Polynomial degree 5		Cosine		Radial Basis Functions	
	PC1	PC2	PC1	PC2	PC1	PC2
	r_s	r_s	r_s	r_s	r_s	r_s
DL	**0.812**	**-0.436**	-0.211	**-0.337**	**0.842**	**-0.408**
DI	0.070	-0.003	-0.080	-0.204	0.166	0.020
AN	**-0.820**	0.130	**0.552**	0.370	**-0.846**	0.021

Table 4 contains the values of r_s that were calculated. Basically, they are very similar to the values of r. Notice that for the degree 5 polynomial and RBF kernels, the SCC generated similar values to PCC values shown in **Table** 3.

5 Conclusions

In this work, the KPCA projections were considered as a starting point to have an intuitive idea about the structure of the submanifold in the feature space F , where images were mapped. The two-dimensional plots reveal global variations and local structures that are not visible in the original pixel-value representation of the images (Ghojogh et al., 2019). In this case, as stated by Schölkopf et al. (1997), by using a kernel function k instead of the standard dot product, one implicitly performs PCA in a possible high-dimensional space F which is nonlinear related to the input space for MNIST, i.e., $\mathbb{R}^{784}$. In this sense, the correlations reported in **Table** 3 provide insight into the geometric factors represented by the KPCA components. Particularly, DL and AN exhibit strong correlation with the first KPCA component for the Polynomial and RBF kernels, indicating substantial explanatory strength. In contrast, several correlations involving DI, although statistically significant due to the large sample size, show weak effect sizes and therefore limited explanatory value. Consequently, the interpretation of the results emphasizes both statistical significance and the magnitude of the observed associations.

The Spearman correlation analysis corroborates the Pearson correlation results. DL and AN exhibit strong monotonic associations with the first KPCA component for the Polynomial and RBF kernels, indicating that these variables represent the primary sources of variation captured by the kernel projections. In contrast, DI shows consistently weak correlations across kernels and components, suggesting a limited contribution to the organization of the KPCA feature space. The close agreement between Pearson and Spearman coefficients further indicates that the dominant relationships are approximately monotonic and, in several cases, nearly linear.

Authors in Schölkopf et al. (1998), consider that choosing the optimal kernel for a given problem remains an open problem, both for Support Vector Machines and Kernel PCA. From this perspective, the reported correlations in this work could be considered as an empirical fact for selecting the optimal kernel in future experimental scenarios. Also, in a complementary manner, future work could investigate others spectral nonlinear techniques such as Local Linear Embedding, Multidimensional Scaling, Sammon mappings or Isomap, t-SNE, UMAP and repeat a similar experimental approach respect to the studied parameters from images.

Acknowledgements

This project was supported by Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) and Instituto Politécnico Nacional de México.

References

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(8), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

Bishop, C. M., & Bishop, H. (2024). Autoencoders. In C. M. Bishop & H. Bishop (Eds.), *Deep Learning: Foundations and Concepts* (pp. 563–579). Springer. https://doi.org/10.1007/978-3-031-45468-4_19

Burger, W., & Burge, M. J. (2022). *Digital image processing: An algorithmic introduction* (3rd ed.). Springer. https://doi.org/10.1007/978-3-031-05744-1

Castaneda-Diaz, R., Lopez-Betancur, D., Guerrero-Mendez, C., Gonzalez-Ramirez, E., Puma-Ttito, F., & Troncoso-Pacheco, R. (2026). Visual analysis of MNIST convolutional autoencoder reconstructions via linear dimensionality reduction. In L. Martínez-Villaseñor, R. A. Vázquez, G. Ochoa-Ruiz, M. Montes Rivera, S. Zapotecas-Martínez, M. L. Barrón-Estrada, E. Mezura-Montes, & A. Gómez-Chávez (Eds.), *Advances in Computational Intelligence. MICAI 2025 International Workshops* (Vol. 16264, pp. 352–367). Springer. https://doi.org/10.1007/978-3-032-17930-2_25

Castaneda-Diaz, R., Lopez-Betancur, D., Guerrero-Mendez, C., González-Ramírez, E., Gómez-Jiménez, S., & Puma-Ttito, F. (2025). Cleaning binary distortion on MNIST dataset. In L. Martínez-Villaseñor, G. Ochoa-Ruiz, M. Montes Rivera, M. L. Barrón-Estrada, & H. G. Acosta-Mesa (Eds.), *Advances in Computational Intelligence. MICAI 2024 International Workshops* (pp. 133–142). Springer. https://doi.org/10.1007/978-3-031-83879-8_11

Cayton, L. (2008). *Algorithms for manifold learning* (Technical Report CS2008-0923). University of California, San Diego.

de Winter, J. C. F., Gosling, S. D., & Potter, J. (2016). Comparing the Pearson and Spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. *Psychological Methods, 21*(3), 273–290. https://doi.org/10.1037/met0000079

Delete this duplicate. Keep only: van der Maaten, L. J. P., & Hinton, G. E. (2008).

Deng, X., Yuan, M., & Sudjianto, A. (2007). A note on robust kernel principal component analysis. In J. S. Verducci, X. Shen, & J. Lafferty (Eds.), *Prediction and Discovery* (Contemporary Mathematics, Vol. 443, pp. 21–34). American Mathematical Society. https://doi.org/10.1090/conm/443/08552

Fefferman, C., Mitter, S., & Narayanan, H. (2016). Testing the manifold hypothesis. *Journal of the American Mathematical Society, 29*(4), 983–1049. https://doi.org/10.1090/jams/852

Fisher, R. A. (1919). XV.—The correlation between relatives on the supposition of Mendelian inheritance. *Earth and Environmental Science Transactions of the Royal Society of Edinburgh, 52*(2), 399–433. https://doi.org/10.1017/S0080456800012163

Ghojogh, B. (2021). *Data reduction algorithms in machine learning and data science* [Doctoral dissertation, University of Waterloo]. UWSpace. https://uwspace.uwaterloo.ca/items/b30d8515-09d1-4063-98dd-1d13a4dc4e56

Ghojogh, B., Crowley, M., Karray, F., & Ghodsi, A. (2023). *Elements of Dimensionality Reduction and Manifold Learning*. Springer. https://doi.org/10.1007/978-3-031-10602-6

Ghojogh, B., Samad, M. N., Mashhadi, S. A., Kapoor, T., Ali, W., Karray, F., & Crowley, M. (2019). *Feature selection and feature extraction in pattern analysis: A literature review* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.1905.02845

Gonzalez, R. C., & Woods, R. E. (2018). *Digital image processing* (4th ed., global ed.). Pearson.

Gorban, A. N., Kégl, B., Wunsch, D. C., & Zinovyev, A. Y. (Eds.). (2008). *Principal Manifolds for Data Visualization and Dimension Reduction* (Vol. 58). Springer. https://doi.org/10.1007/978-3-540-73750-6

He, X., Yan, S., Hu, Y., Niyogi, P., & Zhang, H.-J. (2005). Face recognition using Laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(3), 328–340. https://doi.org/10.1109/TPAMI.2005.55

Hira, Z. M., & Gillies, D. F. (2015). A review of feature selection and feature extraction methods applied on microarray data. *Advances in Bioinformatics, 2015*, Article 198363. https://doi.org/10.1155/2015/198363

Jansen, A., & Niyogi, P. (2013). Intrinsic spectral analysis. *IEEE Transactions on Signal Processing, 61*(7), 1698–1710. https://doi.org/10.1109/TSP.2013.2238931

Jolliffe, I. T. (2014). Principal component analysis. In *Wiley StatsRef: Statistics Reference Online*. Wiley. https://doi.org/10.1002/9781118445112.stat06472

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539

McInnes, L., Healy, J., & Melville, J. (2018). *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.1802.03426

Mittal, M., J, P. G., M S, G. P., Devadas, R. M., Ambreen, L., & Kumar, V. (2024). Dimensionality reduction using UMAP and TSNE technique. In *2024 Second International Conference on Advances in Information Technology (ICAIT)* (Vol. 1, pp. 1–5). IEEE. https://doi.org/10.1109/ICAIT61638.2024.10690797

Rebekić, A., Lončarić, Z., Petrović, S., & Marić, S. (2015). Pearson's or Spearman's correlation coefficient—Which one to use? *Poljoprivreda, 21*(2), 47–54. https://doi.org/10.18047/poljo.21.2.8

Reverter, F., Vegas, E., & Oller, J. M. (2014). Kernel-PCA data integration with enhanced interpretability. *BMC Systems Biology, 8*(Suppl. 2), Article S6. https://doi.org/10.1186/1752-0509-8-S2-S6

Saul, L. K., Weinberger, K. Q., Sha, F., Ham, J., & Lee, D. D. (2006). Spectral methods for dimensionality reduction. In O. Chapelle, B. Schölkopf, & A. Zien (Eds.), *Semi-supervised learning*. MIT Press. https://doi.org/10.7551/mitpress/9780262033589.003.0016

Schölkopf, B., Mika, S., Burges, C. J. C., Knirsch, P., Müller, K.-R., Rätsch, G., & Smola, A. J. (1999). Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks, 10*(5), 1000–1017. https://doi.org/10.1109/72.788641

Schölkopf, B., Smola, A. J., & Müller, K.-R. (1999). Kernel principal component analysis. In B. Schölkopf, C. J. C. Burges, & A. J. Smola (Eds.), *Advances in kernel methods—Support vector learning* (pp. 327–352). MIT Press.

Schölkopf, B., Smola, A., & Müller, K.-R. (1997). Kernel principal component analysis. In W. Gerstner, A. Germond, M. Hasler, & J.-D. Nicoud (Eds.), *Artificial Neural Networks—ICANN'97* (pp. 583–588). Springer. https://doi.org/10.1007/BFb0020217

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation, 10*(5), 1299–1319. https://doi.org/10.1162/089976698300017467

Shah, F. P., & Patel, V. (2016). A review on feature selection and feature extraction for text classification. In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (pp. 2264–2268). IEEE. https://doi.org/10.1109/WiSPNET.2016.7566545

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press. https://doi.org/10.1017/CBO9780511809682

Strange, H., & Zwiggelaar, R. (2014). Spectral dimensionality reduction. In H. Strange & R. Zwiggelaar (Eds.), *Open Problems in Spectral Dimensionality Reduction* (pp. 7–22). Springer. https://doi.org/10.1007/978-3-319-03943-5_2

van der Maaten, L. J. P., & Hinton, G. E. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research, 9*, 2579–2605. https://www.jmlr.org/papers/v9/vandermaaten08a.html

van Vliet, L. J., & Verbeek, P. W. (1995). Estimators for orientation and anisotropy in digitized images. In *Proceedings of the First Annual Conference of the Advanced School for Computing and Imaging* (pp. 442–450). ASCI.

Walpole, R. E., Myers, R. H., Myers, S. L., & Ye, K. (2012). *Probabilidad y estadística para ingeniería y ciencias* (9.ª ed.). Pearson.

Wang, Q. (2012). *Kernel Principal Component Analysis and its Applications in Face Recognition and Active Shape Models* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.1207.3538

Yun, H. S., Jargal, A., Hyun, C. M., & Seo, J. K. (2023). Nonlinear representation and dimensionality reduction. In J. K. Seo (Ed.), *Deep Learning and Medical Applications* (pp. 1–49). Springer. https://doi.org/10.1007/978-981-99-1839-3_1