

# Principal Component Analysis Using Structural Similarity Index for Images

Benyamin Ghogh, Fakhri Karray, Mark Crowley

Department of Electrical and Computer Engineering,  
University of Waterloo, Waterloo, ON, Canada  
{bghogh, karray, mcrowley}@uwaterloo.ca

**Abstract.** Despite the advances of deep learning in specific tasks using images, the principled assessment of image fidelity and similarity is still a critical ability to develop. As it has been shown that Mean Square Error (MSE) is insufficient for this task, other measures have been developed with one of the most effective being Structural Similarity Index (SSIM). Such measures can be used for subspace learning but existing methods in machine learning, such as Principal Component Analysis (PCA), are based on Euclidean distance or MSE and thus cannot properly capture the structural features of images. In this paper, we define an image structure subspace which discriminates different types of image distortions. We propose Image Structural Component Analysis (ISCA) and also kernel ISCA by using SSIM, rather than Euclidean distance, in the formulation of PCA. This paper provides a bridge between image quality assessment and manifold learning opening a broad new area for future research.

**Keywords:** Principal component analysis, structural similarity, SSIM, image structural component analysis, image structure subspace

## 1 Introduction

It has been shown that Mean Square Error (MSE) is not a promising measure for image quality, fidelity, or similarity [1]. The distortions of an image or similarities of two images can be divided into two main categories, i.e., structural and non-structural distortions [2]. The structural distortions, such as JPEG blocking distortion, Gaussian noise, and blurring, are the ones which are easily noticeable by Human Visual System (HVS), whereas the non-structural distortions, such as luminance enhancement and contrast change, do not have large impact on the visual quality of image.

Structural similarity index (SSIM) [2, 3] has been shown to be an effective measure for image quality assessment. It encounters luminance and contrast change as non-structural distortions and other distortions as structural ones. Due to its performance, it has recently been noticed and used in optimization problems [4] for tasks such as image denoising, image restoration, contrast enhancement, image quantization, compression, etc, noticing that the distance based on SSIM is quasi-convex under certain conditions [5].

So far, the fields of manifold learning and machine learning have largely used MSE and Euclidean distance in order to develop algorithms for subspace learning. Principal Component Analysis (PCA) is an example based on Euclidean distance or  $\ell_2$  norm. However, MSE is not as promising as SSIM for image structure measurement [1, 3] making these algorithms not effective enough in terms of capturing the structural features of image. In this paper, we introduce the new concept of and *image structure subspace* which is a subspace capturing the intrinsic features of an image in terms of structural similarity and distortions, and can discriminate the various types of image distortions. This subspace can also be useful for parameter estimation for (or selection between) different denoising methods, but that topic will be dealt with in future work.

The outline and contributions of the paper are as follows: We begin by defining the background methods of SSIM and PCA. We then introduce ISCA using orthonormal bases and kernels by analogy to PCA, where ISCA can be seen as PCA which uses SSIM instead of the  $\ell_2$  norm. We then describe an extensive set of experiments demonstrating the performance of ISCA on projection, reconstruction and out-of-sample analysis tasks compared to various kernel PCA methods. The derivations of expressions in this paper are detailed more in the supplementary-material paper which will be released in <https://arXiv.org>.

## 2 Structural Similarity Index

The SSIM between two reshaped image blocks  $\check{\mathbf{x}}_1 = [x_1^{(1)}, \dots, x_1^{(q)}]^\top \in \mathbb{R}^q$  and  $\check{\mathbf{x}}_2 = [x_2^{(1)}, \dots, x_2^{(q)}]^\top \in \mathbb{R}^q$ , in color intensity range  $[0, l]$ , is [2, 3]:

$$\mathbb{R} \ni \text{SSIM}(\check{\mathbf{x}}_1, \check{\mathbf{x}}_2) := \left( \frac{2\mu_{x_1}\mu_{x_2} + c_1}{\mu_{x_1}^2 + \mu_{x_2}^2 + c_1} \right) \left( \frac{2\sigma_{x_1}\sigma_{x_2} + c_2}{\sigma_{x_1}^2 + \sigma_{x_2}^2 + c_2} \right) \left( \frac{\sigma_{x_1, x_2} + c_3}{\sigma_{x_1}\sigma_{x_2} + c_3} \right), \quad (1)$$

where  $\mu_{x_1} = (1/q) \sum_{i=1}^q x_1^{(i)}$ ,  $\sigma_{x_1} = \left[ (1/(q-1)) \sum_{i=1}^q (x_1^{(i)} - \mu_{x_1})^2 \right]^{0.5}$ ,  $\sigma_{x_1, x_2} = (1/(q-1)) \sum_{i=1}^q (x_1^{(i)} - \mu_{x_1})(x_2^{(i)} - \mu_{x_2})$ ,  $c_1 = (0.01 \times l)^2$ ,  $c_2 = 2c_3 = (0.03 \times l)^2$ , and  $\mu_{x_2}$  and  $\sigma_{x_2}$  are defined similarly for  $\check{\mathbf{x}}_2$ . In this work,  $l = 1$ . The  $c_1$ ,  $c_2$ , and  $c_3$  are for avoidance of singularity [3] and  $q$  is the dimensionality of the reshaped image patch. Note that since  $c_2 = 2c_3$ , we can simplify SSIM to  $\text{SSIM}(\check{\mathbf{x}}_1, \check{\mathbf{x}}_2) = s_1(\check{\mathbf{x}}_1, \check{\mathbf{x}}_2) \times s_2(\check{\mathbf{x}}_1, \check{\mathbf{x}}_2)$ , where  $s_1(\check{\mathbf{x}}_1, \check{\mathbf{x}}_2) := (2\mu_{x_1}\mu_{x_2} + c_1)/(\mu_{x_1}^2 + \mu_{x_2}^2 + c_1)$  and  $s_2(\check{\mathbf{x}}_1, \check{\mathbf{x}}_2) := (2\sigma_{x_1, x_2} + c_2)/(\sigma_{x_1}^2 + \sigma_{x_2}^2 + c_2)$ . If the vectors  $\check{\mathbf{x}}_1$  and  $\check{\mathbf{x}}_2$  have zero mean, i.e.,  $\mu_{x_1} = \mu_{x_2} = 0$ , the SSIM becomes  $\mathbb{R} \ni \text{SSIM}(\check{\mathbf{x}}_1, \check{\mathbf{x}}_2) = (2\check{\mathbf{x}}_1^\top \check{\mathbf{x}}_2 + c)/(\|\check{\mathbf{x}}_1\|_2^2 + \|\check{\mathbf{x}}_2\|_2^2 + c)$ , where  $c = (q-1)c_2$  [6]. We denote the reshaped vectors of the two images by  $\mathbf{x}_1 \in \mathbb{R}^d$  and  $\mathbf{x}_2 \in \mathbb{R}^d$ , and a reshaped block in the two images by  $\check{\mathbf{x}}_1 \in \mathbb{R}^q$  and  $\check{\mathbf{x}}_2 \in \mathbb{R}^q$ . The distance based on SSIM, which we denote by  $\|\cdot\|_S$ , is [6, 5, 7]:

$$\mathbb{R} \ni \|\check{\mathbf{x}}_1 - \check{\mathbf{x}}_2\|_S := 1 - \text{SSIM}(\check{\mathbf{x}}_1, \check{\mathbf{x}}_2) = \frac{\|\check{\mathbf{x}}_1 - \check{\mathbf{x}}_2\|_2^2}{\|\check{\mathbf{x}}_1\|_2^2 + \|\check{\mathbf{x}}_2\|_2^2 + c}, \quad (2)$$

where  $\mu_{x_1} = \mu_{x_2} = 0$ . In ISCA and PCA which inspires ISCA, the data should be centered; therefore, the fact that  $\check{\mathbf{x}}_1$  and  $\check{\mathbf{x}}_2$  should be centered is useful.

### 3 Principal Component Analysis

Since ISCA is inspired by PCA [8] we briefly review it here. Assume that the orthonormal columns of matrix  $\mathbf{U} \in \mathbb{R}^{d \times p}$  are the vectors which span the PCA subspace. Then, the projected data  $\widetilde{\mathbf{X}} \in \mathbb{R}^{p \times n}$  onto PCA subspace and the reconstructed data  $\hat{\mathbf{X}} \in \mathbb{R}^{d \times n}$  are  $\widetilde{\mathbf{X}} = \mathbf{U}^\top \mathbf{X}$  and  $\hat{\mathbf{X}} = \mathbf{U} \widetilde{\mathbf{X}} = \mathbf{U} \mathbf{U}^\top \mathbf{X}$ , respectively. The squared length of the projected data is  $\|\hat{\mathbf{X}}\|_F^2 = \|\mathbf{U} \mathbf{U}^\top \mathbf{X}\|_F^2 = \text{tr}(\mathbf{U}^\top \mathbf{X} \mathbf{X}^\top \mathbf{U})$  where  $\text{tr}(\cdot)$  and  $\|\cdot\|_F$  denote the trace and Frobenius norm of matrix, respectively. Presuming that the data  $\mathbf{X}$  are already centered, the  $\mathbf{S} = \mathbf{X} \mathbf{X}^\top$  is the covariance matrix; therefore:  $\|\hat{\mathbf{X}}\|_F^2 = \text{tr}(\mathbf{U}^\top \mathbf{S} \mathbf{U})$ . Maximizing the squared length of projection where the projection matrix is orthogonal is:

$$\begin{aligned} & \underset{\mathbf{U}}{\text{maximize}} \quad \text{tr}(\mathbf{U}^\top \mathbf{S} \mathbf{U}), \\ & \text{subject to} \quad \mathbf{U}^\top \mathbf{U} = \mathbf{I}, \end{aligned} \quad (3)$$

The Lagrangian [9] is:  $\mathcal{L} = \text{tr}(\mathbf{U}^\top \mathbf{S} \mathbf{U}) - \text{tr}(\mathbf{A}^\top (\mathbf{U}^\top \mathbf{U} - \mathbf{I}))$ , where  $\mathbf{A} \in \mathbb{R}^{p \times p}$  is a diagonal matrix including Lagrange multipliers. Equating derivative of  $\mathcal{L}$  to zero gives us:  $\mathbb{R}^{d \times p} \ni \partial \mathcal{L} / \partial \mathbf{U} = 2 \mathbf{S} \mathbf{U} - 2 \mathbf{U} \mathbf{A} \stackrel{\text{set}}{=} \mathbf{0} \implies \mathbf{S} \mathbf{U} = \mathbf{U} \mathbf{A}$ . Therefore, columns of  $\mathbf{U}$  are the eigenvectors of the covariance matrix  $\mathbf{S}$ .

PCA can be looked at with another point of view. The reconstruction error is  $\mathbf{R} := \mathbf{X} - \hat{\mathbf{X}} = \mathbf{X} - \mathbf{U} \mathbf{U}^\top \mathbf{X}$  where  $\mathbb{R}^{d \times n} \ni \mathbf{R} = [\mathbf{r}_1, \dots, \mathbf{r}_n]$  is the matrix of residuals. We want to minimize the reconstruction error:

$$\begin{aligned} & \underset{\mathbf{U}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{U} \mathbf{U}^\top \mathbf{X}\|_F^2, \\ & \text{subject to} \quad \mathbf{U}^\top \mathbf{U} = \mathbf{I}. \end{aligned} \quad (4)$$

The objective function is  $\|\mathbf{X} - \mathbf{U} \mathbf{U}^\top \mathbf{X}\|_F^2 = \text{tr}(\mathbf{X}^\top \mathbf{X} - \mathbf{X} \mathbf{X}^\top \mathbf{U} \mathbf{U}^\top)$ . The Lagrangian [9] is:  $\mathcal{L} = \text{tr}(\mathbf{X}^\top \mathbf{X}) - \text{tr}(\mathbf{X} \mathbf{X}^\top \mathbf{U} \mathbf{U}^\top) - \text{tr}(\mathbf{A}^\top (\mathbf{U}^\top \mathbf{U} - \mathbf{I}))$ , where  $\mathbf{A} \in \mathbb{R}^{p \times p}$  is a diagonal matrix including Lagrange multipliers. Equating the derivative of  $\mathcal{L}$  to zero gives:  $\partial \mathcal{L} / \partial \mathbf{U} = 2 \mathbf{X} \mathbf{X}^\top \mathbf{U} - 2 \mathbf{U} \mathbf{A} \stackrel{\text{set}}{=} \mathbf{0} \implies \mathbf{X} \mathbf{X}^\top \mathbf{U} = \mathbf{U} \mathbf{A} \implies \mathbf{S} \mathbf{U} = \mathbf{U} \mathbf{A}$ , which is again the eigenvalue problem for the covariance matrix  $\mathbf{S}$ . Therefore, PCA subspace is the best linear projection in terms of reconstruction error.

As shown above, PCA [8] is based on  $\ell_2$  (or Frobenius) norm which is not a promising measure for image quality assessment [1]. In order to have both the minimization of reconstruction error as in PCA and using a proper measure for image fidelity, we propose ISCA.

## 4 Image Structural Component Analysis (ISCA)

### 4.1 Orthonormal Bases for One Image

Our goal is to find a subspace spanned by  $p$  directions for some desired  $p$ . Consider an image block  $\check{\mathbf{x}} \in \mathbb{R}^q$  which is centered (its mean is removed). We want

to project it onto a  $p$ -dimensional subspace and then reconstruct it back, where  $p \leq q$ . Assume  $\mathbb{R}^{q \times p} \ni \mathbf{U} := [\mathbf{u}_1, \dots, \mathbf{u}_p]$  is a matrix whose columns are the projection directions spanning the subspace. The projection and reconstruction of  $\check{\mathbf{x}}$  are  $\mathbf{U}^\top \check{\mathbf{x}}$  and  $\mathbf{U}\mathbf{U}^\top \check{\mathbf{x}}$ , respectively. We want to minimize the reconstruction error with orthonormal bases of the subspace; therefore:

$$\begin{aligned} & \underset{\mathbf{U} \in \mathbb{R}^{q \times p}}{\text{minimize}} && \|\check{\mathbf{x}} - \mathbf{U}\mathbf{U}^\top \check{\mathbf{x}}\|_S, \\ & \text{subject to} && \mathbf{U}^\top \mathbf{U} = \mathbf{I}. \end{aligned} \quad (5)$$

According to Eq. (2) and noticing the orthonormality of projection directions,  $\mathbf{U}^\top \mathbf{U} = \mathbf{I}$ , we have:

$$\mathbb{R} \ni f(\mathbf{U}) := \|\check{\mathbf{x}} - \mathbf{U}\mathbf{U}^\top \check{\mathbf{x}}\|_S = \frac{\check{\mathbf{x}}^\top (\mathbf{I} - \mathbf{U}\mathbf{U}^\top) \check{\mathbf{x}}}{\check{\mathbf{x}}^\top (\mathbf{I} + \mathbf{U}\mathbf{U}^\top) \check{\mathbf{x}} + c}. \quad (6)$$

The gradient of the  $f(\mathbf{U})$  is:

$$\mathbb{R}^{q \times p} \ni \mathbf{G}(\mathbf{U}) := \frac{\partial f(\mathbf{U})}{\partial \mathbf{U}} = \frac{-2(1 + f(\mathbf{U}))}{\|\check{\mathbf{x}}\|_2^2 + \|\mathbf{U}\mathbf{U}^\top \check{\mathbf{x}}\|_2^2 + c} \check{\mathbf{x}} \check{\mathbf{x}}^\top \mathbf{U}. \quad (7)$$

We partition a  $d$ -dimensional image into  $b = \lceil d/q \rceil$  non-overlapping blocks each of which is a reshaped vector  $\check{\mathbf{x}} \in \mathbb{R}^q$ . The parameter  $q$  is an upper bound on the desired dimensionality of the subspace of block ( $p \leq q$ ). This parameter should not be a very large number due to the spatial variety of image statistics, yet also not very small so as to be able to capture the image structure. Also note that  $p$  is an upper bound on the rank of  $\mathbf{U}\mathbf{U}^\top \in \mathbb{R}^{q \times q}$ .

We have  $b$  instances of  $p$ -dimensional subspaces, one for each of the blocks. For projecting an image into the subspace and reconstructing it back, one can project and reconstruct every block of an image separately using the  $p$  bases of the block subspace. The overall bases of an image can be visualized in image-form by putting the bases of blocks next to each other (see the experiments in Section 6).

Considering all the  $b$  blocks in an image, the problem in Eq. (5) becomes:

$$\begin{aligned} & \underset{\mathbf{U}_i \in \mathbb{R}^{q \times p}}{\text{minimize}} && \sum_{i=1}^b \|\check{\mathbf{x}}_i - \mathbf{U}_i \mathbf{U}_i^\top \check{\mathbf{x}}_i\|_S, \\ & \text{subject to} && \mathbf{U}_i^\top \mathbf{U}_i = \mathbf{I}, \quad \forall i \in \{1, \dots, b\}, \end{aligned} \quad (8)$$

where  $\mathbf{x}_i \in \mathbb{R}^q$  and  $\mathbf{U}_i \in \mathbb{R}^{q \times p}$  are the  $i$ -th block and the bases of its subspace, respectively. We can embed the constraint as an indicator function in the objective function [10]:

$$\begin{aligned} & \underset{\mathbf{U}_i, \mathbf{V}_i \in \mathbb{R}^{q \times p}}{\text{minimize}} && \sum_{i=1}^b (f(\mathbf{U}_i) + h(\mathbf{V}_i)), \\ & \text{subject to} && \mathbf{U} - \mathbf{V} = \mathbf{0}, \end{aligned} \quad (9)$$

where  $f(\mathbf{U}_i) := \|\check{\mathbf{x}}_i - \mathbf{U}_i \mathbf{U}_i^\top \check{\mathbf{x}}_i\|_S$  and  $h(\mathbf{V}_i) := \mathbb{I}(\mathbf{V}_i^\top \mathbf{V}_i = \mathbf{I})$ . The  $\mathbb{I}(\cdot)$  denotes the indicator function which is zero if its condition is satisfied and is infinite otherwise. The  $\mathbf{U}$  and  $\mathbf{V}$  are defined as union of partitions to form an image-form array, i.e.,  $\mathbf{U} := \cup_{i=1}^b \mathbf{U}_i$  and  $\mathbf{V} := \cup_{i=1}^b \mathbf{V}_i$  [11].

The Eq. (9) can be solved using Alternating Direction Method of Multipliers (ADMM) [10, 11]. The augmented Lagrangian for Eq. (9) is:  $\mathcal{L}_\rho = \sum_{i=1}^b (f(\mathbf{U}_i) + h(\mathbf{V}_i)) + \text{tr}(\mathbf{A}^\top (\mathbf{U} - \mathbf{V})) + (\rho/2) \|\mathbf{U} - \mathbf{V}\|_F^2 = \sum_{i=1}^b (f(\mathbf{U}_i) + h(\mathbf{V}_i)) + (\rho/2) \|\mathbf{U} - \mathbf{V} + \mathbf{J}\|_F^2 - (\rho/2) \|\mathbf{A}\|_F^2$ , where  $\mathbf{A} := \cup_{i=1}^b \mathbf{A}_i$  is the Lagrange multiplier,  $\rho > 0$  is a parameter, and  $\mathbf{J} := (1/\rho)\mathbf{A} = (1/\rho) \cup_{i=1}^b \mathbf{A}_i = \cup_{i=1}^b \mathbf{J}_i$ . Note that the term  $(\rho/2) \|\mathbf{A}\|_F^2$  is a constant with respect to  $\mathbf{U}$  and  $\mathbf{V}$  and can be dropped. The updates of  $\mathbf{U}$ ,  $\mathbf{V}$ , and  $\mathbf{J}$  are done as [10, 11]:

$$\mathbf{U}_i^{(k+1)} := \arg \min_{\mathbf{U}_i} \left( f(\mathbf{U}_i) + (\rho/2) \|\mathbf{U}_i - \mathbf{V}_i^{(k)} + \mathbf{J}_i^{(k)}\|_F^2 \right), \quad (10)$$

$$\mathbf{V}_i^{(k+1)} := \arg \min_{\mathbf{V}_i} \left( h(\mathbf{V}_i) + (\rho/2) \|\mathbf{U}_i^{(k+1)} - \mathbf{V}_i + \mathbf{J}_i^{(k)}\|_F^2 \right), \quad (11)$$

$$\mathbf{J}^{(k+1)} := \mathbf{J}^{(k)} + \mathbf{U}^{(k+1)} - \mathbf{V}^{(k+1)}. \quad (12)$$

Considering  $\|\mathbf{A}\|_F^2 = \text{tr}(\mathbf{A}^\top \mathbf{A})$  for a matrix  $\mathbf{A}$ , the gradient of the objective function in Eq. (10) with respect to  $\mathbf{U}_i$  is  $\mathbf{G}(\mathbf{U}_i) + \rho(\mathbf{U}_i - \mathbf{V}_i^{(k)} + \mathbf{J}_i^{(k)})$  where  $\mathbf{G}(\mathbf{U}_i)$  is defined in Eq. (7). We can use the gradient decent method [9] for solving the Eq. (10). Our experiments showed that even one iteration of gradient decent suffices for Eq. (10) because the ADMM itself is iterative. Hence, we can replace this equation with one iteration of gradient decent.

The proximal operator is defined as [12]:

$$\text{prox}_{\lambda, h}(\mathbf{v}) := \arg \min_{\mathbf{u}} \left( h(\mathbf{u}) + (\lambda/2) \|\mathbf{u} - \mathbf{v}\|_2^2 \right), \quad (13)$$

where  $\lambda$  is the proximal parameter and  $h$  is the function that the proximal algorithm wants to minimize. According to Eq. (13), the Eq. (11) is equivalent to  $\text{prox}_{\rho, h}(\mathbf{U}_i^{(k+1)} + \mathbf{J}_i^{(k)})$ . As  $h(\cdot)$  is indicator function, its proximal operator is projection [12]. Therefore, Eq. (11) is equivalent to  $\Pi(\mathbf{U}_i^{(k+1)} + \mathbf{J}_i^{(k)})$  where  $\Pi(\cdot)$  denotes projection onto a set. Here, the variable of proximal operator is a matrix and not a vector. According to [12], if  $F$  is a convex and orthogonally invariant function, and it works on the singular values of a matrix variable  $\mathbf{A} \in \mathbb{R}^{q \times p}$ , i.e.,  $F = f \circ \sigma$  where the function  $\sigma(\mathbf{A})$  gives the vector of singular values of  $\mathbf{A}$ , then the proximal operator is:

$$\text{prox}_{\lambda, F}(\mathbf{A}) := \mathbf{Q} \text{diag} \left( \text{prox}_{\lambda, f}(\sigma(\mathbf{A})) \right) \mathbf{\Omega}^\top. \quad (14)$$

The  $\mathbf{Q} \in \mathbb{R}^{q \times p}$  and  $\mathbf{\Omega} \in \mathbb{R}^{p \times p}$  are the matrices of left and right singular vectors of  $\mathbf{A}$ , respectively. In our constraint  $\mathbf{V}^\top \mathbf{V} = \mathbf{I}$ , the function  $F$  deals with the singular values of  $\mathbf{V}$ . The reason is that we want:  $\mathbf{V} \stackrel{\text{SVD}}{=} \mathbf{Q} \mathbf{\Sigma} \mathbf{\Omega}^\top \implies \mathbf{V}^\top \mathbf{V} = \mathbf{\Omega} \mathbf{\Sigma} \mathbf{Q}^\top \mathbf{Q} \mathbf{\Sigma} \mathbf{\Omega}^\top \stackrel{(a)}{=} \mathbf{\Omega} \mathbf{\Sigma}^2 \mathbf{\Omega}^\top \stackrel{\text{set}}{=} \mathbf{I} \implies \mathbf{\Omega} \mathbf{\Sigma}^2 \mathbf{\Omega}^\top \mathbf{\Omega} = \mathbf{\Omega} \stackrel{(b)}{\implies} \mathbf{\Omega} \mathbf{\Sigma}^2 = \mathbf{\Omega} \implies$

$\Sigma = \mathbf{I}$ , where (a) and (b) are because  $\mathbf{Q}$  and  $\mathbf{\Omega}$  are orthogonal matrices. Therefore, we can use Eq. (14) for Eq. (11) where  $\text{prox}_{\rho,h}(\mathbf{U}_i^{(k+1)} + \mathbf{J}_i^{(k)})$  sets the singular values of  $(\mathbf{U}_i^{(k+1)} + \mathbf{J}_i^{(k)})$  to one. In summary, Eqs. (10), (11), and (12) can be restated as:

$$\begin{aligned}\mathbf{U}_i^{(k+1)} &:= \mathbf{U}_i^{(k)} - \eta \mathbf{G}(\mathbf{U}_i^{(k)}) - \eta \rho(\mathbf{U}_i^{(k)} - \mathbf{V}_i^{(k)} + \mathbf{J}_i^{(k)}), \\ \mathbf{V}_i^{(k+1)} &:= \mathbf{Q}_i \text{diag}\left(\text{prox}_{\rho,h}(\sigma(\mathbf{U}_i^{(k+1)} + \mathbf{J}_i^{(k)}))\right) \mathbf{\Omega}_i^\top, \\ \mathbf{J}^{(k+1)} &:= \mathbf{J}^{(k)} + \mathbf{U}^{(k+1)} - \mathbf{V}^{(k+1)},\end{aligned}\tag{15}$$

where columns of  $\mathbf{Q}_i \in \mathbb{R}^{q \times p}$  and  $\mathbf{\Omega}_i \in \mathbb{R}^{p \times p}$  are the left and right singular vectors of  $(\mathbf{U}_i^{(k+1)} + \mathbf{J}_i^{(k)})$  and  $\eta > 0$  is the learning rate. Iteratively solving Eq. (15) until convergence gives us the  $\mathbf{U}_i$  for the image blocks indexed by  $i$ . The  $p$  columns of  $\mathbf{U}_i$  are the bases for the *ISCA subspace* of the  $i$ -th block. Unlike in PCA, the ISCA bases do not have an order of importance but as in PCA, they are orthogonal capturing different features of image structure. The  $i$ -th projected block is  $\mathbf{U}_i^\top \check{\mathbf{x}}_i \in \mathbb{R}^p$  where its dimensions are *image structural components*. Note that  $\check{\mathbf{x}}_i$ , whether it is a block in a training image or an out-of-sample image, is centered. It is noteworthy that if we consider only one block in the images, the subscript  $i$  is dropped from Eq. (15).

## 4.2 Orthonormal Bases for a Set of Images

So far, if we have a set of  $n$  images, we can find the subspace bases  $\mathbf{U}_i$  for the  $i$ -th block in each of them using Eq. (15). Now, we want to find the subspace bases  $\mathbf{U}_i$  for the  $i$ -th block in *all training images of the dataset*. In other words, we want to find the subspace for the best reconstruction of the  $i$ -th block in all training images. For this goal, we can look at the optimization problem in Eq. (8) or (9) as an undercomplete auto-encoder neural network [13] with one hidden layer where the input layer, hidden layer, and output layer have  $q$ ,  $p$ , and  $q$  neurons, respectively. The  $\mathbf{U}_i^\top \check{\mathbf{x}}$  and  $\mathbf{U}_i \mathbf{U}_i^\top \check{\mathbf{x}}$  fill the role of applying the first and second weight matrices to the input, respectively. The weights are  $\mathbf{U}_i \in \mathbb{R}^{q \times p}$ . Therefore, we will have  $b$  auto-encoders, each with one hidden layer.

For training the auto-encoder, we introduce the blocks in an image as the input to this network and update the weights  $\mathbf{U}_i, \forall i$  based on Eq. (15). Note that we do this update of weights with only ‘one’ iteration of ADMM. Then, we move to the blocks in the next image and update the weights  $\mathbf{U}_i, \forall i$  again by an iteration of Eq. (15). We do this for all images one by one until an epoch is completed where an epoch is defined as introducing the block in all training images of dataset to the network. After termination of an epoch, we start another epoch to tune the weights  $\mathbf{U}_i, \forall i$  again. The epochs are repeated until the convergence. The termination criterion can be average reconstruction error  $(1/(nb)) \sum_{i=1}^b \sum_{j=1}^n \|\check{\mathbf{x}}_{j,i} - \mathbf{U}_i \mathbf{U}_i^\top \check{\mathbf{x}}_{j,i}\|_S < \varepsilon$ , where  $\varepsilon$  is a small number and  $\check{\mathbf{x}}_{j,i}$  is the  $i$ -th block in the  $j$ -th image. After training the network, we have one  $p$ -dimensional subspace for every block in all training images where the columns

of the weight matrix  $U_i$  span the subspace. Note that because of ADMM, the auto-encoders are trained simultaneously and in parallel. Again, the  $p$  columns of  $U_i$  are the bases for the *ISCA subspace* of the  $i$ -th block.

## 5 Kernel Image Structural Component Analysis

We can map the block  $\check{x} \in \mathbb{R}^q$  to higher-dimensional feature space hoping to have the data fall close to a simpler-to-analyze manifold in the feature space. Suppose  $\phi : \check{x} \rightarrow \mathcal{H}$  is a function which maps the data  $\check{x}_i$  to the feature space. In other words,  $\check{x} \mapsto \phi(\check{x})$ . Let  $t$  denote the dimensionality of the feature space, i.e.,  $\phi(\check{x}_i) \in \mathbb{R}^t$ . We usually have  $t \gg q$ . The kernel of the  $i$ -th block in images 1 and 2, which are  $\check{x}_{1,i}$  and  $\check{x}_{2,i}$ , is  $\phi(\check{x}_{1,i})^\top \phi(\check{x}_{2,i}) \in \mathbb{R}$  [14]. The kernel matrix for the  $i$ -th block among the  $n$  images is  $\mathbb{R}^{n \times n} \ni K_i := \Phi(\check{X}_i)^\top \Phi(\check{X}_i)$  where  $\Phi(\check{X}_i) := [\phi(\check{x}_{1,i}), \dots, \phi(\check{x}_{n,i})] \in \mathbb{R}^{t \times n}$ . After calculating the kernel matrix, we normalize it [15] as  $K_i(a, b) := K_i(a, b) / \sqrt{K_i(a, a)K_i(b, b)}$  where  $K_i(a, b)$  denotes the  $(a, b)$ -th element of the kernel matrix. Afterwards, the kernel is double-centered as  $K_i := HK_iH$  where  $\mathbb{R}^{n \times n} \ni H := I - (1/n)\mathbf{1}\mathbf{1}^\top$ . The reason for double-centering is that Eq. (2) requires  $\phi(\check{x}_i)$  and thus the  $\Phi(\check{X}_i)$  to be centered (see Eq. (16)). Therefore, in kernel ISCA, we center the kernel rather than centering  $\check{x}$ .

According to representation theory [16], the projection matrix can be expressed as a linear combination of the projected data points. Therefore, we have  $\mathbb{R}^{t \times p} \ni \Phi(U_i) = \Phi(\check{X}_i) \Theta_i$  where every column of  $\Theta_i := [\theta_1, \dots, \theta_p] \in \mathbb{R}^{n \times p}$  is the vector of coefficients for expressing a projection direction as a linear combination of projected image blocks.

As we did for ISCA, first we consider learning the  $b$  subspaces for ‘one’ image, here. Considering  $\Phi(U_i) = \Phi(\check{X}_i) \Theta_i$  for the  $i$ -th block in the image, the objective function of Eq. (8) in feature space is  $\sum_{i=1}^b \|\phi(\check{x}_i) - \Phi(\check{X}_i) \Theta_i \Theta_i^\top k_i\|_S$  where  $\mathbb{R}^n \ni k_i := \Phi(\check{X}_i)^\top \phi(\check{x}_i)$ . Note that  $\Phi(\check{X}_i)$  includes the mapping of the  $i$ -th block in all the  $n$  images while  $\phi(\check{x}_i)$  is the mapping of the  $i$ -th block in the image we are considering. The constraint of Eq. (8) in the feature space is  $\Phi(U_i)^\top \Phi(U_i) = \Theta_i^\top K_i \Theta_i = I$ . Therefore, the Eq. (8) in the feature space is:

$$\begin{aligned} & \underset{\Theta_i \in \mathbb{R}^{n \times p}}{\text{minimize}} && \sum_{i=1}^b \|\phi(\check{x}_i) - \Phi(\check{X}_i) \Theta_i \Theta_i^\top k_i\|_S, \\ & \text{subject to} && \Theta_i^\top K_i \Theta_i = I, \quad \forall i \in \{1, \dots, b\}. \end{aligned} \quad (16)$$

Noticing the constraint  $\Theta_i^\top K_i \Theta_i = I$  and using Eq. (2), we have:

$$\mathbb{R} \ni f(\Theta_i) := \|\phi(\check{x}_i) - \Phi(\check{X}_i) \Theta_i \Theta_i^\top k_i\|_S = \frac{k_i - k_i^\top \Theta_i \Theta_i^\top k_i}{k_i + k_i^\top \Theta_i \Theta_i^\top k_i + c}, \quad (17)$$

where  $\mathbb{R} \ni k_i := \phi(\check{x}_i)^\top \phi(\check{x}_i)$ . The gradient of the  $f(\Theta_i)$  is:

$$\mathbb{R}^{n \times p} \ni G(\Theta_i) := \frac{\partial f(\Theta_i)}{\partial \Theta_i} = \frac{-2(1 + f(\Theta_i))}{k_i + k_i^\top \Theta_i \Theta_i^\top k_i + c} k_i k_i^\top \Theta_i. \quad (18)$$

We can simplify the constraint  $\Theta_i^\top K_i \Theta_i = I$ . As the kernel  $K_i$  is positive semi-definite, we can decompose it as:

$$\mathbb{R}^{n \times n} \ni K_i \stackrel{\text{SVD}}{=} \Psi \Upsilon \Psi^\top = \Psi \Upsilon^{(1/2)} \Upsilon^{(1/2)} \Psi^\top = \Delta^\top \Delta,$$

where  $\mathbb{R}^{n \times n} \ni \Delta := \Upsilon^{(1/2)} \Psi^\top$ . Therefore, the constraint can be written as:  $\Theta_i^\top K_i \Theta_i = \Theta_i^\top \Delta^\top \Delta \Theta_i = (\Delta \Theta_i)^\top (\Delta \Theta_i) = I$ . In Eq. (16), if we embed the constraint in the objective function [10], we have:

$$\begin{aligned} & \underset{\Theta_i, V_i \in \mathbb{R}^{n \times p}}{\text{minimize}} && \sum_{i=1}^b (f(\Theta_i) + h(\Delta V_i)), \\ & \text{subject to} && \Theta - V = 0, \end{aligned} \quad (19)$$

where  $h(\Delta V_i) = \mathbb{I}((\Delta V_i)^\top (\Delta V_i) = I)$  and  $\Theta := \cup_{i=1}^b \Theta_i$  and  $V := \cup_{i=1}^b V_i$ . Taking  $\mathbb{R}^{n \times p} \ni W_i := \Delta V_i$ , we can restate Eq. (19) as:  $\underset{\Theta_i, W_i}{\text{minimize}} \sum_{i=1}^b (f(\Theta_i) + h(W_i))$ , subject to  $\Delta \Theta - W = 0$ , where  $W := \cup_{i=1}^b W_i$ . The ADMM solution to this optimization problem is [10, 11]:

$$\Theta_i^{(k+1)} := \arg \min_{\Theta_i} \left( f(\Theta_i) + (\rho/2) \|\Delta \Theta_i - W_i^{(k)} + J_i^{(k)}\|_F^2 \right), \quad (20)$$

$$W_i^{(k+1)} := \arg \min_{W_i} \left( h(W_i) + (\rho/2) \|\Delta \Theta_i^{(k+1)} - W_i + J_i^{(k)}\|_F^2 \right), \quad (21)$$

$$J^{(k+1)} := J^{(k)} + \Delta \Theta^{(k+1)} - W^{(k+1)}. \quad (22)$$

With the similar explanations which we had for Eq. (15), we have:

$$\begin{aligned} \Theta_i^{(k+1)} &:= \Theta_i^{(k)} - \eta G(\Theta_i^{(k)}) - \eta \rho \Delta^\top (\Delta \Theta_i^{(k)} - W_i^{(k)} + J_i^{(k)}), \\ W_i^{(k+1)} &:= Q_i \text{diag} \left( \text{prox}_{\rho, h}(\sigma(\Delta \Theta_i^{(k+1)} + J_i^{(k)})) \right) \Omega_i^\top, \\ J^{(k+1)} &:= J^{(k)} + \Delta \Theta^{(k+1)} - W^{(k+1)}, \end{aligned} \quad (23)$$

where columns of  $Q_i \in \mathbb{R}^{q \times p}$  and  $\Omega_i \in \mathbb{R}^{p \times p}$  are the left and right singular vectors of  $(\Delta \Theta_i^{(k+1)} + J_i^{(k)})$ . Iteratively solving Eq. (23) until convergence gives us the  $\Theta_i$  for the image blocks indexed by  $i$ . The  $p$  columns of  $\Theta_i$  are the bases for the *kernel ISCA subspace* of the  $i$ -th block. The  $i$ -th projected block is  $\Theta_i^\top k_i \in \mathbb{R}^p$  and its dimensions are the *kernel image structural components*. Note that  $k_i$ , whether it is the kernel over a block in a training image or an out-of-sample image, is normalized and centered. Also, note that we had considered the blocks of only one image for Eq. (23). Again, with the auto-encoder approach, we can solve these equations in successive epochs in order to find the  $b$  subspaces for all the  $n$  training images.

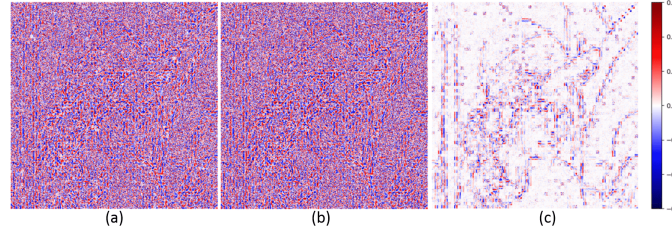
## 6 Experiments

**Training Dataset:** We formed a dataset out of the standard *Lena* image. Six different types of distortions were applied on the original *Lena* image (see Fig. 1),





**Fig. 1.** Examples from the training dataset: (a) original image, (b) contrast stretched, (c) Gaussian noise, (d) luminance enhanced, (e) Gaussian blurring, (f) salt & pepper impulse noise, and (g) JPEG distortion.

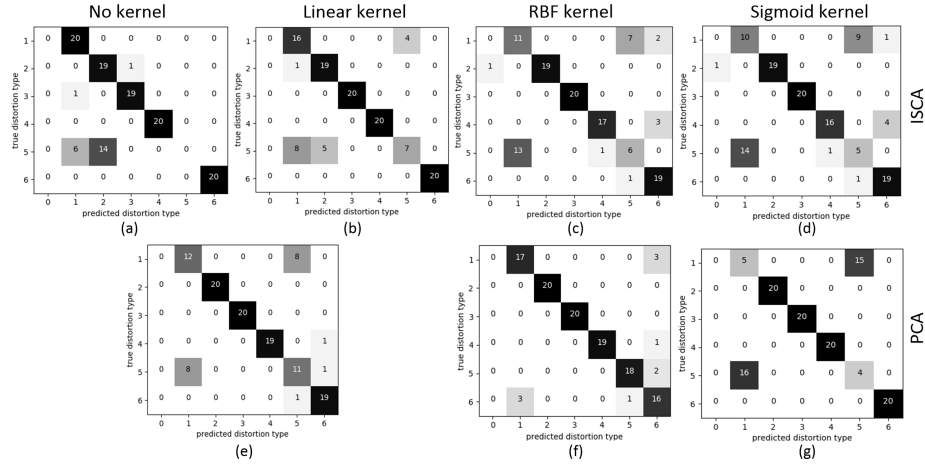


**Fig. 2.** The first dimension of the trained (a)  $\mathbf{U}$ , (b)  $\mathbf{V}$ , and (c)  $\mathbf{J}$  for ISCA.

each of which has 20 images in the dataset with different MSE values. Therefore, the size of the training set is 121 including the original image. For every type of distortion, 20 different levels of MSE, i.e., from  $\text{MSE} = 45$  to  $\text{MSE} = 900$  with step 45, were generated to have images on the equal-MSE or *iso-error* hypersphere [3].

**Training:** In our experiments for ISCA, the parameters used were  $\rho = 1$  and  $\eta = 0.1$ , and for kernel ISCA, we used  $\rho = 0.1$  and  $\eta = 0.1$ . We took  $q = 64$  ( $8 \times 8$  blocks inspired by [6, 11]),  $p = 4$ , and  $d = 512 \times 512 = 262144$ . One of the dimensions of the trained  $\mathbf{U} = \cup_{i=1}^b \mathbf{U}_i$ ,  $\mathbf{V} = \cup_{i=1}^b \mathbf{V}_i$ , and  $\mathbf{J} = \cup_{i=1}^b \mathbf{J}_i$  for ISCA are shown in Fig. 2. The dual variable  $\mathbf{J}$  has captured the edges because edges carry much of the structure information. As expected,  $\mathbf{U}$  and  $\mathbf{V}$  are close (*Lena* can be seen in them by noticing scrupulously). Note that the variables in kernel ISCA are not  $q$ -dimensional and thus cannot be displayed in image form.

**Projections and Comparisons:** In order to evaluate the trained ISCA and kernel ISCA subspaces, we projected the training images onto these subspaces. For projecting an image, each of its blocks is projected onto the subspace of that block. After projecting all the images, we used the 1-Nearest Neighbor (1NN) classifier to recognize the distortion type of every block. The 1NN is useful to evaluate the subspace by closeness of the projected distortions. The distortion type of an image comes from a majority vote among the blocks. The linear, Radial Basis function (RBF), and sigmoid kernels were tested for kernel ISCA. The confusion matrices for distortion recognition are shown in Fig. 3. Mostly kernel ISCA performed better than ISCA because it works in feature space; although, ISCA performed better for some distortions like contrast stretching



**Fig. 3.** Confusion matrices for recognition of distortion types with a 1NN classifier used in the subspace. Matrices (a) and (e) correspond to ISCA and PCA (or linear-kernel PCA), respectively. Matrices (b) to (d) are for kernel ISCA with linear, RBF, and sigmoid kernels. Matrices (f) and (g) are for kernel PCA with RBF and sigmoid kernels. The 0 label in matrices correspond to the original image and the labels 1 to 6 are the distortion types with the same order as in Fig. 1.



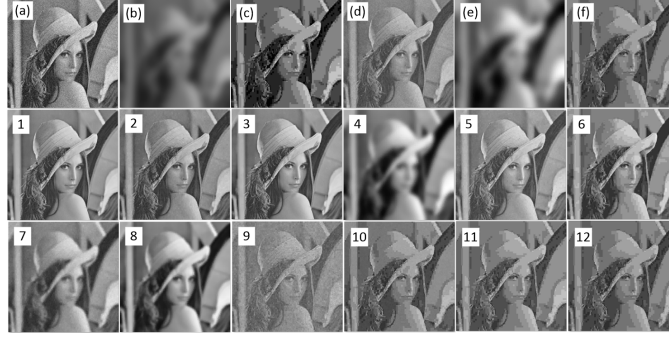
**Fig. 4.** Out-of-sample images with different types of distortions having  $MSE = 500$ : (1) stretching contrast, (2) Gaussian noise, (3) luminance enhancement, (4) Gaussian blurring, (5) impulse noise, (6) JPEG distortion, (7) Gaussian blurring + Gaussian noise, (8) Gaussian blurring + luminance enhancement, (9) impulse noise + luminance enhancement, (10) JPEG distortion + Gaussian noise, (11) JPEG distortion + luminance enhancement, and (12) JPEG distortion + stretching contrast.

and blurring. Moreover, we compared with PCA and kernel PCA. PCA showed weakness in contrast stretching. RBF and sigmoid kernels in kernel PCA do not perform well for JPEG distortion and contrast stretching, respectively.

**Out-of-sample Projections:** For out-of-sample projection, we created 12 test images with  $MSE = 500$  having different distortions and some having a combination of different distortions (see Fig. 4). We did the same 1NN classification for these images. Table 1 reports the top two votes of blocks for every image with the percentage of blocks voting for those distortions. ISCA did not recognize

**Table 1.** Recognition of distortions for out-of-sample images. Letters O, C, G, L, B, I, and J correspond to original image, contrast stretch, Gaussian noise, luminance enhanced, blurring, impulse noise, and JPEG distortion, respectively.

image	1	2	3	4	5	6	7	8	9	10	11	12
distortion	C	G	L	B	I	J	B + G	B + L	I + L	J + G	J + L	J + C
ISCA	69.3% O 30.2% C	49.1% G 27.2% I	69.7% O 29.6% C	99.8% B 0.2% J	30.3% G 23.8% I	96.4% J 3.6% B	55.2% B 19.9% G	98.7% B 1.3% J	48.9% G 33.3% I	39.4% J 32.9% B	96.4% J 3.6% B	97.9% J 2.1% B
kernel ISCA (linear)	88.1% C 11.2% I	59.2% G 25.2% I	99.8% L 0.1% O	95.9% B 3.4% J	37.4% G 32.3% I	80.4% J 17.6% B	40.8% G 33.4% B	93.4% B 5.8% J	45.6% I 39.4% G	38.7% G 21.7% J	70.2% J 27.0% B	74.1% J 25.0% B
kernel ISCA (RBF)	72.0% C 10.9% I	79.1% G 5.0% B	99.2% L 0.5% G	70.6% B 13.1% C	39.6% I 36.4% C	74.3% J 13.5% C	44.8% G 28.5% B	88.1% L 6.6% B	48.2% L 37.7% G	33.1% G 21.6% L	82.8% L 8.9% G	43.1% J 30.0% B
kernel ISCA (sigmoid)	80.3% C 7.6% I	76.3% G 6.8% I	99.6% L 0.2% G,B	76.2% B 10.6% J	38.5% I 36.5% C	79.3% J 10.6% C	47.9% G 24.6% B	81.7% L 9.8% B	52.1% L 26.6% G	37.9% G 19.8% L	80.7% L 11.3% G	43.9% J 31.0% B



**Fig. 5.** Reconstruction of images in ISCA. Reconstruction of the training images (a), (b), and (c) are shown in (d), (e), and (f), respectively. The reconstruction of out-of-sample images shown in Fig. 4 are shown in the second and third rows.

luminance enhancement well enough because, for Eq. (2), the block is centered while in kernel ISCA, the block is centered in feature space. Overall, both ISCA and kernel ISCA performed very compelling even in recognizing the combination of distortions.

**Reconstruction:** The images can be reconstructed after the projection onto the ISCA subspace. For reconstruction, every block is reconstructed as  $U_i U_i^T \tilde{x}_i \in \mathbb{R}^q$  where the mean of block should be added to the reconstruction. Similar to kernel PCA, reconstruction cannot be done in kernel ISCA because  $\Theta_i \Theta_i^T k_i \in \mathbb{R}^n \neq \mathbb{R}^q$ . Figure 5 shows reconstruction of some of training and out-of-sample images. As expected, the reconstructed images, for both training and out-of-sample images, are very similar to the original images.

## 7 Conclusion and Future Direction

This paper introduces the concept of an image structure subspace which captures the structure of an image and discriminates the distortion types. We hope this

will open a broad new field for research in this area and build a greatly needed bridge between the worlds of image quality assessment and manifold learning.

For image structure subspace learning, ISCA and kernel ISCA were proposed, taking inspiration from PCA. As future work, we can consider designing deeper auto-encoder [13] with non-linear activation functions for image structure subspace learning.

## References

1. Wang, Z., Bovik, A.C.: Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine* **26**(1) (2009) 98–117
2. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* **13**(4) (2004) 600–612
3. Wang, Z., Bovik, A.C.: Modern image quality assessment. *Synthesis Lectures on Image, Video, and Multimedia Processing* **2**(1) (2006) 1–156
4. Brunet, D., Channappayya, S.S., Wang, Z., Vrsnay, E.R., Bovik, A.C.: Optimizing image quality. In: *Handbook of Convex Optimization Methods in Imaging Science*. Springer (2018) 15–41
5. Brunet, D., Vrsnay, E.R., Wang, Z.: On the mathematical properties of the structural similarity index. *IEEE Transactions on Image Processing* **21**(4) (2012) 1488–1499
6. Otero, D., Vrsnay, E.R.: Unconstrained structural similarity-based optimization. In: *International Conference Image Analysis & Recognition*, Springer (2014) 167–176
7. Brunet, D., Vrsnay, E.R., Wang, Z.: A class of image metrics based on the structural similarity quality index. In: *International Conference Image Analysis and Recognition*, Springer (2011) 100–110
8. Jolliffe, I.: *Principal component analysis*. Springer (2011)
9. Boyd, S., Vandenberghe, L.: *Convex optimization*. Cambridge university press (2004)
10. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* **3**(1) (2011) 1–122
11. Otero, D., La Torre, D., Michailovich, O.V., Vrsnay, E.R.: Alternate direction method of multipliers for unconstrained structural similarity-based optimization. In: *International Conference Image Analysis & Recognition*, Springer (2018) 20–29
12. Parikh, N., Boyd, S.: Proximal algorithms. *Foundations and Trends® in Optimization* **1**(3) (2014) 127–239
13. Goodfellow, I., Bengio, Y., Courville, A.: *Deep learning*. Volume 1. MIT press Cambridge (2016)
14. Hofmann, T., Schölkopf, B., Smola, A.J.: Kernel methods in machine learning. *The annals of statistics* (2008) 1171–1220
15. Ah-Pine, J.: Normalized kernels as similarity indices. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer (2010) 362–373
16. Alperin, J.L.: *Local representation theory: Modular representations as an introduction to the local representation theory of finite groups*. Volume 11. Cambridge University Press (1993)