IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 June 2023, accepted 31 August 2023, date of publication 15 September 2023, date of current version 5 October 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3315831*

RESEARCH ARTICLE

# Image Recognition and Reconstruction With Machine Learning: An Inverse Problem Approach

**WIESLAW CITKO AND WIESLAW SIENKO, (Life Member, IEEE)**
Faculty of Electrical Engineering, Gdynia Maritime University, 81-225 Gdynia, Poland
Corresponding author: Wieslaw Citko (w.citko@we.umg.edu.pl)

**ABSTRACT** Image recognition and reconstruction are common problems in many image processing systems. These problems can be formulated as a solution to the linear inverse problem. This article presents a machine learning system model that can be used in the reconstruction and recognition of vectorized images. The analyzed inverse problem is given by the equations $F(\boldsymbol{x_i}) = \boldsymbol{y}_i$ and $\boldsymbol{x}_i = F^{-1}(\boldsymbol{y}_i)$, $i = 1, \ldots, N$, where $F(\cdot)$ is a linear mapping for $\boldsymbol{x}_i \in X \subset R^n$, $\boldsymbol{y}_i \in Y \subset R^m$. Thus, $\boldsymbol{y}_i$ can be seen as a projection of image $\boldsymbol{x}_i$, and $\boldsymbol{x}_i$ should be reconstructed as a solution to the inverse problem. We consider image reconstruction as an inverse problem using two different schemes. The first one, when $\boldsymbol{x}_i = F^{-1}(\boldsymbol{y}_i)$, can be seen as an operation with associative memory, and the second one, when $\boldsymbol{x}_i = F^{-1}(\boldsymbol{y}_i)$, can be implemented by creating random vectors for training sets. Moreover, we point out that the solution to the inverse problem can be generalized to complex-valued images $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$. In this paper, we propose a machine learning model based on a spectral processor as an alternative solution to deep learning based on optimization procedures.

**INDEX TERMS** Machine learning systems, image reconstruction and recognition, inverse problem.

## I. INTRODUCTION

The current trend in image reconstruction and recognition systems involves supplementing and partially replacing the classical methods with artificial intelligence systems, which incorporate machine learning algorithms. Various papers [1], [2], [3], [4] have reviewed machine learning methods in image recognition, with particular emphasis on medical applications. It is worth noting that modern image reconstruction and recognition systems typically rely on optimization algorithms with constraints while utilizing appropriately selected regularization methods [5].

Deep learning algorithms have recently gained widespread usage and have sparked a renewed interest in artificial neural networks and their applications. The majority of the known deep learning algorithms are implemented using deep learning neural networks, which are trained by minimizing a loss function on a training set. This approach to deep learning can be considered a special case within the theory of optimization methods. Standard types of deep learning neural networks include multilayer perceptrons (MLPs), convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs) [6], [7], [8], [9]. After analyzing the available publications, it can be concluded that the optimal network topology and the technology of its implementation have not yet been determined. The relationship between the network topology and its performance has not been properly investigated [10]. However, it is evident that neural networks (NNs) are universally algorithmic and physical models utilized in computational intelligence systems. Hopfield neural networks, a subtype of NNs, are physical models and algorithms used in neural computing. In previous papers, we proposed an extended Hopfield neural network model defined by the following equation [11]:

$$\dot{x} = (\eta \boldsymbol{W} - w_0 \boldsymbol{1} + \varepsilon \boldsymbol{W}_s)\, \boldsymbol{\theta}(\boldsymbol{x}) + \boldsymbol{I}_d \tag{1}$$

where $\boldsymbol{W}$— Antisymmetric orthogonal matrix
$\boldsymbol{W}_s$— Real symmetric matrix
$\boldsymbol{1}$— Identity matrix
$\boldsymbol{\theta}(\boldsymbol{x})$ — Activation function vector
$\boldsymbol{I}_d$— Input vector
$\varepsilon$, $w_0$, $\eta$— Parameters.

The associate editor coordinating the review of this manuscript and approving it for publication was Wenbing Zhao.

The equilibrium state of the network (1) takes the following form:

$$(\eta \boldsymbol{W} - w_0 \mathbf{1} + \varepsilon \boldsymbol{W}_s)\, \boldsymbol{\theta}\,(\boldsymbol{x}) + \boldsymbol{I}_d = \mathbf{0} \tag{2}$$

Equation (2) constitutes the basis for universal machine learning models based on biorthogonal transformations, enabling the implementation of typical learning system functions. One of these functions is the implementation of associative memories. The use of the system for reconstruction and recognition of distorted/noisy images with the use of associative memory was described in more detail in papers [11], [12], [13].

In this paper, we investigated the implementation of the machine learning system to solve inverse problems. In the experiments presented in the paper, the original image was processed by a linear matrix operator, the dimensions of which were not square. Thus, there was no inverse operator in the sense of matrix algebra. The appropriately designed machine learning system reconstructed the original image based on its projection.

## II. HOPFIELD NAURAL NETWORK AS A MACHINE LEARNING SYSTEM

By machine learning, we mean here input-output mapping approximation, where nodes of approximation are given by the set of training pairs $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N}$, $\boldsymbol{x}_i \in X \subset R^n, \boldsymbol{y}_i \in Y \subset R^m$. Hence, one aims to realize the mapping $F\colon X \to Y$, where the value of such a mapping (or multivariable function $f(\cdot)$ for $\boldsymbol{y}_i \in Y \subset R$) is known at the training points. Associative memory realization, classification, and pattern recognition issues can be seen as important problems in mapping approximation. As mentioned above, a nonlinear mapping

$$\boldsymbol{y}_i = F\,(\boldsymbol{x}_i)\,, \quad i = 1, \ldots, N \tag{3}$$

can be realized in the form of a Hopfield neural network as follows.

For training vectors $\boldsymbol{x}_i \in R^n$ and $\boldsymbol{y}_i \in R^m$, one defines the following input vectors $\boldsymbol{I}_d$:

$$\boldsymbol{I}_d = \boldsymbol{x}_i$$

where $dim \boldsymbol{I}_d = n$.

Hence Equation (1) takes the form:

$$\dot{s} = (\boldsymbol{W} - w_0 \mathbf{1} + \boldsymbol{W}_s)\, \boldsymbol{\Theta}(\boldsymbol{s}) + \boldsymbol{I}_d \tag{4}$$

where $\boldsymbol{s}(t)$ is the state vector of this network.

Thus, the equilibrium state of the network (4) takes the following form:

$$(\boldsymbol{W} - w_0 \mathbf{1} + \boldsymbol{W}_s)\, \boldsymbol{\Theta}\,(\boldsymbol{s}) + \boldsymbol{I}_d = \mathbf{0}. \tag{5}$$

It is clear that the neural networks described by Equation (4) can be classified as a recurrent neural network. Due to the assumption that matrix $\boldsymbol{W}$ is determined as orthogonal antisymmetric, the structures of this neural network can be composed by the connection of pairs of dynamic neurons. An example of such structures and their description is presented in Fig. 1.

$$\begin{bmatrix} \dot{s_1} \\ \dot{s_2} \end{bmatrix} = \left( \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix} - w_0 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \right) \begin{bmatrix} \Theta(s_1) \\ \Theta(s_2) \end{bmatrix} + \begin{bmatrix} I_{d_1} \\ I_{d_2} \end{bmatrix};\ w_{12} = w_{21} \tag{6}$$

The realization of mapping (3) by the dynamical networks (4) demands that in the synaptic connections' learning procedure, one should create $N$ attractors. Alternatively, we treat the equation state (5) as a pure algebraic algorithm to realize mapping (3). It gives rise to a machine learning system. The key point in the learning/synthesis of such a system is a reformulation of training points to the form:

$$\boldsymbol{u}_i = \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \end{bmatrix}, \quad i = 1, \ldots, N \tag{7}$$

where $dim \boldsymbol{u}_i = n + m, n + m = 2^q, q = 2, 3, 4, \ldots$ and, further, using an orthogonal transformation, a creation of the spectra $\boldsymbol{m}_i = T\,(\boldsymbol{u}_i)\,, i = 1, \ldots, N$ :

$$\boldsymbol{m}_i = \frac{1}{2}\,(\boldsymbol{W} + \mathbf{1})\, \boldsymbol{u}_i = T(\boldsymbol{u}_i) \tag{8}$$

where $\boldsymbol{W}$ – antisymmetric orthogonal matrix: $\boldsymbol{W}^2 = -\mathbf{1}$.

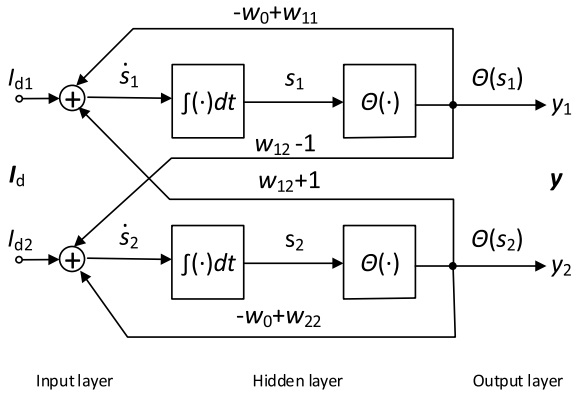



**FIGURE 1.** Model of a two-neuron network.

The stable equilibria of the neural network (4) constitute the pattern for the following transformation $T_s\,(\cdot)$:

$$(\boldsymbol{W} - w_0 \cdot \mathbf{1} + \boldsymbol{W}_s)\, \boldsymbol{m}_i + \boldsymbol{u}_i = \mathbf{0} \tag{9}$$

where $\boldsymbol{u}_i$ — input vectors (7).

For $w_0 = 2$, one obtains $\boldsymbol{m_i} = T_s\,(\boldsymbol{u_i})$

$$\boldsymbol{m}_i = (2 \cdot \mathbf{1} - \boldsymbol{W}_s - \boldsymbol{W})^{-1}\, \boldsymbol{u}_i \tag{10}$$

i.e., $T_s\,(\cdot) = (2 \cdot \mathbf{1} - \boldsymbol{W}_s - \boldsymbol{W})^{-1}\,(\cdot)$

where $\boldsymbol{W}_s = \boldsymbol{M}\left(\boldsymbol{M}^T \boldsymbol{M}\right)^{-1} \boldsymbol{M}^T$

and

$$\boldsymbol{M} = \{\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_N\} \tag{11}$$

is the spectrum matrix of $\boldsymbol{u_i}$ from Equation (8).

It should be noted that $\left(\boldsymbol{M}^T \boldsymbol{M}\right)^{-1} \boldsymbol{M}^T$ is the Moore-Penrose pseudoinverse matrix of $\boldsymbol{M}$, i.e.,

$\boldsymbol{M}^{(+)} = \lim_{\mu \to 0} \left(\boldsymbol{M}^T \boldsymbol{M} + \mu \boldsymbol{1}\right)^{-1} \boldsymbol{M}^T$ always exists. Thus, $\boldsymbol{M}(\boldsymbol{M}^T \boldsymbol{M} + \mu \boldsymbol{1})^{-1} \boldsymbol{M}^T$, $\mu \neq 0$ can be seen as Tikhonov's regularization [1]. It is clear that the transformation $T_s(\cdot)$ projects training points $\boldsymbol{u}_i$ into $\boldsymbol{m}_i$ as given by Equation (10). Hence, one obtains an inverse transformation:

$$\boldsymbol{u_i} = T^{-1}(\boldsymbol{m}_i) = (-\boldsymbol{W} + \boldsymbol{1})\, \boldsymbol{m}_i \tag{12}$$

i.e., $T^{-1}(\cdot) = (-\boldsymbol{W} + \boldsymbol{1})(\cdot)$.

The transformations $T_s(\cdot)$ and $T^{-1}(\cdot)$, arranged as a realization of a mapping $F(\boldsymbol{x})$, have the block structure as shown in Fig. 2.

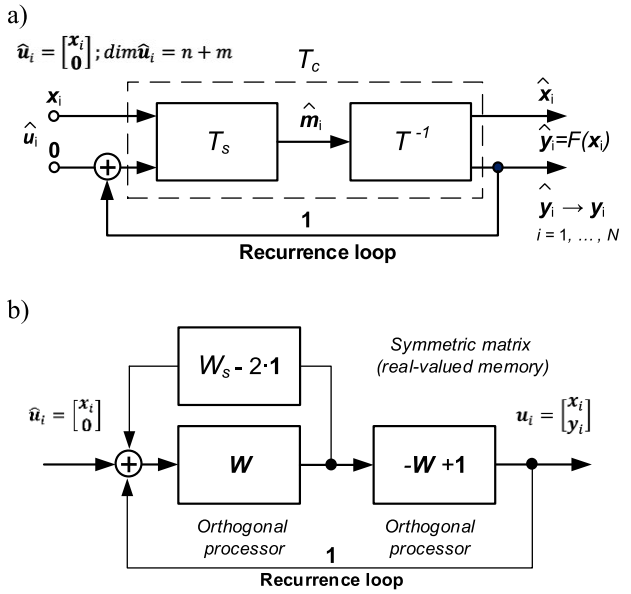



**FIGURE 2. Block structure of machine learning system as the approximator. (a) block structure with distributed memory. (b) block structure with lumped memory.**

The block structure with "distributed memory," presented in Fig. 2(a), can be reconfigured to the form with "lumped memory," as shown in Fig. 2(b).

*Note 1:*

It is worth noting that, according to the structure from Fig. 2, such an approximator performs the function of spectrum estimation $\{\hat{\boldsymbol{m}}_i\}$:

$$\hat{\boldsymbol{m}}_i = T_s \left( \begin{bmatrix} \boldsymbol{x}_i \\ \cdots \\ \boldsymbol{0} \end{bmatrix} \right), \quad i = 1, \ldots, N \tag{13}$$

and $\hat{\boldsymbol{y}}_i = F(\boldsymbol{x}_i)$ – estimation of the output $\boldsymbol{y}_i$; $i = 1, \ldots, N$.

Hence, due to the feedback loop action, one implements a recurrence:

$$\begin{aligned} &\hat{\boldsymbol{m}}_i \to \boldsymbol{m}_i, \hat{\boldsymbol{y}}_i \to \boldsymbol{y}_i \\ &\hat{\boldsymbol{y}}_i \to \boldsymbol{y}_i = F(\boldsymbol{x}_i), \quad i = 1, \ldots, N \end{aligned} \tag{14}$$

at the output of this approximator.

It is easy to note that the structure from Fig. 2 implements an input-output mapping $\phi(\cdot)$, as follows:

$$\boldsymbol{u}_i = \phi(\boldsymbol{u}_i), \quad i = 1, \ldots, N. \tag{15}$$

Thus, vectors $\boldsymbol{u}_i, i = 1, \ldots, N$ are invariant points of $\phi(\cdot)$, and vectors $\boldsymbol{u}_i$ are asymptotic centers of attractors $i = 1, \ldots, N$. Moreover, mapping $\phi(\cdot)$ is given by the following matrix transformation:

$$\phi(\cdot) = \boldsymbol{L}(\varepsilon) = (-\boldsymbol{W} + \boldsymbol{1})(2 \cdot \boldsymbol{1} - \boldsymbol{W} - \varepsilon \boldsymbol{W}_s)^{-1} \tag{16}$$

and its Lipschitz constant $k$ fulfills:

$$k \leq 1 \text{ for } \varepsilon \leq 1. \tag{17}$$

Hence, $\phi(\cdot)$ is a non-expansive mapping. Note the block $\boldsymbol{T}_c$ in Fig. 2 implementing this mapping. The recurrence is convergent under the linear independence of patterns $\boldsymbol{u}_i$ and the number of patterns $N$ must fulfill:

$$N < 0.5(n + m) \tag{18}$$

where $n + m = \dim \boldsymbol{u}_i$ (7).

## III. MACHINE LEARNING SYSTEM FOR IMAGE PROCESSING

The general learning algorithm described in the previous section can be used to solve the image processing tasks. We consider a set of $N$ black-and-white images represented by $k$ rows and $l$ columns (i.e., a set of $(k \cdot l)$ pixels with different shades of gray). In the case of vector analysis, each image is vectorized into the column vector $\boldsymbol{x_i}(k \cdot l \times 1)$, $i = 1, \ldots, N$. Thus, the set of $N$ pictures is represented by the following matrix:

$$\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N], \dim \boldsymbol{x}_i = k \cdot l = 2^q, \quad q = 3, 4, \ldots, . \tag{19}$$

The set of distorted images is represented by the matrix:

$$\boldsymbol{X}^{(s)} = \left[\boldsymbol{x}_1^{(s)}, \boldsymbol{x}_2^{(s)}, \ldots, \boldsymbol{x}_N^{(s)}\right]. \tag{20}$$

We notice that the training set can be captured as follows:

$$S = \left\{\boldsymbol{x}_i^{(s)}, \boldsymbol{x}_i\right\}_{i=1}^{N} \tag{21}$$

The set $S$ creates a mapping $F(\cdot)$ defined by the following properties:

$$\boldsymbol{x}_i = F\left(\boldsymbol{x}_i^{(s)}\right), \quad i = 1, \ldots, N. \tag{22}$$

In this way, the mapping $F$ is implemented as a machine learning system for image reconstruction. The structure realizing the mapping $F(\cdot)$, defined by Equation (22), can be obtained as solutions to the equilibrium state (9). Thus, for Equation (21), we obtain:

$$(\boldsymbol{W} - 2 \cdot \boldsymbol{1} + \boldsymbol{W}_s)\, \boldsymbol{m}_i + \boldsymbol{x}_i^{(s)} = \boldsymbol{0} \tag{23}$$

where $\boldsymbol{W}^2 = -\boldsymbol{1}$, $\boldsymbol{W}$ is an antisymmetric orthogonal matrix.

Thus, we obtain $N$ solutions:

$$\boldsymbol{m}_i = (2 \cdot \boldsymbol{1} - \boldsymbol{W}_s - \boldsymbol{W})^{-1} \boldsymbol{x}_i^{(s)}, \quad i = 1, \ldots, N \tag{24}$$

where

$$\begin{aligned} \boldsymbol{W}_s &= \boldsymbol{M}\left(\boldsymbol{M}^T \boldsymbol{M}\right)^{-1} \boldsymbol{M}^T \\ \boldsymbol{M} &= \{\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_N\} \end{aligned} \tag{25}$$

is the spectrum matrix of vectors $\boldsymbol{x}_i$:

$$\boldsymbol{m}_i = \frac{1}{2}(\boldsymbol{W}+\boldsymbol{1})\,\boldsymbol{x}_i$$

so

$$\boldsymbol{x}_i = (-\boldsymbol{W}+\boldsymbol{1})\,\boldsymbol{m}_i, \quad i = 1,\ldots,N. \tag{26}$$

Vectors $\boldsymbol{x}_i$ are the system vectors, i.e., $\boldsymbol{u}_i \equiv \boldsymbol{x}_i$, $i = 1,\ldots N$ ($\boldsymbol{x}_i$ are undistorted images).

Equation (24) defines the biorthogonal transformation $T_s(\cdot)$ :

$$\begin{aligned} \boldsymbol{m}_i &= T_s\left(\boldsymbol{x}_i^{(s)}\right) \\ T_s(\cdot) &= (2\cdot\boldsymbol{1}-\boldsymbol{W}_s-\boldsymbol{W})^{-1}(\cdot) \end{aligned} \tag{27}$$

and Equation (26) defines the orthogonal transformation:

$$\begin{aligned} \boldsymbol{x}_i &= T^{-1}(\boldsymbol{m}_i) \\ \boldsymbol{x}_i &= T^{-1}(\cdot) = (-\boldsymbol{W}+\boldsymbol{1})(\cdot). \end{aligned} \tag{28}$$

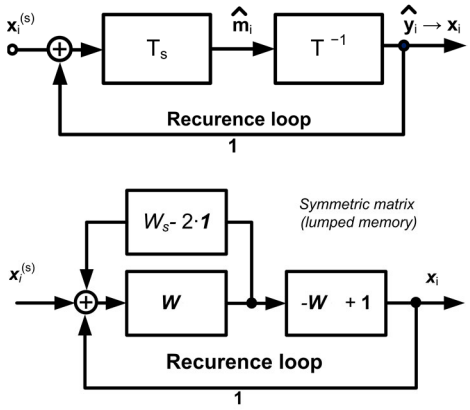



**FIGURE 3.** Structure of a machine learning system for image reconstruction.

The transformations $T_s(\cdot)$ and $T^{-1}(\cdot)$, which implement the mapping $F(\cdot)$, have the block structure shown in Fig. 3 [11]. The orthogonal transformation $T_s(\cdot)$, which utilizes the family of Hurwitz-Radon matrices, allows the determination of the Haar-Fourier spectra of vectors $\boldsymbol{x}_i$. The structure depicted in Fig. 3 functions as an estimator of the spectrum $\{\hat{m}_i\}$ using Equation (27):

$$\hat{\boldsymbol{m}}_i = T_s\left(\boldsymbol{x}_i^{(s)}\right), \quad i = 1,\ldots,N. \tag{29}$$

The feedback loop within the system leads to the convergence of the vectors as shown in Equations (30) and (31):

$$\hat{\boldsymbol{m}}_i \to \boldsymbol{m}_i \tag{30}$$

$$\hat{\boldsymbol{y}}_i \to \boldsymbol{x}_i, \quad i = 1,\ldots,N. \tag{31}$$

The convergence of the process is obtained after $L$ iterations, with the number $L$ varying for different reconstruction processes. Moreover, it should be noted that for input image $\boldsymbol{z} \neq \boldsymbol{x}_i, i = 1,\ldots,N$, the system output is given by the superposition of the system vectors, represented by Equation (32).

$$F(\boldsymbol{z}) = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i,\ \alpha_i \in R \tag{32}$$

The system vectors $\boldsymbol{u}_i = \boldsymbol{x}_i$ create attraction centers in the system, as shown in Fig. 3. This results in the implementation of the associative memory. We observe that the image recognition and reconstruction system presented in Fig. 3 can be defined as an associative memory. The original images are retrieved by utilizing distorted key images that are subject to reconstruction. Fig. 4 illustrates this reconstruction process through face recognition with masks.

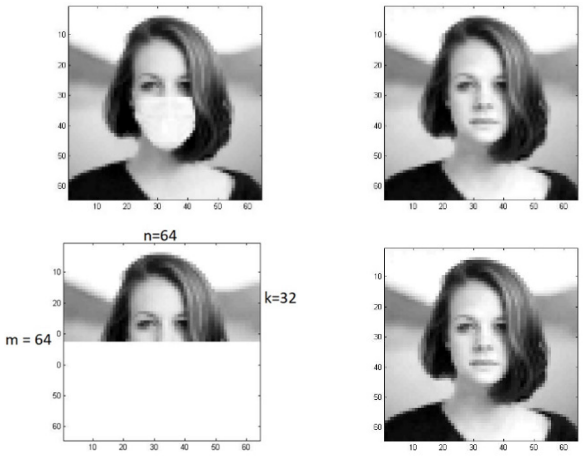



**FIGURE 4.** Illustration of face recognition by the system functioning as an associative memory.

## IV. IMAGE RECOGNITION AND RECONSTRUCTION AS INVERSE PROBLEM

The image reconstruction models presented in the previous section are based on the availability of training sets, $S$, containing the original and distorted patterns. Alternatively, the overall image reconstruction model can be written as shown in Equation (33).

$$A(\boldsymbol{x}) = \tilde{\boldsymbol{y}};\ \boldsymbol{A}\boldsymbol{x} = \tilde{\boldsymbol{y}} \tag{33}$$

where $A(\cdot)$—Known linear operator (e.g., matrix $\boldsymbol{A}$)

$\boldsymbol{x}$— Original image

$\tilde{\boldsymbol{y}}$— Degenerate image observed.

Without limiting the generality of our considerations, we assume that the linear operation $A(\cdot)$ takes the form of matrix multiplications. According to Equation (33), image reconstruction leads to the solution of the inverse problem. The majority of the solutions to Equation (33) that are known from the literature utilize optimization methods [5], [14], such as:

$$\min_x \|\tilde{\boldsymbol{y}} - \boldsymbol{A}\boldsymbol{x}\|_2^2, \quad \text{s. t.}\, \boldsymbol{x} \in K \tag{34}$$

$$\min_x \|\tilde{\boldsymbol{y}} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \beta R(\boldsymbol{x}) \tag{35}$$

where $K$— A set of feasible solutions
$R(\boldsymbol{x})$ − Regularizer
$\beta$ — Regularization parameter

As mentioned previously, different types of neural networks are now used to solve inverse problems in imaging, including image reconstruction. Many approaches to this problem can be found in review papers [14], [15]. Using the machine learning model shown in Fig. 3 to solve Equation (33) leads to the solution to the following problem:

$$F(\boldsymbol{x}) : \boldsymbol{A}\boldsymbol{x} = \tilde{\boldsymbol{y}}$$
$$\boldsymbol{x} = \boldsymbol{F}^{-1}(\tilde{\boldsymbol{y}}) \tag{36}$$

where $\boldsymbol{A} - (m \times n)$ Known real matrix, $m \neq n$
$\tilde{\boldsymbol{y}} - (m \times 1)$ Real vector
$\boldsymbol{x} - (n \times 1)$ Real vector
$m + n = 2^q,\ q = 3, 4, \ldots$.

The generation of the training set $S = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N}$ for Equation (33) is given by the formula

$$\boldsymbol{A}\boldsymbol{x}_i = \boldsymbol{y}_i, \quad i = 1, 2, \ldots, N \tag{37}$$

where $\boldsymbol{x}_i, i = 1, 2, \ldots, N$ are the vector forms of the original training images. Assuming that the projection matrix $\boldsymbol{A}\ (m \times n)$ in Equation (36) is a random matrix, the images $\boldsymbol{y}_i$ in the training set become random vectors. An exemplary image and its projection are shown in Fig. 5.

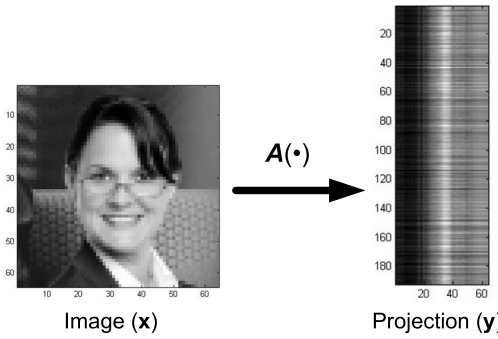



**FIGURE 5.** The original image and its projection.

As shown in Fig. 5, the vector transformation of the selected test image takes the form:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \tag{38}$$

where $\dim \boldsymbol{A} = (m \times n),\ m > n$.

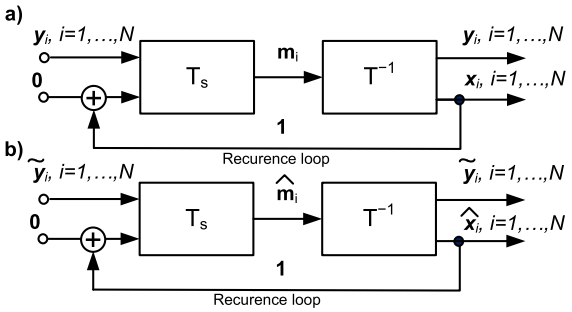
By creating system vectors $\boldsymbol{u}_i$ of the form

$$\boldsymbol{u}_i = \begin{bmatrix} \boldsymbol{y}_i \\ \boldsymbol{x}_i \end{bmatrix}, \quad i = 1, \ldots, N \tag{39}$$

the structure of the inverse mapping in Equation (36) implemented by the system in Fig. 3 is presented in Fig. 6(a) and 6(b), where

$$\boldsymbol{x}_i = F^{-1}(\boldsymbol{y}_i), \quad i = 1, \ldots, N. \tag{40}$$

Note that the biorthogonal transformation $T_s(\cdot)$ and the orthogonal transformation $T^{-1}(\cdot)$ in Fig. 6 are defined by




$\hat{\boldsymbol{x}}_i$— Image estimator of $\boldsymbol{x}_i$
$\tilde{\boldsymbol{y}}_i$— Projection of the degenerate image $\boldsymbol{x}_i$

**FIGURE 6.** The structure of the system implementing the inverse transformation $F^{-1}(\cdot)$. (a) $y_i$ – projection of a non-degenerate image. (b) $\tilde{y}$ – projection of a degenerate image.

Equations (27) and (28), respectively:

$$\boldsymbol{m}_i = T_s\left(\begin{bmatrix} \boldsymbol{y}_i \\ \boldsymbol{0} \end{bmatrix}\right) \tag{41}$$
$$\boldsymbol{u}_i = T^{-1}(\boldsymbol{m}_i) \tag{42}$$

where $\boldsymbol{u}_i$— system vectors in Equation (39).

In the system depicted in Fig. 6, distorted projections of images $\tilde{\boldsymbol{y}}_i, i = 1, \ldots, N$ are reconstructed. To illustrate the properties of the reconstruction system shown in Fig. 6, the training set $S$ was generated using Equation (37) with images $\boldsymbol{x}_i, i = 1, \ldots, N$ constituted the set. Fig. 7 shows an example of image transformation. In the system shown in Fig. 6(b), we obtain:

$$\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2 = 0 \tag{43}$$

where $\boldsymbol{x}$ is given by Equation (38).

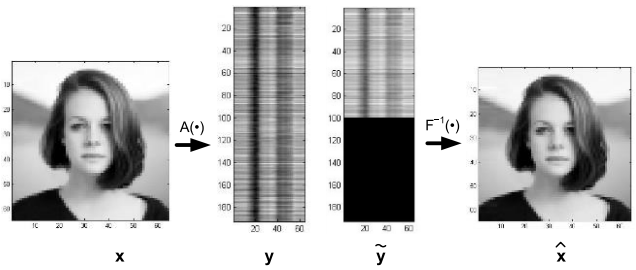



**FIGURE 7.** Exemplary reconstruction ($F^{-1}(\cdot)$ – by the system from Fig. 6(b) .

To recap, Fig. 3 and Fig. 6 depict image reconstruction systems that utilize associative memory to recognize degenerative images. However, it is worth noting that the system in Fig. 6 performs the inverse mapping and solves the optimization tasks constrained by the images stored in the system's memory. Regarding the convergence of the recursive sequence, it should be noted that it is valid for $N < \frac{1}{2}(n+m)$.

The system presented in Fig. 6 also allows us to solve the linear Equation (36) using a random form of training vectors $\boldsymbol{x}_i$ in Equation (37). Thus, the new structure of the machine learning system for solving the inverse problem in Equation (36) is obtained by generating the training set using

random vectors $\boldsymbol{x}_i$ in Equation (37). So $\boldsymbol{x}_i, i = 1, \ldots, N$ are random forms of training vectors (images), where

case $m > n$

$$N = m, \dim \boldsymbol{A} = (m \times n), \quad m > n \tag{44}$$

and $\boldsymbol{y}_i, i = 1, \ldots, N$ are the projections of random images $\boldsymbol{x}_i$. The system's structure is given by Equations (7)–(12), where the symmetric matrix $\boldsymbol{W}_s$ in Equation (10) is subject to regularization using the regularization rule for the pseudo-inverse matrix:

$$\boldsymbol{W}_s = \boldsymbol{M}\left(\boldsymbol{M}^T\boldsymbol{M} + \gamma \boldsymbol{1}\right)^{-1}\boldsymbol{M}^T \tag{45}$$

where $\gamma \geq 0$ (regularization parameter).

The system vectors $\boldsymbol{u}_i$ have the following form:

$$\boldsymbol{u}_i = \begin{bmatrix} \boldsymbol{y}_i \\ \boldsymbol{x}_i \end{bmatrix}, \; i = 1, \ldots, N, N = m, dim\boldsymbol{u}_i = m + n \tag{46}$$

Therefore, the following statement can be made:

Statement 1

Each image $\boldsymbol{I}\dim(n \times 1)$ can be reconstructed in the system from Fig. 6 under the conditions $\boldsymbol{I}_p \neq \boldsymbol{0}, \boldsymbol{A}(\boldsymbol{I}_1) \neq \boldsymbol{A}(\boldsymbol{I}_2)$, for $\boldsymbol{I}_1 \neq \boldsymbol{I}_2$, when its projection $\boldsymbol{I}_p$ is known:

$$\boldsymbol{I}_p = \boldsymbol{A}\boldsymbol{I}. \tag{47}$$

The quality of the reconstructed image, represented by $\tilde{\boldsymbol{I}}$ and measured by the MSE, depends on the selection of the parameter $\gamma$ in Equation (45). An exemplary image reconstruction as a solution to the inverse problem is shown in Fig. 8.

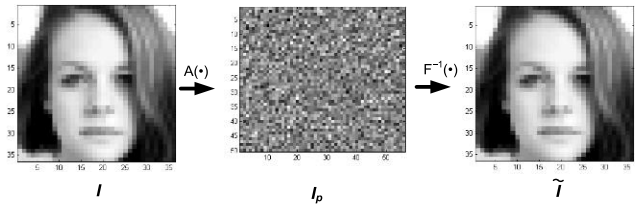



**FIGURE 8.** Image reconstruction in a system with randomly generated system vectors ($\gamma = 0.1$).

The computational experiment presented in Fig. 8 can be described as follows:

The system synthesis:

- Random image vectors $\boldsymbol{x}_i, i = 1, \ldots, 2800$, $\dim \boldsymbol{x}_i = 1296$ were generated.
- A random projection matrix $\boldsymbol{A}$ was generated, $\dim \boldsymbol{A} = (2800 \times 1296)$, where the components of matrix $\boldsymbol{A}$ are random and follow a normal distribution.
- Projections $\boldsymbol{y}_i$ of vectors $\boldsymbol{x}_i$ were determined:

$$\boldsymbol{y}_i = \boldsymbol{A}\boldsymbol{x}_i, i = 1, \ldots, 2800, \dim \boldsymbol{y}_i = 2800.$$

- System vectors $\boldsymbol{u}_i$ were determined by the vectors' concatenation $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$:

$$\boldsymbol{u}_i = \begin{bmatrix} \boldsymbol{y}_i \\ \boldsymbol{x}_i \end{bmatrix}, i = 1, \ldots, 2800, \dim \boldsymbol{u}_i = 4096.$$

- The system in Fig. 6 was synthesized according to the procedure outlined in Equations (23)–(28).

Image reconstruction

The image was $36 \times 36$ pixels and analyzed and vectorized into the vector $\boldsymbol{I}$, $\dim \boldsymbol{I} = 1296$. The resulting projection was obtained according to the formula in Equation (47):

$$\boldsymbol{I}_p = \boldsymbol{A}\boldsymbol{I}, \quad \dim \boldsymbol{I}_p = 2800.$$

For the purposes of the presentation, the vector $\boldsymbol{I}_p$ is shown in the form of a grayscale matrix ($50 \times 56$ pixels). The system was given the vector projection $\boldsymbol{I}_p$ of the analyzed image as input and then recursively reconstructed the vector $\tilde{\boldsymbol{I}}$ assigned to the original image. The image shown in Fig. 8 was created in the reverse process of concatenation from the vector $\tilde{\boldsymbol{I}}$. The quality of the reconstruction was evaluated based on the mean square error $MSE\left(\boldsymbol{I}, \tilde{\boldsymbol{I}}\right)$ between the original image and the replica obtained in the system. The values of the mean square error depending on the value of the regularization parameter $\gamma$ in Equation (45) are presented in Table 1.

**TABLE 1.** Dependence of the mean square error of the image reconstruction on the value of the regularization parameter.

| Regularization parameter $\gamma$ | $MSE\left(\boldsymbol{I}, \tilde{\boldsymbol{I}}\right)$ |
|---|---|
| 0.5 | $4.7 \cdot 10^{-10}$ |
| 0.1 | $6.2 \cdot 10^{-8}$ |
| 0.01 | $1.8 \cdot 10^{-4}$ |
| 0.005 | 0.003 |
| 0.003 | 0.022 |
| 0.002 | 0.154 |
| $> 0.001$ | Convergent process starts |

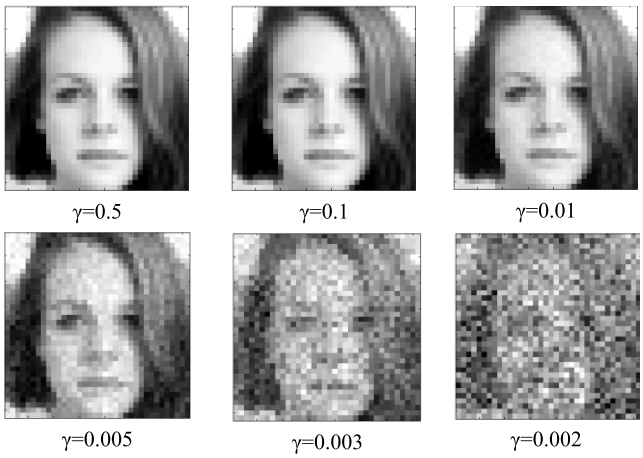



**FIGURE 9.** Influence of the regularization parameter $\gamma$ on the image reconstruction quality.

Fig. 9 illustrates how the parameter $\gamma$ affects the quality of image reconstruction in the machine learning system. We observed a specific limit value of the regularization parameter, which the reconstruction process diverges.

Note that one can state: $\tilde{\boldsymbol{I}} \cong \boldsymbol{I}$ when $\text{MSE}\left(\tilde{\boldsymbol{I}}, \boldsymbol{I}\right) < 1.8 \cdot 10^{-4}$ (data from Table 1).

Statement 2

The relationship in Equation (47) obtained in the image/vector inverse problem can be extended to the problem of realizing an unknown mapping, $F(\cdot)$. Assuming that the training set $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}$ is generated by such a linear mapping $F(\cdot)$, that is:

$$\boldsymbol{y}_i = F(\boldsymbol{x}_i), \quad i = 1, 2, \ldots, N \tag{48}$$

where $\boldsymbol{y}_i \in R^m, \boldsymbol{x}_i \in R^n, m = N, n < m$.

The system in Fig. 6, defined in Statement 1, models the mapping $F(\cdot)$ and its inverse $F^{-1}(\cdot)$. It is important to note that the system $F(\cdot)$ implemented in accordance with Equation (48) also accurately maps the vectors $\boldsymbol{x}_p$ that do not belong to the training set. In other words, the following condition is satisfied:

$$\boldsymbol{y}_p = F(\boldsymbol{x}_p), p \notin \{1, 2, \ldots, N\} \tag{49}$$

and

$$\boldsymbol{x}_p = F^{-1}(\boldsymbol{y}_p); (\boldsymbol{x}_p, \boldsymbol{y}_p) \notin \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N. \tag{50}$$

The structure of the system that realizes the mapping $F(\cdot)$ is based on system vectors $\boldsymbol{u}_i, i = 1, 2, \ldots N$ shown in Equation (46).

It is worth noting that the system presented in Fig. 6 can solve Equation (36) using a random form of training vectors $\boldsymbol{x}_i$ in Equation (37), even when $m < n$, so:

case $m < n$

$$\dim \boldsymbol{A} = (m \times n), m < n. \tag{51}$$

Hence, the random training vectors $\boldsymbol{x}_i; i = 1, 2, \ldots, N$ and $N = n$.

Thus, the system vectors $\boldsymbol{u}_i$ have the form:

$$\boldsymbol{u}_i = \begin{bmatrix} \boldsymbol{y}_i \\ \boldsymbol{x}_i \end{bmatrix}, \quad i = 1, \ldots, N, N = n, \dim \boldsymbol{u}_i = m + n \tag{52}$$

where $\boldsymbol{y}_i$ are the projections of random images/vectors $\boldsymbol{x}_i$.

It should be noted that the inverse mapping $F^{-1}(\cdot)$ in Equation (36), implemented by the system from Fig. 6, depends on the form of the training set.

Thus, the solution to the inverse problem determined by Equation (36) sets up some solutions. However, it is clear that the quality measured by the MSE of such solutions depends on the regularization parameter $\gamma$.

However, the computational experiment presented in Fig. 8 can be repeated for case $m < n$, as follows:

- One creates two sets of random vectors $\boldsymbol{x}_i$, i.e.,

$$S_1 = \left\{\boldsymbol{x}_i^{(1)}, \boldsymbol{y}_i^{(1)}\right\}_{i=1}^N, \; S_2 = \left\{\boldsymbol{x}_i^{(2)}, \boldsymbol{y}_i^{(2)}\right\}_{i=1}^N$$

  where $\boldsymbol{y}_i^{(1)} = \boldsymbol{A}\boldsymbol{x}_i^{(1)}, \boldsymbol{y}_i^{(2)} = \boldsymbol{A}\boldsymbol{x}_i^{(2)}$, $\dim \boldsymbol{A} = (m \times n), m = 1296, n = 2800, N = n$.
- Two systems based on Fig. 6 were used to solve the inverse problem given by the equation

$$\boldsymbol{I}_p = \boldsymbol{A}\boldsymbol{I}$$

where $\boldsymbol{I}$ – the test image from Fig. 8

$\boldsymbol{I}_p$ – the projection.

The systems were trained using the training sets $S_1$ and $S_2$, respectively.
- The image reconstruction results are shown in Fig. 10.

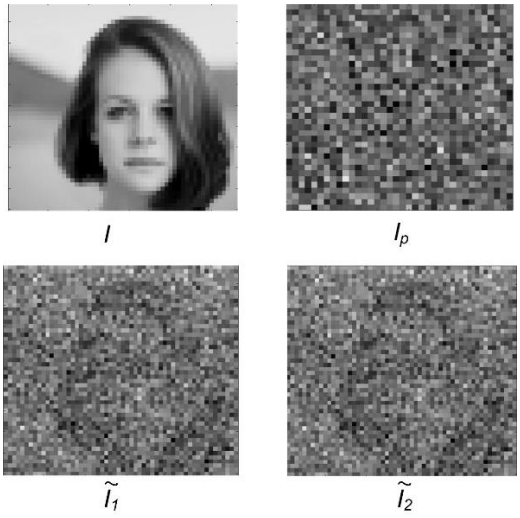



**FIGURE 10.** Results of the computational experiment (Matlab software).

The images $\tilde{\boldsymbol{I}}_1$ and $\tilde{\boldsymbol{I}}_2$ are the outputs of systems based on different sets $S_1$ and $S_2$, respectively. It is clear that $\boldsymbol{I}$ is not equal to $\tilde{\boldsymbol{I}}_1$, and $\tilde{\boldsymbol{I}}_2$, but the following equations are valid:

$$\boldsymbol{A}\boldsymbol{I} = \boldsymbol{I}_p$$

and

$$\boldsymbol{A}\tilde{\boldsymbol{I}}_1 - \boldsymbol{I}_p \cong \boldsymbol{0}; \quad \text{MSE}\left(\boldsymbol{A}\tilde{\boldsymbol{I}}_1, \boldsymbol{I}_p\right) \cong 0$$
$$\boldsymbol{A}\tilde{\boldsymbol{I}}_2 - \boldsymbol{I}_p \cong \boldsymbol{0}; \quad \text{MSE}\left(\boldsymbol{A}\tilde{\boldsymbol{I}}_2, \boldsymbol{I}_p\right) \cong 0$$

but

$$\tilde{\boldsymbol{I}}_1 \cong \tilde{\boldsymbol{I}}_2; \quad \text{MSE}\left(\tilde{\boldsymbol{I}}_1, \tilde{\boldsymbol{I}}_2\right) = 4.87 \cdot 10^{-10}.$$

To summarize, the experiments described lead to the conclusion that it is possible to design a machine learning system in which the solution of an inverse problem does not depend on the specific training set used. But the solutions (i.e., $\tilde{\boldsymbol{I}}_1$, and $\tilde{\boldsymbol{I}}_2$) are only optimal in the sense of LS.

On the other hand, many machine learning algorithms and applications rely on solutions of linear equations of the form:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{53}$$

where $\boldsymbol{A} - (m \times n)$ real matrix, $m < n$.

To solve this equation using the structure from Fig. 6, one can generate a set of $k$ training sets $S_k = \{\boldsymbol{x}_i, \boldsymbol{b}_i\}_{i=1}^N$, $k = 1, 2, \ldots$

$$\boldsymbol{A}\boldsymbol{x}_i = \boldsymbol{b}_i; i = 1, \ldots, N \text{ and } N = m. \tag{54}$$

Note that in Equation (51), one assumes $N = n$.

The structure of the mapping model presented in Fig. 6 can deliver any number of exact solutions for the equation $\boldsymbol{A}\boldsymbol{x}_k = \boldsymbol{b}$ by using different sets $S_k$.

$$S_k = \{\boldsymbol{x}_i, \boldsymbol{b}_i\}_{i=1}^{N}, \quad k = 1, 2, \ldots$$

where $N = m < \frac{1}{2}(m+n)$, $n+m = 2^q$, $q = 3, 4, \ldots$

Note 2

It can be easily observed that a pure algebraic operation can also obtain the solution to linear Equation (53). Indeed, the possible solutions are given by the following formula:

$$\tilde{\boldsymbol{x}}_k = \boldsymbol{X}_k \boldsymbol{B}_k^{-1} \boldsymbol{b}, \quad k = 1, 2, \ldots \tag{55}$$

where $\tilde{\boldsymbol{x}}_k$ – $k$th solution of linear Equation (53)

$\boldsymbol{X}_k = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m]$ $(m \times n)$ – matrix of training vectors from $k$th training set

$\boldsymbol{B}_k = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_m]$ $(m \times m)$ – quadratic matrix of vectors in Equation (54)

Assuming that matrix $\boldsymbol{A}$ consists of $n$ linearly independent columns, matrix $\boldsymbol{B}_k^{-1}$ exists for $k = 1, 2, \ldots$.

Hence, for matrix $\boldsymbol{A}$:

$$\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n] \, (m \times n). \tag{56}$$

The elements of vectors $\tilde{\boldsymbol{x}}_k$ can be treated as superposition coefficients, i.e.:

$$\tilde{\boldsymbol{x}}_{k1}\boldsymbol{a}_1 + \tilde{\boldsymbol{x}}_{k2}\boldsymbol{a}_2 + \ldots + \tilde{\boldsymbol{x}}_{kn}\boldsymbol{a}_n = \boldsymbol{b}, \; k = 1, 2, \ldots. \tag{57}$$

Thus, the inverse problem solutions formulated by Equation (53) could give rise to some form of wavelets analysis when $\boldsymbol{A}$ is a set of wavelets $\{\boldsymbol{a}_i\}_{i=1}^{n}$ and $\boldsymbol{b}$ is a given analyzed signal. Audio unmixing could be mentioned as an example of such analysis [16].

## V. Q-INSPIRED INVERSE PROBLEM

Q-inspired neural networks are non-quantum versions of structures with complex-valued parameters. Equation (2) gives rise to Q-inspired machine learning when the $\boldsymbol{W}_s$ matrix becomes complex-valued. Thus, Equation (23) is transformed into the form:

$$(\boldsymbol{W} - 2 \cdot \boldsymbol{1} + \boldsymbol{W}_H)\, \boldsymbol{m_i} + \boldsymbol{I}_d = \boldsymbol{0} \tag{58}$$

where $\boldsymbol{W}_H$ – Hermitian matrix $(\boldsymbol{W}_H = \boldsymbol{W}_H^{+})$

$\boldsymbol{I}_d$ – Input vector.

Hence, the inverse problem formulated by Equation (36), that is:

$$\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{y} \tag{59}$$

where $\boldsymbol{A}$ – complex-valued matrix $(m \times n)$

$\boldsymbol{x}$ – a complex solution of inverse problem $(\boldsymbol{x} \in C)$

$\boldsymbol{y}$ – a given complex vector $(n \times 1)$.

Equation (59) can be solved by using the system from Fig. 3 if the vectors of training set $S = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N}$ are complex. For complex-valued image processing, the machine learning system structure is presented in Fig. 11.

To illustrate the solution of the inverse problem formulated in Equation (59), two images processed by the system from Fig. 11 are presented in Fig. 12. It is clear that the two images $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ are vectorized into one complex-valued vector:

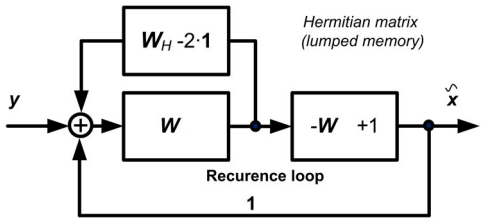



**FIGURE 11.** Machine learning system with Hermitian memory.

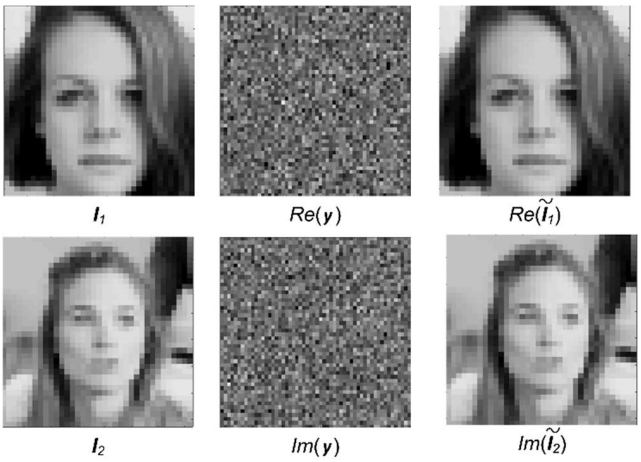



**FIGURE 12.** Results of the computational experiments (Eq. 59).

$$\boldsymbol{x} = \boldsymbol{I}_1 + j\boldsymbol{I}_2; \; \dim \boldsymbol{x} = 1296, j^2 = -1. \tag{60}$$

The projection $\boldsymbol{y}$ of $\boldsymbol{x}$ by the complex-valued matrix $\boldsymbol{A}$ $(2800 \times 1296)$; $a_{ij} \in C$ is shown in the vectorized form as:

$$\boldsymbol{y} = Re(\boldsymbol{y}) + jIm(\boldsymbol{y}) = \boldsymbol{A}(\boldsymbol{x}). \tag{61}$$

The solution to Equation (59) is illustrated in Fig. 12:

$$\tilde{\boldsymbol{x}} = \boldsymbol{A}^{-1}(\boldsymbol{y}) \tag{62}$$

and MSE $(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \cong 0$.

Statement 3

The solution to the inverse problem formulated in Equation (59) can be obtained by using the methods formulated in Statements 1 and 2 for the real-valued problem.

## VI. SOME REMARKS ON THE DEEP LEARNING STRUCTURES

It is worth noting that deep learning neural networks, such as multilayer perceptron, CNNs, RNNs, and autoencoders, can be seen as special types of nonlinear filters endowed with memories that are created by optimization procedures (e.g., back-propagation, gradient descent). The effectiveness of these networks depends on their specific tasks. For example, CNNs are popular as computer-vision processing tools. Contrary to the previously mentioned deep neural networks, the machine learning system presented in this paper can be viewed as a linear filter processing the spectra of the training

sets' data. Therefore, the learning can be implemented by a synthesis-like procedure. It should be clear that this system obtains two features, as identified in the text:

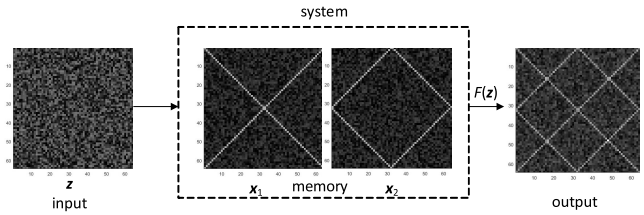



**FIGURE 13. Illustration (proof of the principles) of an image generated by the superposition.**

- It is associative memory retrieved by the degenerated keys.
- It is a mapping approximator based on superposition of system vectors (memory vectors).

This superposition feature gives rise to image generation. To illustrate such a feature, two images presented in Fig. 13 are combined in the system from Fig. 3, according to Equation (32):

$$F(z) = x_1 + x_2 \tag{63}$$

where $z-$ A noise input vector/image
$F(z) -$ An output image
$x_1, x_2 -$System vectors (memory vectors/images)

To summarize, in this paper, we proposed a machine learning model that uses biorthogonal and orthogonal transformations based on spectral processing as alternative solutions to deep learning based on optimization procedures.

## VII. CONCLUSION

This paper aims to present a machine learning model to solve inverse problems arising from linear mappings commonly used in image processing. Two types of inverse problems are considered and classified. First, the machine learning system is used as an associative memory to perform inverse transformation and solve optimization tasks constrained by the images stored in the system. Second, the model solves inverse problems by generating random training sets with particular interest given to those consisting of both real and complex-valued vectors.

## REFERENCES

[1] S. K. Pal, A. Ghosh, and M. K. Kundu, *Soft Computing for Image Processing* (Studies in Fuzziness and Soft Computing). New York, NY, USA: Physica-Verlang, 2000.
[2] Z. Huang, S. Ye, M. T. McCann, and S. Ravishankar, "Model-based reconstruction with learning: From unsupervised to supervised and beyond," 2021, *arXiv:2103.14528*.
[3] S. S. Kaderuppan, E. W. L. Wong, A. Sharma, and W. L. Woo, "Smart nanoscopy: A review of computational approaches to achieve super-resolved optical microscopy," *IEEE Access*, vol. 8, pp. 214801–214831, 2020.
[4] S. Ramanarayanan, B. Murugesan, K. Ram, and M. Sivaprakasam, "DC-WCNN: A deep cascade of wavelet based convolutional neural networks for MR image reconstruction," in *Proc. IEEE 17th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2020, pp. 1069–1073.
[5] J. A. Fessler, "Optimization methods for magnetic resonance image reconstruction: Key models and optimization algorithms," *IEEE Signal Process. Mag.*, vol. 37, no. 1, pp. 33–40, Jan. 2020.
[6] H. Zheng, S. W. A. Sherazi, S. H. Son, and J. Y. Lee, "A deep convolutional neural network-based multi-class image classification for automatic wafer map failure recognition in semiconductor manufacturing," *Appl. Sci.*, vol. 11, no. 20, p. 9769, Oct. 2021.
[7] S. K. Zhou, H. Greenspan, C. Davatzikos, J. S. Duncan, B. Van Ginneken, A. Madabhushi, J. L. Prince, D. Rueckert, and R. M. Summers, "A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises," *Proc. IEEE*, vol. 109, no. 5, pp. 820–838, May 2021.
[8] T. M. Quan, T. Nguyen-Duc, and W.-K. Jeong, "Compressed sensing MRI reconstruction using a generative adversarial network with a cyclic loss," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1488–1497, Jun. 2018.
[9] M. Mardani, E. Gong, J. Y. Cheng, S. S. Vasanawala, G. Zaharchuk, L. Xing, and J. M. Pauly, "Deep generative adversarial neural networks for compressive sensing MRI," *IEEE Trans. Med. Imag.*, vol. 38, no. 1, pp. 167–179, Jan. 2019.
[10] D. Liang, J. Cheng, Z. Ke, and L. Ying, "Deep magnetic resonance image reconstruction: Inverse problems meet neural networks," *IEEE Signal Process. Mag.*, vol. 37, no. 1, pp. 141–151, Jan. 2020.
[11] W. Citko and W. Sienko, "Hamiltonian and Q-inspired neural network-based machine learning," *IEEE Access*, vol. 8, pp. 220437–220449, 2020.
[12] W. Citko and W. Sienko, "Inpainted image reconstruction using an extended Hopfield neural network based machine learning system," *Sensors*, vol. 22, p. 813, Jan. 2022.
[13] D. Gilton, G. Ongie, and R. Willett, "Deep equilibrium architectures for inverse problems in imaging," 2021, *arXiv:2102.07944*.
[14] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb, "Solving inverse problems using data-driven models," *Acta Numerica*, vol. 28, pp. 1–174, May 2019.
[15] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, "Deep learning techniques for inverse problems in imaging," 2020, *arXiv:205.06001*.
[16] Q. Li, Y. Ding, and J. Olan, "How audio is getting its groove back: Deep learning is delivering the century-old promise of truly realistic sound reproduction," *IEEE Spectr.*, vol. 59, no. 10, pp. 46–52, Oct. 2022.

**WIESLAW CITKO** received the M.S. degree in solid state physics from the Faculty of Applied Physics and Mathematics, Gdańsk University of Technology, and the Ph.D. degree in electronics from the Lodz University of Technology. He is currently an Assistant Professor with the Department of Electrical Engineering, Gdynia Maritime University, Gdynia, Poland. His current research interests include artificial intelligence, machine learning, neural networks, and image processing.

**WIESLAW SIENKO** (Life Member, IEEE) received the M.S. and Ph.D. degrees in electronics from the Gdańsk University of Technology. He is currently a Professor with the Department of Electrical Engineering, Gdynia Maritime University, Gdynia, Poland. His current research interests include artificial intelligence, machine learning, quantum signal processing, and digital signal processing.