UDC 004.032.26::004.93

V.V. Romanuke

Khmelnitskiy National University, Khmelnitskiy

TWO-LAYER PERCEPTRON FOR CLASSIFYING SCALED-TURNED-SHIFTED OBJECTS BY 26 CLASSES GENERAL TOTALITY OF MONOCHROME 60-BY-80-IMAGES VIA TRAINING WITH PIXEL-DISTORTED SCALED-TURNED-SHIFTED IMAGES

There is being tried the two-layer perceptron for its identification in classifying diversely distorted objects. The two-layer perceptron is modeled, trained and tested within MATLAB. Having increased the passes number up to 280 for training over scaled-turned-shifted images with pixel distortion, the perceptron performance is improved as for pure scaled-turned-shifted images, as well as for pixel-distorted those ones, wherein pixel-distorted images are classified excellently anyway. Eventually, as neocognitron on the same MATLAB and operating system configuration is many times slower, two-layer perceptron is asserted to be capable to substitute neocognitron in classifying diversely distorted objects, needing peculiarly longer training process and specific ratio for types of distortion, what depends on the object type.

Keywords: *classification of diversely distorted objects, neocognitron, two-layer perceptron, monochrome image, pixel-distortion, training set, classification error percentage.*

Problems in classifying diversely distorted objects

Nowadays the applied computer science is developed for intensifying automatization and automatic monitoring, check, and control. Object classification is a great part of this global problem. In mathematical terms and statements, the object classification problem and its solution methods are formulated strictly. Nonetheless the elegancy of math in it, the classification of diversely distorted objects (DDO) is still a difficult task. This task is tried to be solved with neural nets, seeming to be universal classifiers due to a lot of theorems on algorithmic solvability and classification algorithm convergence [1, 2]. But despite the Rosenblatt theorem on perceptron convergence, it performs poorly over DDO. Neocognitron, being a hierarchical multilayered neural network [3, 4], performs perfectly over objects, distorted in diverse ways [5, 6]. Particularly, it is used for optical and handwritten character recognition [7, 8]. However, projection of neocognitron and hierarchical neural networks on its basis is a complicated process, as well as its training process through learning [9]. Another problem, standing before classification procedure, is segmentation of a field of those objects [10, 11] to be classified: the better segmentation is accomplished by a criterion, the easier neocognitron projection process and learning is running to be done.

Restrictions of neocognitron for distortion-imperceptible classification

At first blush, multilayer perceptron is not a competitor to neocognitron, because perceptron performance in classifying DDO cannot even be compared to high-accuracy neocognitron performance in that. However, operation speed of neocognitron is too low for declaring hierarchical neural networks on its basis to be ideal in distortion-imperceptible classification [12]. Besides, these networks require much RAM and hard disk space, what makes them be unwieldy so far [13, 14]. It concerns especially objects with greater number of features: while there are a few features in objects, neocognitron can be the normal operation speed classifier; and as just this number increases, neocognitron slows down distinctly [15]. Thus, those restrictions of neocognitron for distortion-imperceptible classification over objects with great number of features hint at that multilayer perceptron might be re-considered as a means for classifying DDO, needing only some special training process [16, 17]. Advantages of multilayer perceptron are its swiftness and low resources consumption. Only they come into force after perceptron has been successfully trained [17, 18]. Actually, the perceptron training process adjustment and optimization determine its classification capabilities [19]. And it would be worth to try to develop the perceptron training process for getting its maximal performance in classifying DDO. So below is a try to adjust it that perceptron could substitute neocognitron.

Article goal and tasks

The prime question is how many layers perceptron should have and what types of distortion are going to be tried. Without pretending to generalization, two layers are enough for perceptron, although there exists a hypothesis about that more layers may lead to reducing the total number of neurons. Types of distortion depend on what kind of object is considered. Whatever it is, there must be tried as many types of distortion, as it is sufficient for presenting the diversely distorted object. An image is a good model of the object, as image distortions or deformations are well-visible and comprehensi-

ble, not latent or implied mathematically. The image color is not necessary, and the object will be the monochrome image. For the monochrome image there are four types of its distortion: pixel inversion, scaling, turning (rotation or angulation), and shift. These types are stated in order of complexity to train the two-layer perceptron (TLP), where pixel inversion distortion is solved by any trained TLP, and the rest three types of distortion still stay unsolved for TLP.

This article goal is to obtain TLP classifier, whose performance is near-minimal classification error percentage (CEP) over DDO from some general totality of monochrome images of the same format. Images, running to be classified, may be scaled-turned-shifted in diverse ratios and ways. Firstly, the general totality will be defined. Then there will be modeled pixel-distorted monochrome images (PDMI), scaled-turned-shifted monochrome images (STSMI), and pixel-distorted STSMI (PDSTSMI). Secondly, it is needed to be checked once again that for the defined general totality that TLP, trained with PDMI only, cannot classify nor STSMI, nor PDSTSMI. Thirdly, there is TLP to be checked out whether it can be trained with STSMI. Perhaps, it will be a fail in it, and TLP is going to be trained with PDSTSMI, where part of STSMI within the training set is slight. Then, increasing the part of STSMI within the training set of PDSTSMI for TLP, will see whether the positive dynamics in classifying STSMI is maintained. If it is then TLP, trained with PDSTSMI, can be a fine classifier over DDO, needing peculiar long training process and specific ratio for types of distortion, though.

General totality and number of classes in it

The general totality of monochrome images should be taken for obtaining classification results both as fast as possible and adequate. Therefore the format of this general totality element must be selected properly. Also number of classes in the general totality must be not very great, lest it would linger the investigation processes, while training and testing. Getting started on this, there is selected format 60×80 for monochrome images, an appropriate format for further computations. A real monochrome image is the enlarged capital letter, thus making up 26 classes (fig. 1). The file format is bitmap, which is naturally coded with ones and zeros, and the announced general totality of 60×80 matrices of ones and zeros is constituted from 2^{4800} monochrome images.

Fig. 1. Non-distorted representatives of 26 classes as 60×80 monochrome images of 26 English alphabet capital letters (viewed as bitmap files)

Note that the letter image is not the continuous black cast on the continuous white background. There in fig. 1 can be seen regular crosshatching pixel black-into-white inversions. That means an interesting fact of that the initial pixel inversion distortion is already put into non-distorted representatives.

Models of PDMI, STSMI, PDSTSMI

The investigations are going to be carried out from MATLAB environment. In MATLAB the monochrome 60×80 image is presented as a 60×80 matrix, wherein the white color is coded with ones, and the black color is coded with zeros (fig. 2). Hence let $\mathbf{A}_q = \left(a_{uv}^{\langle q \rangle}\right)_{60\times 80}$ be the matrix of elements $a_{uv}^{\langle q \rangle} \in \{0, 1\}$ by $q = \overline{1, 26}$ that maps the q-th class non-distorted representative (fig. 1). Henceforward any class representative as a 60×80 matrix is reshaped into 4800-length-column.

For the training process with PDMI all the representatives as 4800-length-columns are concatenated horizontally. Thus the 26 non-distorted representatives are concatenated into the matrix $\mathbf{A} = \left(\overline{a}_{jq}\right)_{4800\times 26}$, where its q-th column is the column-reshaped representative of the q-th class. Model of PDMI is in forming the matrix $\mathbf{A}_{\text{PDMI}}^{\langle k \rangle} = \left(\overline{a}_{jq}^{\langle k\text{-PDMI} \rangle}\right)_{4800\times 26}$ of 26 pixel-distorted representatives (each for its class) through the addition

$$\mathbf{A}_{\text{PDMI}}^{\langle k \rangle} = \mathbf{A} + \sigma_{\text{pixel}}^{\langle k \rangle} \cdot \Xi \qquad (1)$$

by standard deviation (SD)

$$\sigma_{\text{pixel}}^{\langle k \rangle} = \frac{k}{F} \cdot \sigma_{\text{pixel}}^{\langle \max \rangle} \quad \forall\, k = \overline{1, F} \qquad (2)$$

and its maximum $\sigma_{\text{pixel}}^{\langle \max \rangle} > 0$ at 4800×26 matrix Ξ of values of normal variate (NV) with zero expectation and unit variance (ZEUV). In the assignment (2) there is calculated the pixel inversion SD on the k-th step of forming F matrices $\left\{\mathbf{A}_{\text{PDMI}}^{\langle k \rangle}\right\}_{k=1}^{F}$ with PDMI.

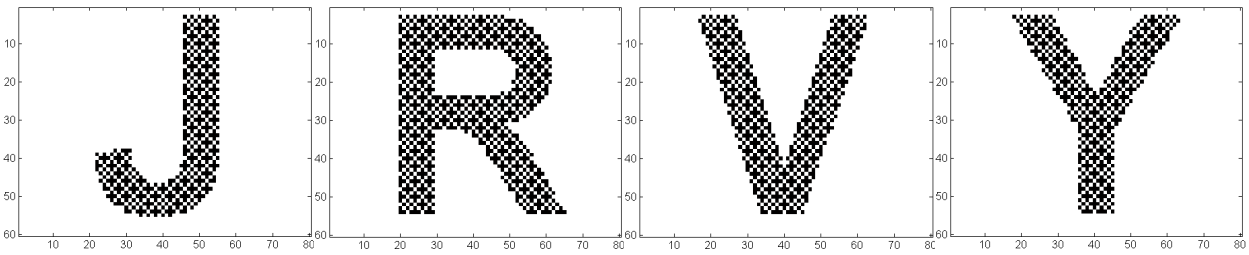

Fig. 2. Non-distorted 60 × 80 letters “J”, “R”, “V”, “Y”, viewed from within MATLAB

For accomplishing the training process there is available MATLAB training function “traingda” as one of the fastest training functions for TLP, trained with backpropagation algorithm [20, 21]. In the training process with PDMI the input of TLP is fed with the training set

$$\left\{\tilde{\mathbf{P}}_i^{\langle \mathrm{PDMI}\rangle}\right\}_{i=1}^{C+F}=\left\{\left\{\mathbf{A}\right\}_{l=1}^{C},\left\{\mathbf{A}_{\mathrm{PDMI}}^{\langle k\rangle}\right\}_{k=1}^{F}\right\} \tag{3}$$

of C replicas of matrix $\mathbf{A}$ of non-distorted representatives and F matrices of PDMI by the set of identifiers (targets)

$$\left\{\mathbf{T}_i\right\}_{i=1}^{C+F}=\left\{\mathbf{I}\right\}_{i=1}^{C+F} \tag{4}$$

with identity 26×26 matrix $\mathbf{I}$. The set (3), being formed by (1) and (2), is passed through TLP with targets (4) for Q_{pass} times.

In model of STSMI the image successively is scaled, turned, and shifted. Such succession is necessary because it is running from the easy-to-model distortion type towards the worst distortion type.

A scaled image can be formed within MATLAB by means of the function “imresize”. The map ρ, implementing this function partly, is applied to the image $\mathbf{A}_q$ as

$$\mathbf{A}_q^{\langle S\rangle}(k)=\rho\left(\mathbf{A}_q,\varsigma\left(\sigma_{\mathrm{scale}}^{\langle k\rangle}\right)\right) \tag{5}$$

with the scale coefficient $\varsigma\left(\sigma_{\mathrm{scale}}^{\langle k\rangle}\right)$ by SD

$$\sigma_{\mathrm{scale}}^{\langle k\rangle}=\frac{k}{F}\cdot\sigma_{\mathrm{scale}}^{\langle \max\rangle}\quad\forall\, k=\overline{1,F}\ \text{ for }\ \sigma_{\mathrm{scale}}^{\langle \max\rangle}>0. \tag{6}$$

The scale coefficient

$$\varsigma\left(\sigma_{\mathrm{scale}}^{\langle k\rangle}\right)=\sigma_{\mathrm{scale}}^{\langle k\rangle}\xi_{\mathrm{scale}}(k)+1 \tag{7}$$

is determined by the value $\xi_{\mathrm{scale}}(k)$ of NV with ZEUV, raffled at the k-th stage of STSMI set formation. If occurs $\varsigma\left(\sigma_{\mathrm{scale}}^{\langle k\rangle}\right)\leqslant 0$ then the corresponding NV with ZEUV is re-raffled until $\varsigma\left(\sigma_{\mathrm{scale}}^{\langle k\rangle}\right)>0$. The input image $\mathbf{A}_q$ is enlarged by $\varsigma\left(\sigma_{\mathrm{scale}}^{\langle k\rangle}\right)$ times within the map (5) if $\varsigma\left(\sigma_{\mathrm{scale}}^{\langle k\rangle}\right)>1$; the input image $\mathbf{A}_q$ is reduced by $1\Big/\varsigma\left(\sigma_{\mathrm{scale}}^{\langle k\rangle}\right)$ times within the map (5) if $\varsigma\left(\sigma_{\mathrm{scale}}^{\langle k\rangle}\right)<1$; the input image $\mathbf{A}_q$ remains non-scaled if $\varsigma\left(\sigma_{\mathrm{scale}}^{\langle k\rangle}\right)=1$.

Another NV with ZEUV is raffled at the k-th stage of STSMI set formation for turning the scaled image $\mathbf{A}_q^{\langle S\rangle}(k)$. Its value $\xi_{\mathrm{turn}}(k)$ is used into MATLAB function “imrotate”. The procedure of turning the scaled image $\mathbf{A}_q^{\langle S\rangle}(k)$ is stated implicitly as

$$\mathbf{A}_q^{\langle ST\rangle}(k)=1-\tau\left(1-\mathbf{A}_q^{\langle S\rangle}(k),\beta\left(\sigma_{\mathrm{turn}}^{\langle k\rangle}\right)\right), \tag{8}$$

where the map τ in (8) along with the argument $1-\mathbf{A}_q^{\langle S\rangle}(k)$ takes the turn angle $\beta\left(\sigma_{\mathrm{turn}}^{\langle k\rangle}\right)$ by SD

$$\sigma_{\mathrm{turn}}^{\langle k\rangle}=\frac{\sigma_{\mathrm{turn}}^{\langle \max\rangle}}{F}\cdot k\quad\forall\, k=\overline{1,F}\ \text{ for }\ \sigma_{\mathrm{turn}}^{\langle \max\rangle}>0. \tag{9}$$

Actually, the map (8) takes the negative of the scaled image $\mathbf{A}_q^{\langle S\rangle}(k)$ and rotates the input image $1-\mathbf{A}_q^{\langle S\rangle}(k)$ by

$$\beta\left(\sigma_{\mathrm{turn}}^{\langle k\rangle}\right)=\frac{180}{\pi}\sigma_{\mathrm{turn}}^{\langle k\rangle}\xi_{\mathrm{turn}}(k) \tag{10}$$

degrees around its center point. The image is turned in counterclockwise direction if $\beta\left(\sigma_{\mathrm{turn}}^{\langle k\rangle}\right)>0$; for $\beta\left(\sigma_{\mathrm{turn}}^{\langle k\rangle}\right)<0$ the image is turned clockwise; for $\beta\left(\sigma_{\mathrm{turn}}^{\langle k\rangle}\right)=0$ the image remains unturned. The returned image after map τ in (8) is inverted back, and the size of the matrix $\mathbf{A}_q^{\langle ST\rangle}(k)$ is the same as the size of the matrix $\mathbf{A}_q^{\langle S\rangle}(k)$.

Before shifting the scaled-turned image, it must have the source format 60×80. The scaled-turned by $\varsigma\left(\sigma_{\mathrm{scale}}^{\langle k\rangle}\right)\neq 1$ image is the matrix $\mathbf{A}_q^{\langle ST\rangle}(k)$ of the intermediary format $V\times H$. If $\varsigma\left(\sigma_{\mathrm{scale}}^{\langle k\rangle}\right)>1$ then the scaled-turned image is cropped as lines of their numbers

$$\left\{\left\{\overline{1,N_V}\right\},\left\{\overline{61+N_V,V}\right\}\right\} \tag{11}$$

and columns of their numbers

$$\left\{\left\{\overline{1,N_H}\right\},\left\{\overline{81+N_H,H}\right\}\right\} \tag{12}$$

in the matrix $\mathbf{A}_q^{\langle ST\rangle}(k)$ are discarded, where $\eta(x)$ is a function, returning the integer part of the number x, for integers

$$N_V=\eta\left((V-60)/2\right)+\left(\left((1+\operatorname{sign}\zeta_V)/2\right)\cdot\operatorname{sign}\left|\zeta_V\right|\right)\times$$
$$\times\operatorname{sign}\left[V/2-\eta(V/2)\right]=\eta(V/2)-30+$$
$$+\left(\frac{1+\operatorname{sign}\zeta_V}{2}\cdot\operatorname{sign}\left|\zeta_V\right|\right)\cdot\operatorname{sign}\left[\frac{V}{2}-\eta\left(\frac{V}{2}\right)\right], \tag{13}$$

$$N_H=\eta(H/2)-40+\left(\frac{1+\operatorname{sign}\zeta_H}{2}\cdot\operatorname{sign}\left|\zeta_H\right|\right)\times$$
$$\times\operatorname{sign}\left[H/2-\eta(H/2)\right], \tag{14}$$

calculated by the values $\{\zeta_V,\zeta_H\}$ of two independent NV with ZEUV, raffled every time, when the function

$\eta(x)$ is applied. If $\varsigma\left(\sigma_{scale}^{\langle k\rangle}\right)<1$ then $V<60$, $H<80$, and the scaled-turned image is contoured rectangularly with the background white color: the matrix $\mathbf{A}_q^{\langle ST\rangle}(k)$ is padded from left for

$$N_{left}=\eta\left(\frac{80-H}{2}\right)+\left(\frac{1+\operatorname{sign}\zeta_H}{2}\cdot\operatorname{sign}\left|\zeta_H\right|\right)\times \times\operatorname{sign}\left[H/2-\eta(H/2)\right] \quad (15)$$

columns of ones and from right for

$$N_{right}=80-H-N_{left} \quad (16)$$

columns of ones, and it is padded from the top for

$$N_{top}=\eta\left(\frac{60-V}{2}\right)+\left(\frac{1+\operatorname{sign}\zeta_V}{2}\cdot\operatorname{sign}\left|\zeta_V\right|\right) \times\operatorname{sign}\left[H/2-\eta(H/2)\right] \quad (17)$$

lines of ones and from bottom for

$$N_{bottom}=60-V-N_{top} \quad (18)$$

lines of ones. The cropped or contoured scaled-turned image as 60×80 matrix $\tilde{\mathbf{A}}_q^{\langle ST\rangle}(k)=\left[\tilde{a}_{uv}^{\langle q\rangle}(k)\right]_{60\times80}$ is shifted horizontally and vertically for some number of pixels. A shift constant is from two components, horizontal and vertical, where is used SD

$$\sigma_{shift}^{\langle k\rangle}=\frac{\sigma_{shift}^{\langle max\rangle}}{F}\cdot k \quad \forall\, k=\overline{1,F} \text{ and } \sigma_{shift}^{\langle max\rangle}>0. \quad (19)$$

For a 60×80 image the horizontal pixel shift (HPS) is

$$s_{hor}\left(\sigma_{shift}^{\langle k\rangle}\right)=\eta\left(8\sigma_{shift}^{\langle k\rangle}\cdot\xi_{hor}(k)\right)\times \times\left(1-\operatorname{sign}\left(\left|\eta\left(8\sigma_{shift}^{\langle k\rangle}\cdot\xi_{hor}(k)\right)\right|-80\right)\right)\Big/2+ +80\cdot\left(1+\operatorname{sign}\left(\left|\eta\left(8\sigma_{shift}^{\langle k\rangle}\cdot\xi_{hor}(k)\right)\right|-80\right)\right)\Big/2, \quad (20)$$

where $\xi_{hor}(k)$ is a value of NV with ZEUV, raffled at the k-th stage of STSMI set formation for HPS. Concurrently, vertical pixel shift (VPS) is

$$s_{ver}\left(\sigma_{shift}^{\langle k\rangle}\right)=\eta\left(6\sigma_{shift}^{\langle k\rangle}\cdot\xi_{ver}(k)\right)\times \times\left(1-\operatorname{sign}\left(\left|\eta\left(6\sigma_{shift}^{\langle k\rangle}\cdot\xi_{ver}(k)\right)\right|-60\right)\right)\Big/2+ +60\cdot\left(1+\operatorname{sign}\left(\left|\eta\left(6\sigma_{shift}^{\langle k\rangle}\cdot\xi_{ver}(k)\right)\right|-60\right)\right)\Big/2, \quad (21)$$

where $\xi_{ver}(k)$ is value of NV with ZEUV, raffled at the k-th stage of STSMI set formation for VPS. Due to the horizontal shift the matrix $\tilde{\mathbf{A}}_q^{\langle ST\rangle}(k)=\left[\tilde{a}_{uv}^{\langle q\rangle}(k)\right]_{60\times80}$ is transformed into the matrix

$$\tilde{\mathbf{A}}_q^{\langle ST\text{-}HPS\rangle}(k)=\left[\tilde{a}_{uv}^{\langle q\text{-}HPS\rangle}(k)\right]_{60\times80}.$$

For $s_{hor}\left(\sigma_{shift}^{\langle k\rangle}\right)>0$ elements of the matrix $\tilde{\mathbf{A}}_q^{\langle ST\text{-}HPS\rangle}(k)$ are

$$\tilde{a}_{uv}^{\langle q\text{-}HPS\rangle}(k)=1 \quad \text{for } u=\overline{1,60} \text{ and } v=\overline{1,s_{hor}\left(\sigma_{shift}^{\langle k\rangle}\right)} \quad (22)$$

by

$$\tilde{a}_{uv}^{\langle q\text{-}HPS\rangle}(k)=\tilde{a}_{ut}^{\langle q\rangle}(k) \quad \text{at } t=v-s_{hor}\left(\sigma_{shift}^{\langle k\rangle}\right)$$

$$\text{for } u=\overline{1,60} \text{ and } v=\overline{s_{hor}\left(\sigma_{shift}^{\langle k\rangle}\right)+1,80}. \quad (23)$$

For $s_{hor}\left(\sigma_{shift}^{\langle k\rangle}\right)<0$ elements of the matrix $\tilde{\mathbf{A}}_q^{\langle ST\text{-}HPS\rangle}(k)$ are

$$\tilde{a}_{uv}^{\langle q\text{-}HPS\rangle}(k)=\tilde{a}_{ut}^{\langle q\rangle}(k) \quad \text{at } t=v-s_{hor}\left(\sigma_{shift}^{\langle k\rangle}\right)$$

$$\text{for } u=\overline{1,60} \text{ and } v=\overline{1,80+s_{hor}\left(\sigma_{shift}^{\langle k\rangle}\right)} \quad (24)$$

by

$$\tilde{a}_{uv}^{\langle q\text{-}HPS\rangle}(k)=1 \quad \text{for } u=\overline{1,60} \text{ and } v=\overline{80+s_{hor}\left(\sigma_{shift}^{\langle k\rangle}\right)+1,80}. \quad (25)$$

Clearly, for $s_{hor}\left(\sigma_{shift}^{\langle k\rangle}\right)=0$ the q-th image is not shifted horizontally:

$$\tilde{a}_{uv}^{\langle q\text{-}HPS\rangle}(k)=\tilde{a}_{uv}^{\langle q\rangle}(k) \quad \text{for } u=\overline{1,60} \text{ and } v=\overline{1,80}. \quad (26)$$

After horizontal shift, due to the vertical shift the matrix $\tilde{\mathbf{A}}_q^{\langle ST\text{-}HPS\rangle}(k)$ is transformed into the matrix

$$\tilde{\mathbf{A}}_q^{\langle STSMI\rangle}(k)=\left[\tilde{a}_{uv}^{\langle q\text{-}STSMI\rangle}(k)\right]_{60\times80}.$$

For $s_{ver}\left(\sigma_{shift}^{\langle k\rangle}\right)>0$ elements of the matrix $\tilde{\mathbf{A}}_q^{\langle STSMI\rangle}(k)$ are

$$\tilde{a}_{uv}^{\langle q\text{-}STSMI\rangle}(k)=\tilde{a}_{rv}^{\langle q\text{-}HPS\rangle}(k) \quad \text{at } r=u+s_{ver}\left(\sigma_{shift}^{\langle k\rangle}\right)$$

$$\text{for } u=\overline{1,60-s_{ver}\left(\sigma_{shift}^{\langle k\rangle}\right)} \text{ and } v=\overline{1,80} \quad (27)$$

by

$$\tilde{a}_{uv}^{\langle q\text{-}STSMI\rangle}(k)=1 \quad \text{for } u=\overline{60-s_{ver}\left(\sigma_{shift}^{\langle k\rangle}\right)+1,60} \text{ and } v=\overline{1,80}. \quad (28)$$

For $s_{ver}\left(\sigma_{shift}^{\langle k\rangle}\right)<0$ new matrix elements are

$$\tilde{a}_{uv}^{\langle q\text{-}STSMI\rangle}(k)=1 \quad \text{for } u=\overline{1,-s_{ver}\left(\sigma_{shift}^{\langle k\rangle}\right)} \text{ and } v=\overline{1,80} \quad (29)$$

by

$$\tilde{a}_{uv}^{\langle q\text{-}STSMI\rangle}(k)=\tilde{a}_{rv}^{\langle q\text{-}HPS\rangle}(k) \quad \text{at } r=u+s_{ver}\left(\sigma_{shift}^{\langle k\rangle}\right)$$

$$\text{for } u=\overline{-s_{ver}\left(\sigma_{shift}^{\langle k\rangle}\right)+1,60} \text{ and } v=\overline{1,80}. \quad (30)$$

Clearly, for $s_{ver}\left(\sigma_{shift}^{\langle k\rangle}\right)=0$ the q-th horizontally shifted image is not shifted vertically:

$$\tilde{a}_{uv}^{\langle q\text{-}STSMI\rangle}(k)=\tilde{a}_{uv}^{\langle q\text{-}HPS\rangle}(k) \quad \text{for } u=\overline{1,60} \text{ and } v=\overline{1,80}. \quad (31)$$

After every image became scaled-turned-shifted, their matrices are column-reshaped and concatenated into the matrix $\mathbf{A}_{STSMI}^{\langle k\rangle}=\left(\tilde{a}_{jq}^{\langle k\text{-}STSMI\rangle}\right)_{4800\times26}$, ready for including it into the training set or for adding the pixel inversion distortion. In the training process with STSMI the input of TLP is fed with the training set

$$\left\{\tilde{\mathbf{P}}_i^{\langle STSMI\rangle}\right\}_{i=1}^{C+F}=\left\{\left\{\mathbf{A}\right\}_{l=1}^{C},\left\{\mathbf{A}_{STSMI}^{\langle k\rangle}\right\}_{k=1}^{F}\right\} \quad (32)$$

by targets (4), being passed through TLP for Q_{pass} times.

Model of PDSTSMI is in forming the matrix $\mathbf{A}_{PDSTSMI}^{\langle k\rangle}=\left(\tilde{a}_{jq}^{\langle k\text{-}PDSTSMI\rangle}\right)_{4800\times26}$ through the addition

$$\mathbf{A}^{\langle k\rangle}_{\text{PDSTSMI}} = \mathbf{A}^{\langle k\rangle}_{\text{STSMI}} + \sigma^{\langle k\rangle}_{\text{pixel}} \cdot \mathbf{O} \qquad (33)$$

by SD (2). Appositely, values of matrices (1) and (33) are compared to the value 0.5 via 0.5-crossing comparator: if $\overline{a}^{\langle \text{k-PDMI}\rangle}_{jq} \leqslant 0.5$ or $\tilde{a}^{\langle \text{k-PDSTSMI}\rangle}_{jq} \leqslant 0.5$ then the pixel of vertical-horizontal coordinates $\{j, q\}$ is set to the black color, else this pixel is set to the white (fig. 3). And in the training process with PDSTSMI the input of TLP is fed with the training set

$$\left\{\tilde{\mathbf{P}}^{\langle \text{PDSTSMI}\rangle}_i\right\}_{i=1}^{C+F} = \left\{\{\mathbf{A}\}_{l=1}^{C}, \left\{\mathbf{A}^{\langle k\rangle}_{\text{PDSTSMI}}\right\}_{k=1}^{F}\right\} \qquad (34)$$

by targets (4), being passed through TLP for Q_{pass} times.

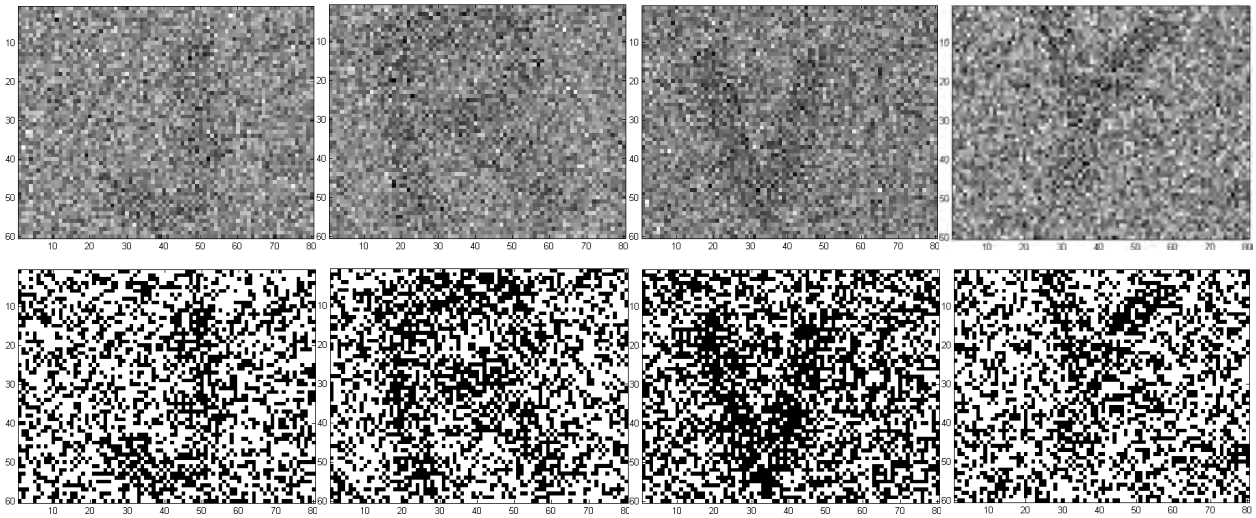

Fig. 3. Letters "J", "R", "V", "Y", viewed over grayscale from within MATLAB after their scale, turn, shift, and pixel-distortion by the corresponding SD $\sigma_{scale} = 0.1$, $\sigma_{turn} = 0.1$, $\sigma_{shift} = 0.5$, $\sigma_{pixel} = 1$, and PDSTSMI of these letters as they're set back to monochrome representation via 0.5-crossing comparator

In model of STSMI with scaling by (5) – (7), turning by (8) – (10), cropping or contouring by (11) – (18), shifting by (19) – (31) the values of the corresponding SD in (6), (9), (19) are being varied synchronously, like these SD are parts of a whole SD for forming randomized STSMI. This is realized by setting a ratio between ultimate values $\sigma^{\langle max\rangle}_{scale}$, $\sigma^{\langle max\rangle}_{turn}$, $\sigma^{\langle max\rangle}_{shift}$, being maintained for current values σ_{scale}, σ_{turn}, σ_{shift}. The same is about model of PDSTSMI, where the values of the corresponding SD in (2), (6), (9), (19) are being varied synchronously also, like these four SD are parts of a whole SD for forming randomized PDSTSMI, what is realized by setting a ratio between ultimate values $\sigma^{\langle max\rangle}_{pixel}$, $\sigma^{\langle max\rangle}_{scale}$, $\sigma^{\langle max\rangle}_{turn}$, $\sigma^{\langle max\rangle}_{shift}$, being maintained for current values σ_{pixel}, σ_{scale}, σ_{turn}, σ_{shift}.

PDMI-trained TLP fail in classifying STSMI

Let the size of TLP hidden layer be 250 neurons at $\sigma^{\langle max\rangle}_{pixel} = 1$, $C = 2$, $F = 8$, $Q_{pass} = 10$ for (1) – (4). These parameters are excellently practiced for PDMI-trained TLP [22, 23], performing perfectly in classifying PDMI. For seeing how PDMI-trained TLP classifies STSMI there is SD ratio $\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn}$ at $\sigma^{\langle max\rangle}_{shift} = 0.5$, what is assigned heuristically on the experience ground. Henceforward, in visualizing classification results over STSMI, there is the abscissa axis as shift-distortion SD σ_{shift} (fig. 4). It is no wonder that CEP $p_{error}(\sigma_{shift})$ over SD range $[0; 0.5]$ increases quasi-linearly, where starting off the point $\sigma_{shift} = 0.05$ rightwards there is quite intolerable CEP. It shows the predictable fail of PDMI-trained TLP in classifying STSMI.

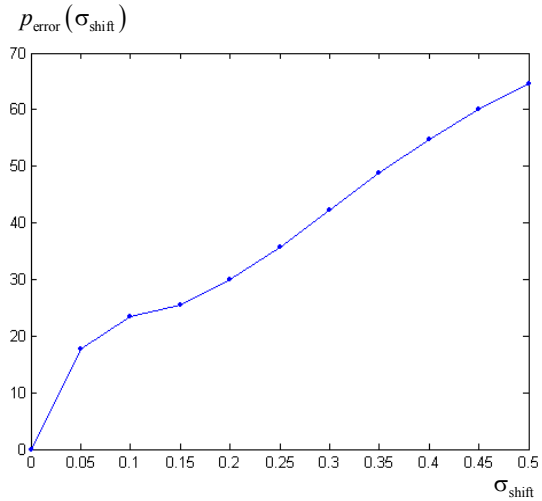

Fig. 4. CEP $p_{error}(\sigma_{shift})$ over SD range [0, 0.5] by SD ratio $\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn}$ in STSMI after 2000 batch testings of PDMI-trained TLP

For seeing how PDMI-trained TLP classifies PDSTSMI there is the heuristically assigned SD ratio $\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn} = 0.5\sigma_{pixel}$ at $\sigma^{\langle max\rangle}_{shift} = 0.5$, where in

visualizing classification results over PDSTSMI the abscissa axis is as shift-distortion SD σ_{shift} (fig. 5). Here the same intolerable CEP is, starting off the point $\sigma_{shift} = 0.05$ rightwards, and the shape of $p_{error}(\sigma_{shift})$ for PDSTSMI is almost repeated after the shape of $p_{error}(\sigma_{shift})$ for STSMI, saying that PDMI-trained TLP cannot be classifier neither for PDSTSMI, nor for STSMI.

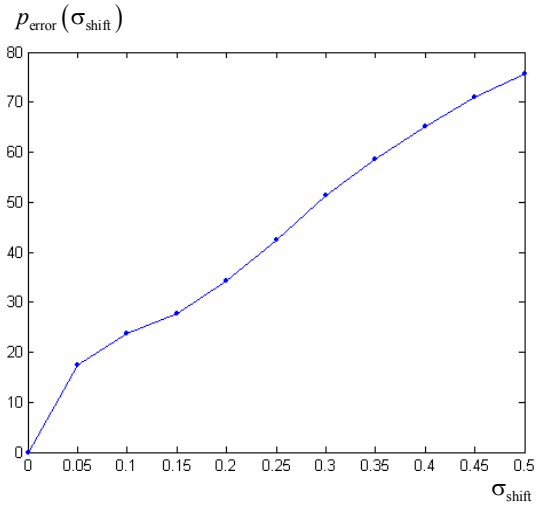

Fig. 5. CEP $p_{error}(\sigma_{shift})$ over SD range [0, 0.5] by SD ratio $\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn} = 0.5\sigma_{pixel}$ in PDSTSMI after 2000 batch testings of PDMI-trained TLP

Now TLP is to be checked out whether it can be trained with STSMI. It will be tested for classifying STSMI and PDSTSMI over the spoken above SD range [0, 0.5] of the shift distortion.

STSMI-trained TLP for classifying STSMI, PDSTSMI, PDMI

For identifying STSMI-trained TLP the size of TLP hidden layer and the rest key parameters remain the same (250 neurons in TLP hidden layer at $C = 2$ and $F = 8$), although Q_{pass} for (32) may be increased up from $Q_{pass} = 10$. Fig. 6 visualizes that STSMI-trained TLP classifies STSMI by SD ratio $\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn}$ at $\sigma_{shift}^{\langle max \rangle} = 0.5$ much better than PDMI-trained TLP, and with increasing Q_{pass} positive dynamics in CEP decrement is maintained. Nevertheless, there is much of fail when STSMI-trained TLP classifies STSMI at higher SD, closer to $\sigma_{shift} = 0.5$ point. Besides, the training process for identifying STSMI-trained TLP is some overextended, having many passes with unmet goals on reaching minimum gradient.

PDSTSMI are classified with STSMI-trained TLP seemingly twice worse (fig. 7). And fig. 8 visualizes preserved capacity of STSMI-trained TLP for classifying PDMI.

As STSMI-trained TLP is identified long and it is a bad classifier for PDSTSMI, then TLP is going to be trained with PDSTSMI by increasing the part of STSMI within the training set (34). Though it is likely that identification of PDSTSMI-trained TLP will make the training process last much longer.

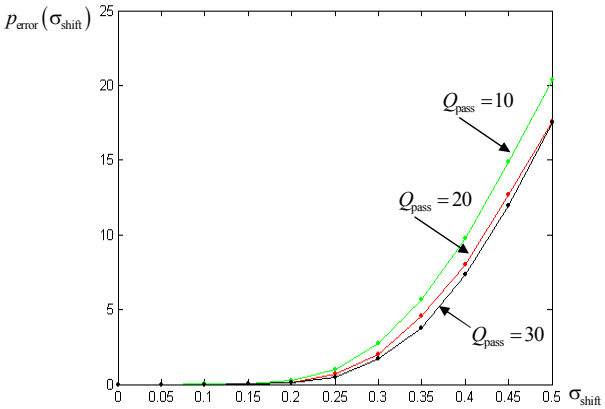

Fig. 6. CEP $p_{error}(\sigma_{shift})$ over SD range [0, 0.5] by SD ratio $\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn}$ in STSMI after 2000 batch testings of STSMI-trained TLP

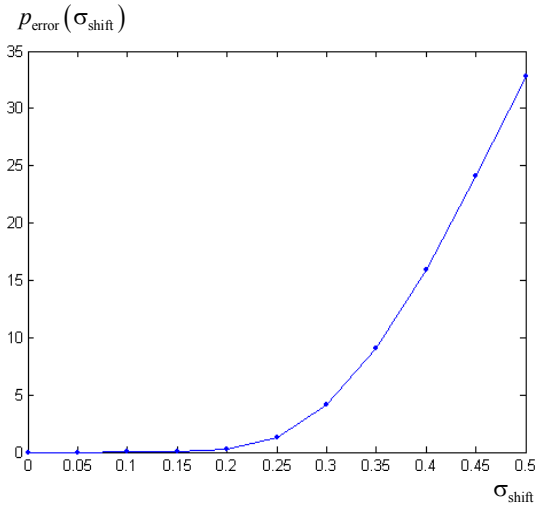

Fig. 7. CEP $p_{error}(\sigma_{shift})$ over SD range [0, 0.5] by SD ratio $\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn} = 0.5\sigma_{pixel}$ in PDSTSMI after 2000 batch testings of STSMI-trained TLP ($Q_{pass} = 30$)

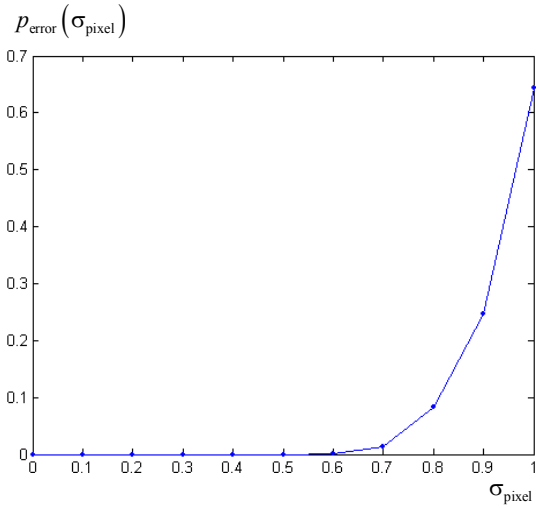

Fig. 8. CEP $p_{error}(\sigma_{shift})$ over SD range [0, 1] in PDMI after 2000 batch testings of STSMI-trained TLP ($Q_{pass} = 30$)

PDSTSMI-trained TLP for classifying STSMI, PDSTSMI, PDMI

In identification of PDSTSMI-trained TLP by 250 neurons in its hidden layer at $C = 2$ and $F = 8$, one must be ready for its overextended training process anyway. Consequently, the value of Q_{pass} for (34) may be set as great as admissible. The increment of the part of STSMI within the training set (34) is defined by increment of $\sigma_{shift}^{\langle max \rangle}$ at $\sigma_{pixel}^{\langle max \rangle} = 1$. Well, fig. 9 shows that along with increasing the part of STSMI within the training set of PDSTSMI (34) due to

$$\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn} = 0.125\sigma_{pixel},$$
$$\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn} = 0.25\sigma_{pixel},$$
$$\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn} = 0.5\sigma_{pixel}, \quad (35)$$

the positive dynamics in classifying STSMI is maintained. SD ratio in STSMI part has not been changed, $\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn}$ up at $\sigma_{shift}^{\langle max \rangle} = 0.5$, but preliminarily carried out trials hinted that taking $\sigma_{pixel}^{\langle max \rangle} = 0.125$ for SD ratio in PDSTSMI at $\sigma_{shift}^{\langle max \rangle} = 0.5$ is better than taking $\sigma_{pixel}^{\langle max \rangle} = 1$, $\sigma_{pixel}^{\langle max \rangle} = 0.5$ or $\sigma_{pixel}^{\langle max \rangle} = 0.25$, because SD $\sigma_{pixel}^{\langle max \rangle} = 0.125$ lets reach lower CEP than $\sigma_{pixel}^{\langle max \rangle} > 0.125$. So, TLP is trained with lesser part of PDMI in PDSTSMI than the part of PDMI in PDSTSMI, used for testing the trained TLP: here TLP is tested by $\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn} = 0.5\sigma_{pixel}$ at $\sigma_{pixel}^{\langle max \rangle} = 1$ in PDSTSMI, but TLP is trained by $\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn} = 4\sigma_{pixel}$ at $\sigma_{pixel}^{\langle max \rangle} = 0.125$ in PDSTSMI for obtaining a fine classifier over DDO with $\sigma_{shift}^{\langle max \rangle} = 0.5$.

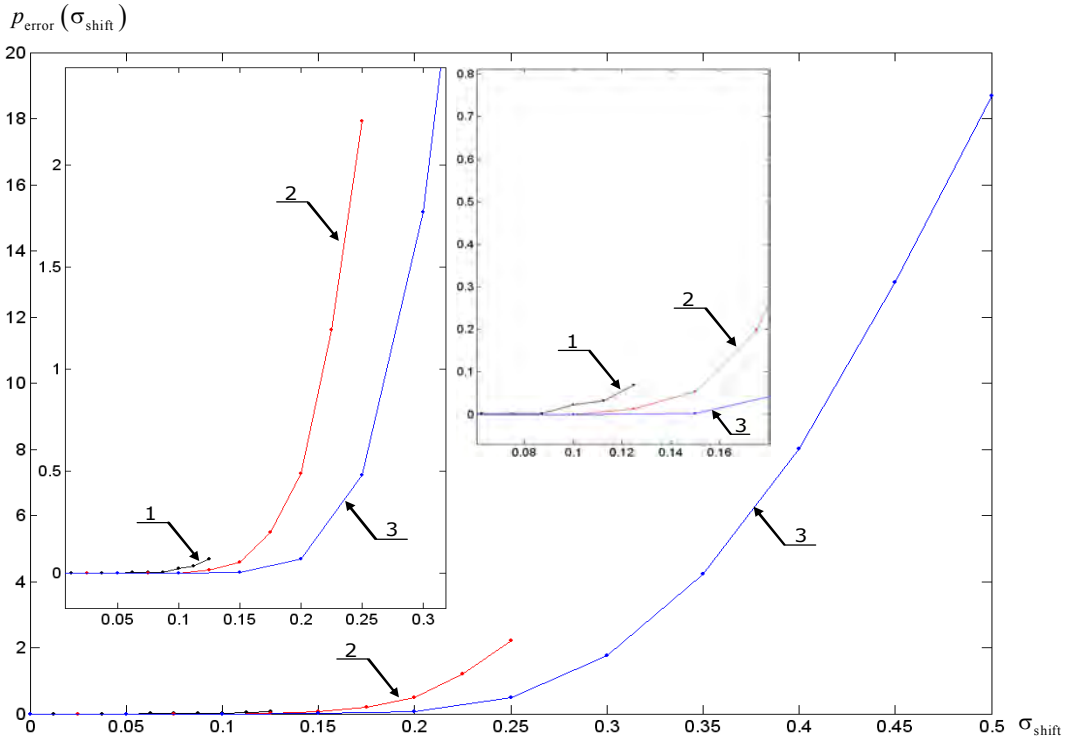

Fig. 9. CEP $p_{error}(\sigma_{shift})$ over SD ranges {[0, 0.125], [0, 0.25], [0, 0.5]} by SD ratio $\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn}$ in STSMI after 2000 batch testings of three PDSTSMI-trained TLP by increment of $\sigma_{shift}^{\langle max \rangle}$ due to (35) at $\sigma_{pixel}^{\langle max \rangle} = 1$ ($Q_{pass} = 120$): 1 – with $\sigma_{shift}^{\langle max \rangle} = 0.125$; 2 – with $\sigma_{shift}^{\langle max \rangle} = 0.25$; 3 – with $\sigma_{shift}^{\langle max \rangle} = 0.5$

Comparison of fig. 10 with PDSTSMI-trained TLP performance in it and Figure 6 reports that PDSTSMI-trained TLP performance excels STSMI-trained TLP performance. Fig. 11 shows that PDSTSMI-trained TLP by

$$\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn} = 4\sigma_{pixel} \text{ at } \sigma_{pixel}^{\langle max \rangle} = 0.125$$

classifies PDSTSMI by $\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn} = 0.5\sigma_{pixel}$ at $\sigma_{pixel}^{\langle max \rangle} = 1$ fine, whereas Figure 7 showed previously that STSMI-trained TLP was a coarser classifier over PDSTSMI. And PDSTSMI-trained TLP also is capable to classify PDMI, where CEP $p_{error}(\sigma_{pixel})$ in fig. 12 relatively doesn't yield much to CEP $p_{error}(\sigma_{pixel})$ in fig. 8, what totalizes PDSTSMI-trained TLP preference to STSMI-trained TLP.

Despite PDSTSMI-trained TLP appears a fine classifier, SD $\sigma_{shift}^{\langle max \rangle} = 0.5$ is insufficient to say TLP can classify DDO. It is sufficient to have at least SD $\sigma_{shift}^{\langle max \rangle} = 0.75$ in STSMI which are classified with TLP, whereupon PDSTSMI-trained TLP can be said it classifies DDO. This TLP can be identified on the basis of TLP, trained with PDSTSMI by $\sigma_{shift} = 5\sigma_{scale} = 5\sigma_{turn} = 4\sigma_{pixel}$

at $\sigma_{\text{pixel}}^{\langle\max\rangle} = 0.1875$ (fig. 13). From fig. 13 it is seen that CEP at $\sigma_{\text{shift}} = 0.5$ is less than 4 %, being nearly twice lower that for PDSTSMI-trained TLP by

$$\sigma_{\text{shift}} = 5\sigma_{\text{scale}} = 5\sigma_{\text{turn}} = 4\sigma_{\text{pixel}}$$

at $\sigma_{\text{pixel}}^{\langle\max\rangle} = 0.125$ and $Q_{\text{pass}} = 220$.

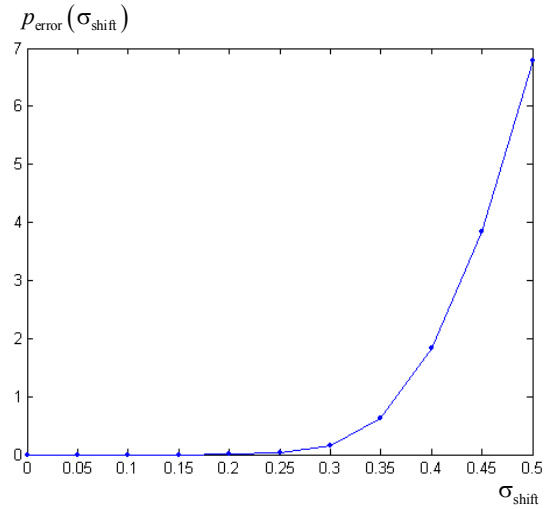

Fig. 10. CEP $p_{\text{error}}(\sigma_{\text{shift}})$ over SD range [0, 0.5] by SD ratio $\sigma_{\text{shift}} = 5\sigma_{\text{scale}} = 5\sigma_{\text{turn}}$ in STSMI after 2000 batch testings of PDSTSMI-trained TLP by $\sigma_{\text{shift}} = 5\sigma_{\text{scale}} = 5\sigma_{\text{turn}} = 4\sigma_{\text{pixel}}$ at $\sigma_{\text{pixel}}^{\langle\max\rangle} = 0.125$ and $Q_{\text{pass}} = 220$

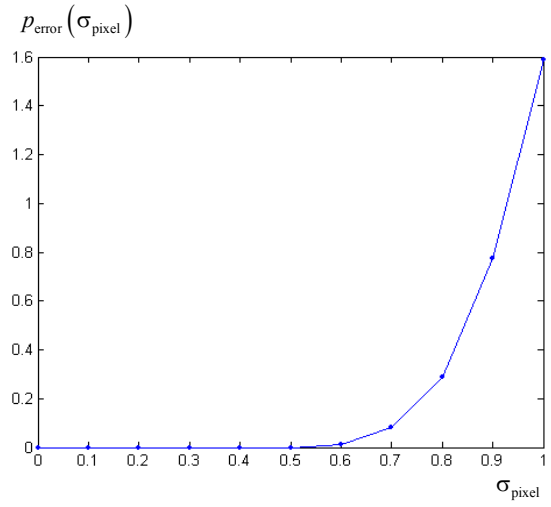

Fig. 12. CEP $p_{\text{error}}(\sigma_{\text{shift}})$ over SD range [0, 1] in PDMI after 2000 batch testings of PDSTSMI-trained TLP by $\sigma_{\text{shift}} = 5\sigma_{\text{scale}} = 5\sigma_{\text{turn}} = 4\sigma_{\text{pixel}}$ at $\sigma_{\text{pixel}}^{\langle\max\rangle} = 0.125$ and $Q_{\text{pass}} = 220$

Conclusion and problem of optimization of sigmas ratio

The results in Figures 4 – 13 underscore the principal possibility of identifying TLP classifier, which would produce near-minimal CEP over DDO. Moreover, PDSTSMI-trained TLP with capability of classifying STSMI was obtained under that ratio

$$\sigma_{\text{shift}} = 5\sigma_{\text{scale}} = 5\sigma_{\text{turn}} = 4\sigma_{\text{pixel}} \text{ at } \sigma_{\text{pixel}}^{\langle\max\rangle} = 0.1875,$$

Operation speed of PDSTSMI-trained TLP is alright: it took about 96 seconds to classify 10000 DDO of the format 60×80 on dual-core Intel® Celeron® CPU E1500 (512 K Cache, 2.20 GHz, 800 MHz FSB) by 2 GB of RAM. Neocognitron on the same configuration is many times slower [12, 24].

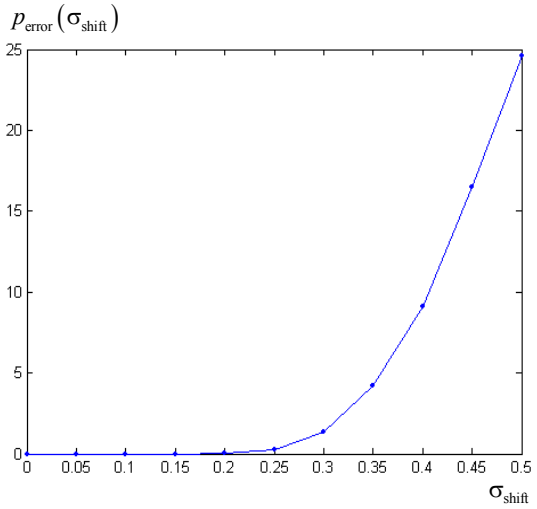

Fig. 11. CEP $p_{\text{error}}(\sigma_{\text{shift}})$ over SD range [0, 0.5] by SD ratio $\sigma_{\text{shift}} = 5\sigma_{\text{scale}} = 5\sigma_{\text{turn}} = 0.5\sigma_{\text{pixel}}$ in PDSTSMI after 2000 batch testings of PDSTSMI-trained TLP by $\sigma_{\text{shift}} = 5\sigma_{\text{scale}} = 5\sigma_{\text{turn}} = 4\sigma_{\text{pixel}}$ at $\sigma_{\text{pixel}}^{\langle\max\rangle} = 0.125$ and $Q_{\text{pass}} = 220$

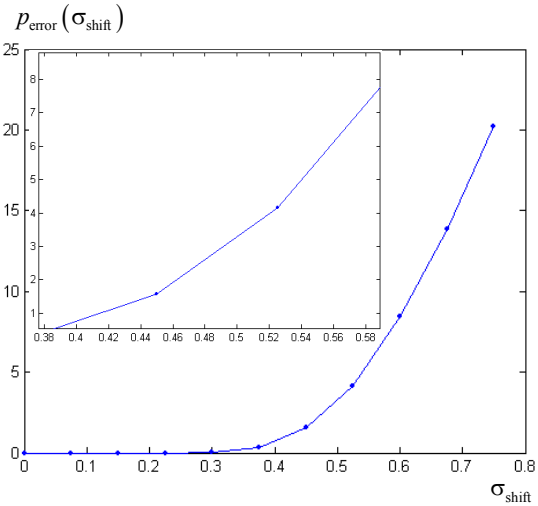

Fig. 13. CEP $p_{\text{error}}(\sigma_{\text{shift}})$ over SD range [0, 0.75] by SD ratio $\sigma_{\text{shift}} = 5\sigma_{\text{scale}} = 5\sigma_{\text{turn}}$ in STSMI after 2000 batch testings of PDSTSMI-trained TLP by $\sigma_{\text{shift}} = 5\sigma_{\text{scale}} = 5\sigma_{\text{turn}} = 4\sigma_{\text{pixel}}$ at $\sigma_{\text{pixel}}^{\langle\max\rangle} = 0.1875$ and $Q_{\text{pass}} = 280$

although this ratio is scarcely the best. Yes, the ratio of sigmas could be improved, what might give the advanced PDSTSMI-trained TLP, producing lower CEP than $p_{\text{error}}(\sigma_{\text{shift}})$ in Figure 13. At that the training process may last longer, being slow-convergent or (and) requiring greater value of Q_{pass} for (34). Surely, before ratio of sigmas $\{\sigma_{\text{shift}}, \sigma_{\text{scale}}, \sigma_{\text{turn}}, \sigma_{\text{pixel}}\}$ is optimized, problem of optimization of these sigmas ratio should be

stated [25, 26]. It's not a facile task, as there are two criterions in this optimization – to minimize CEP and to minimize the training process length, being measured in epochs for a current pass and in total number of passes Q_{pass}. But even if to work only on CEP minimization, there arises a problem with three from four coefficients at sigmas as variables in positive octant of three-dimensional Euclidean space. Solving this single-criterion problem on a subset of three-component positive points will require most computational resources. And as soon as the specific or optimized ratio for types of distortion in STSMI is determined, after the corresponding training process there will be identified the advanced PDSTSMI-trained TLP, capable to substitute neocognitron in classifying DDO.

List of references

1. Arulampalam G. A generalized feedforward neural network architecture for classification and regression / G. Arulampalam, A. Bouzerdoum // Neural Networks. – 2003. – Vol. 16, Iss. 5 – 6. – P. 561 – 568.

2. Хайкин С. Нейронные сети: полный курс / С. Хайкин. – М. : Изд. дом "Вильямс", 2006. – 1104 с.

3. Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position / K. Fukushima // Biological Cybernetics. – 1980. – Vol. 36, Iss. 4. – P. 193 – 202.

4. Fukushima K. Neural network model for selective attention in visual pattern recognition and associative recall / K. Fukushima // Applied Optics. – 1987. – Vol. 26, Iss. 23. – P. 4985 – 4992.

5. Fukushima K. Increasing robustness against background noise: Visual pattern recognition by a neocognitron / K. Fukushima // Neural Networks. – 2011. – Vol. 24, Iss. 7. – P. 767 – 778.

6. Fukushima K. Artificial vision by multi-layered neural networks: Neocognitron and its advances / K. Fukushima // Neural Networks. – 2013. – Vol. 37. – P. 103 – 119.

7. Fukushima K. Neocognitron: A hierarchical neural network capable of visual pattern recognition / K. Fukushima // Neural Networks. – 1988. – Vol. 1, Iss. 2. – P. 119 – 130.

8. Fukushima K. Neocognitron for handwritten digit recognition / K. Fukushima // Neurocomputing. – 2003. – Vol. 51. – P. 161 – 180.

9. Fukushima K. Training multi-layered neural network neocognitron / K. Fukushima // Neural Networks. – 2013. – Vol. 40. – P. 18 – 31.

10. You J. Classification and segmentation of rotated and scaled textured images using texture "tuned" masks / J. You, H. A. Cohen // Pattern Recognition. – 1993. – Vol. 26, Iss. 2. – P. 245 – 258.

11. Wei W. Rapid, man-made object morphological segmentation for aerial images using a multi-scaled, geometric image analysis / W. Wei, Y. Xin // Image and Vision Computing. – 2010. – Vol. 28, Iss. 4. – P. 626 – 633.

12. Poli G. Parallel face recognition processing using neocognitron neural network and GPU with CUDA high performance architecture / G. Poli, J. H. Saito // Face Recognition. – Ed. by Milos Oravec. – InTech, Rijeka, Croatia, 2010. – P. 381 – 404.

13. Cireşan D. Multi-column deep neural network for traffic sign classification / D. Cireşan, U. Meier, J. Masci, J. Schmidhuber // Neural Networks. – 2012. – Vol. 32. – P. 333 – 338.

14. Siniscalchi S. M. Exploiting deep neural networks for detection-based speech recognition / S. M. Siniscalchi, D. Yu, L. Deng, C.-H. Lee // Neurocomputing. – 2013. – Vol. 106. – P. 148 – 157.

15. Mrazova I. Can deep neural networks discover meaningful pattern features? / I. Mrazova, M. Kukacka // Procedia Computer Science. – 2012. – Vol. 2. – P. 194 – 199.

16. Hagan M. T. Training feedforward networks with the Marquardt algorithm / M. T. Hagan, M. B. Menhaj // IEEE Transactions on Neural Networks. – 1994. – Vol. 5, Iss. 6. – P. 989 – 993.

17. Yu C. An efficient hidden layer training method for the multilayer perceptron / C. Yu, M.T. Manry, J.Li, P.L. Narasimha // Neurocomputing. – 2006. – Vol. 70, Iss. 1 – 3. – P. 525 – 535.

18. Plaza J. On the use of small training sets for neural network-based characterization of mixed pixels in remotely sensed hyperspectral images / J. Plaza, A. Plaza, R. Perez, P. Martinez // Pattern Recognition. – 2009. – Vol. 42, Iss. 11. – P. 3032 – 3045.

19. Hagiwara K. Upper bound of the expected training error of neural network regression for a Gaussian noise sequence / K. Hagiwara, T. Hayasaka, N. Toda, S. Usui, K. Kuno // Neural Networks. – 2001. – Volume 14, Issue 10. – P. 1419 – 1429.

20. Moller M. F. A scaled conjugate gradient algorithm for fast supervised learning / M. F. Moller // Neural Networks. – 1993. – Vol. 6, Iss. 4. – P. 525 – 533.

21. Kathirvalavakumar T. Neighborhood based modified backpropagation algorithm using adaptive learning parameters for training feedforward neural networks / T. Kathirvalavakumar, S. Jeyaseeli Subavathi // Neurocomputing. – 2009. – Vol. 72, Iss. 16 – 18. – P. 3915 – 3921.

22. Castillo P.A. Comparing evolutionary hybrid systems for design and optimization of multilayer perceptron structure along training parameters / P.A. Castillo, J.J. Merelo, M.G. Arenas, G. Romero // Information Sciences. – 2007. – Vol. 177, Iss. 14. – P. 2884 – 2905.

23. Romanuke V.V. Setting the hidden layer neuron number in feedforward neural network for an image recognition problem under Gaussian noise of distortion / V.V. Romanuke // Computer and Information Science. – 2013. – Vol. 6, N. 2. – P. 38 – 54.

24. Kangin D. Further parameters estimation of neocognitron neural network modification with FFT convolution / D. Kangin, G. Kolev, A. Vikhoreva // Journal of Telecommunication, Electronic and Computer Engineering. – 2012. – Vol. 4, Iss. 2. – P. 21 – 26.

25. Romanuke V.V. Pixel-to-scale standard deviations ratio optimization for two-layer perceptron training on pixel-distorted scaled 60-by-80-images in scaled objects classification problem / V.V. Romanuke // Вісник Кременчуцького національного університету імені Михайла Остроградського. – 2014. – Вип. 2 (85). – С. 96 – 105.

26. Romanuke V.V. Pixel-to-turn standard deviations ratio optimization for training two-layer perceptron on pixel-distorted turned 60-by-80-images in turned objects classification problem / V.V. Romanuke // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. Математичне моделювання і прикладна механіка. – Запоріжжя: Запорізький національний університет, 2014. – № 1. – С. 147 – 170.

Надійшла до редколегії 22.04.2015

Рецензент: д-р техн. наук, проф. Р.В. Сорокатий, Хмельницький національний університет, Хмельницький.

ДВОШАРОВИЙ ПЕРСЕПТРОН ДЛЯ КЛАСИФІКАЦІЇ МАСШТАБОВАНИХ ОБ'ЄКТІВ З ПОВОРОТОМ І ЗСУВОМ ЗА ГЕНЕРАЛЬНОЇ СУКУПНОСТІ З 26 КЛАСІВ З МОНОХРОМНИХ ЗОБРАЖЕНЬ ФОРМАТУ 60-НА-80 ЗА ДОПОМОГОЮ НАВЧАННЯ НА МАСШТАБОВАНИХ ЗОБРАЖЕННЯХ З ПОВОРОТОМ І ЗСУВОМ ПРИ ПІКСЕЛЬНИХ СПОТВОРЕННЯХ

В. В. Романюк

Випробовується двошаровий персептрон на свою ідентифікацію для класифікації різноманітно спотворених об'єктів. Двошаровий персептрон моделюється, навчається та тестується у середовищі MATLAB. Збільшивши число проходів до 280 для навчання на масштабованих зображеннях з поворотом і зсувом та з піксельним спотворенням, продуктивність персептрона покращується як для виключно масштабованих зображень з поворотом і зсувом, так і для масштабованих зображень з поворотом і зсувом та з піксельними спотвореннями, де зображення з піксельними спотвореннями класифікуються так чи інакше відмінно. Зрештою, оскільки неокогнітрон за тієї самої конфігурації MATLAB й операційної системи є у багато разів повільнішим, то стверджується, що двошаровий персептрон здатний замінити неокогнітрон для класифікації різноманітно спотворених об'єктів, потребуючи незвично довшого процесу навчання та специфічного відношення для типів спотворення, що залежить від типу об'єкта.

Ключові слова: *класифікація різноманітно спотворених об'єктів, неокогнітрон, двошаровий персептрон, монохромне зображення, піксельні спотворення, навчальна вибірка, відсотковий рівень помилок класифікації.*

ДВУХСЛОЙНЫЙ ПЕРСЕПТРОН ДЛЯ КЛАССИФИКАЦИИ МАСШТАБИРОВАННЫХ ОБЪЕКТОВ С ПОВОРОТОМ И СДВИГОМ ПРИ ГЕНЕРАЛЬНОЙ СОВОКУПНОСТИ ИЗ 26 КЛАССОВ ИЗ МОНОХРОМНЫХ ИЗОБРАЖЕНИЙ ФОРМАТА 60-НА-80 ПОСРЕДСТВОМ ОБУЧЕНИЯ НА МАСШТАБИРОВАННЫХ ИЗОБРАЖЕНИЯХ С ПОВОРОТОМ И СДВИГОМ ПРИ ПИКСЕЛЬНЫХ ИСКАЖЕНИЯХ

В.В. Романюк

Испытывается двухслойный персептрон на свою идентификацию для классификации разнообразно искажённых объектов. Двухслойный персептрон моделируется, обучается и тестируется в среде MATLAB. Увеличив число проходов до 280 для обучения на масштабированных изображениях с поворотом и сдвигом при пиксельном искажении, производительность персептрона улучшается как для исключительно масштабированных изображений с поворотом и сдвигом, так и для масштабированных изображений с поворотом и сдвигом при пиксельных искажениях, где изображения с пиксельными искажениями классифицируются так или иначе отлично. В конце концов, поскольку неокогнитрон при той же конфигурации MATLAB и операционной системы медленнее во много раз, то утверждается, что двухслойный персептрон способен заменить неокогнитрон для классификации разнообразно искажённых объектов, требуя непривычно длинного процесса обучения и специфического отношения для типов искажения, что зависит от типа объекта.

Ключевые слова: *классификация разнообразно искажённых объектов, неокогнитрон, двухслойный персептрон, монохромное изображение, пиксельные искажения, обучающая выборка, процентный уровень ошибок классификации.*