

Analysis of X-Ray Moiré Images Using Artificial Neural Networks

Serhiy V. Balovsyak^a, Ihor M. Fodchuk^a, Khrystyna S. Odaiska^a, Yuriy T. Roman^a, Elena Zaitseva^b

^a Yuriy Fedkovych Chernivtsi National University, 2, Kotsiubynsky str., Chernivtsi, 58012, Ukraine

^b Zilina University, Univerzitna 8215, Zilina, 01026, Slovakia

Abstract

A method for the analysis of digital X-ray moiré images using artificial neural networks such as "multilayer perceptron" has been developed. Moiré images are obtained by the action of many forces on the surface of the analyzer crystal. Image analysis consisted in solving the inverse problem, namely in calculating the values of forces based on the intensity distribution of the moiré image. Moiré image contours after logarithmization, zooming and convolution were used as input signals for the neural network. Due to this image processing, the learning time of the neural network is reduced and the learning error remains low. The output signals of the neural network are the values of the set (series) of forces that generated the moiré image. Artificial neural network training was performed by the back propagation method. The training sample consisted of a series of calculated moiré images with known values of sets of forces. The artificial neural network was developed by Python in Google Colab cloud platform. The results of testing the developed program showed high accuracy of restoring the values of the set of forces in the analysis of calculated and experimental moiré images.

Keywords

Artificial neural network, multilayer perceptron, inverse problem, X-ray moiré image, image contours, Python, Google Colab.

1. Introduction

X-ray moiré images carry valuable information about the studied crystals due to their high sensitivity to small deformations of the crystal lattice, which allows to determine the relative deformations of the lattice with an accuracy of 10^{-8} [1-2]. However, the task of calculating the intensity distribution of the X-ray moiré image obtained using an LLL interferometer (triple Laue interferometer) is quite difficult due to the influence of many different factors: defects of the crystal, geometric characteristics of the interferometer and others [3-5]. In this paper, it is believed that the deformation of the crystal is due to many forces that cause the appearance of a characteristic intensity distribution of the moiré image in the form of light and dark bands. One way to calculate the values of the intensity of the moiré image is the numerical solution of a system of differential equations of the hyperbolic type (Takagi equations) [2-5]. In addition, the inverse problem of calculating the values of the set of forces based on the moiré image is even more complex and is not solved analytically.

Therefore, the paper proposes to calculate the values of the forces set on the basis of moiré images using artificial neural networks (ANN) type "multilayer perceptron" [6-9]. The advantage of using ANN as a means of artificial intelligence in this case is the ability to train ANN on examples (training set), which does not require the construction of a complex mathematical model of X-ray

IntelliTSIS'2022: 3rd International Workshop on Intelligent Information Technologies and Systems of Information Security, March 23–25, 2022, Khmelnytskyi, Ukraine

EMAIL: s.balovsyak@chnu.edu.ua (S. Balovsyak); ifodchuk@ukr.net (I. Fodchuk); k.odaiska@chnu.edu.ua (Kh. Odaiska); y.roman@chnu.edu.ua (Yu. Roman); elena.zaitseva@fri.uniza.sk (E. Zaitseva)

ORCID: <https://orcid.org/0000-0002-3253-9006> (S. Balovsyak); 0000-0001-6772-6920 (I. Fodchuk); 0000-0002-3167-1195 (Kh. Odaiska); 0000-0002-9964-9971 (Yu. Roman); 0000-0002-9087-0311 (E. Zaitseva)



© 2022 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

interferometer. Moiré image contours after logarithmization, zooming and convolution were used as input signals for the ANN [10-12]. The image contours are analyzed because the values of band periods (and their respective spatial frequencies) are extremely informative for X-ray moiré images, and the values of such band periods are proportional to the distances between the band contours. Logarithmization of the image intensity distribution allows to analyze both strong and weak signals, and image convolution is used to smooth them and identify common patterns. Zooming of images was used to reduce the training time of ANN. The output signals of the ANN are the values of the set of forces that caused the appearance of the moiré image. ANN training was performed by the method of back propagation. The training set consists of a series of calculated moiré images with known values of force sets. The ANN was developed by Python language [13-15] in Google Colab cloud platform [16]. After training, ANN is able to solve the inverse problem, i.e. to restore the value of the forces set acting on the crystal, based on the analysis of calculated and experimental moiré images.

2. Method of analysis of X-ray moiré images using artificial neural networks

2.1. Sequence of formation and processing of moiré images

Initial digital moiré images f are formed and processed as rectangular matrices $f = (f(i, k))$, where $i = 0, \dots, M_i-1$; $k = 0, \dots, N_k-1$; M_i is the height of the image in pixels, N_k is the width of the image in pixels [10-12]. The brightness of the images f is normalized in the range from 0 to 1. Experimental moiré images are scanned from photographic plates or read from CCD-detectors, and the simulated images are calculated by numerical solution of the system of differential equations [1-5]. In all cases, the intensity distribution of the image f depends on the values of the set of forces P_n , where $n = 0, \dots, N-1$, which act in the X-ray LLL interferometer on the surface of the studied crystal. The set of N such forces are in some way placed spatially, for example, linearly with a given step.

Before analyzing moiré images f by ANN, it is necessary to perform their special processing to highlight the informative signs of the images and reduce their dimensionality. Such image processing consists of contouring, logarithmization, zooming and low-pass filtering.

The selection of contours [11-12] is performed to quantify the period of dark and light bands on moiré images, because the distance between the contours describes the periods of the respective bands. The calculation of contours is performed by the Sobel method by convolving the image f with the kernel of the filter w_{SX} to select horizontal contours S_X and with the kernel of the filter w_{SY} to select vertical contours S_Y , where the filter kernels are described by formulas

$$w_{SX} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, w_{SY} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}. \quad (1)$$

The resulting contours $c_{n0} = (c_{n0}(i, k))$, where $i = 0, \dots, M_{iC}-1$; $k = 0, \dots, N_{kC}-1$, are calculated as the root of the sum of the squares of the horizontal S_X and vertical S_Y contours. To avoid edge effects, the image of the contours ($M_{iC} \times N_{kC}$ pixels) is reduced (compared to image f) by the bandwidth B_C ($B_C = 5$ pixels) around the perimeter of the image, i.e. $M_{iC} = M_i - 2B_C$, $N_{kC} = N_k - 2B_C$. In the images of contours c_{n0} , the intensity values of different sections can differ significantly, so for further analysis of all components of the signal with different intensities, the image of contours c_{nA} with a logarithmic scale in intensity is used.

The obtained images of contours c_{nA} have a fairly high resolution, so the supply time to the inputs of the ANN pixel intensities training time will be long. Therefore, to reduce the training time of ANN on the basis of images c_{nA} , the images of contours c_n on a reduced scale ($M_{iw} \times N_{kw}$ pixels) were calculated by cubic interpolation. Reducing the scale of images c_n is done to such a size $M_{iw} \times N_{kw}$, which does not yet increase the training error of ANN.

Scalable images of contours c_n describe the local features of each band and do not have generalizing characteristics. That is, when training the images of the contours of the ANN will be able to accurately restore the values of the forces P_n only for those images that almost completely coincide with the images of the training set. Therefore, to give the ANN generalizing properties the low-frequency filtering [10-12] of the image of contours c_n by convolution with a kernel $w = (w(m, n))$ of size $M_w \times N_w$ in the spatial region is used according to the formula

$$f_w(i_w, k_w) = \sum_{m=0}^{M_w-1} \sum_{n=0}^{N_w-1} c_n(i_w - m - m_c, k_w - n - n_c) \cdot w(m, n) \quad (2)$$

where $f_w = (f_w(i_w, k_w))$ is filtered image; $i_w = 0, \dots, M_{i_w}-1, k_w = 0, \dots, N_{k_w}-1$;
 $m_c = (M_{w2} + 1)$ is the center of the filter kernel in height;
 $n_c = (N_{w2} + 1)$ is the center of the filter kernel in width;
 M_{w2}, N_{w2} – whole parts of half the size of the filter kernel.

The operation of convolution the image c_n with the kernel w is simply written in the form $f_w = c_n * w$. The two-dimensional Gaussian function with standard deviation (SD) σ_w was used as the kernel of the filter w .

2.2. Mathematical model of an artificial neural network

A three-layer perceptron with back propagation training method was used as ANN [7-9]. A mathematical model of ANN has been developed, which describes the structure of the neural network and its functioning in the modes of training, testing and forecasting.

The structure of the ANN is as follows (Fig. 1):

1. The input layer X ; the states of its elements are written in the vector $X = (X_i)$, where $i = 0, \dots, Q_X$. The size of the training set (number of vectors) at the inputs X is equal to Q_N , the vector number $n_i = 0, \dots, Q_N-1$. The inputs of layer X are fed to the values of the corresponding pixels of the filtered image of the contours f_w (2), normalized in the range from 0 to 1. Layer X does not contain a matrix of weights, so it is not taken into account in the total number of layers.

2. Hidden layers

Layer V^1 (level $L = 1$); the states of its elements are written in the vector $V^1 = (V^1_{k1})$, where $k_1 = 0, \dots, Q_{V1}$; the weights of the layer are written in the matrix $W^1 = (W^1_{i, k1})$; the difference of the layer vectors (during training) is written in the vector $D^1 = (D^1_{k1})$, where $k_1 = 0, \dots, Q_{V1}$.

Layer V^2 ($L = 2$): the states of its elements are written in the vector $V^2 = (V^2_{k2})$, where $k_2 = 0, \dots, Q_{V2}$; the weights of the layer are written in the matrix $W^2 = (W^2_{k1, k2})$; the difference of the layer vectors (during training) is written in the vector $D^2 = (D^2_{k2})$, where $k_2 = 0 \dots Q_{V2}$.

3. Source layer Y ($L = 3$): the states of its elements are written in the vector $Y = (Y_j)$, where $j = 0, \dots, Q_Y$; the weights of the layer are written in the matrix $W^3 = (W^3_{k2, j})$; the difference of the layer vectors (during training) is written in the vector $D^3 = (D^3_j)$. True output (true) is described by the vector $Y^T = (Y^T_j)$, where $j = 0, \dots, Q_Y$, which is recorded normalized in the range from 0 to 1 values of forces P_n , where $n = 0, \dots, N-1$ ($Q_Y = N-1$).

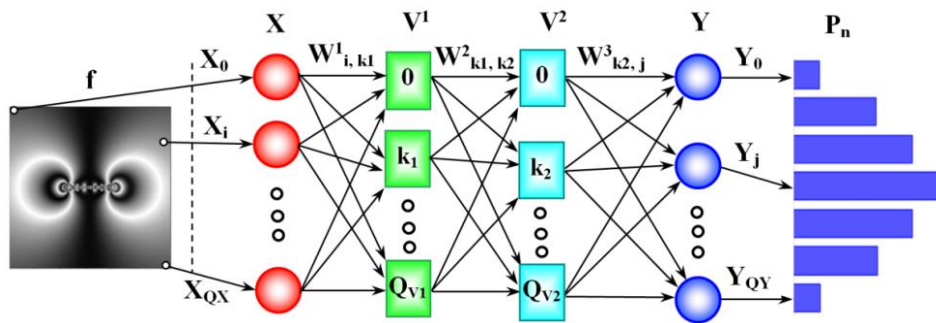


Figure 1: Structure of a multilayer perceptron with input and output signals

The normalization of the values of the vector Y^T is performed by the formula

$$Y_j^T = P_j / P_{nMax} \quad (3)$$

where P_{nMax} is the maximum value of the set of forces P_n ; $j = 0 \dots Q_Y$.

Accordingly, to calculate the predicted values of forces P_{nN} at the outputs of the ANN, the values of the vector Y are multiplied by the coefficient P_{nMax} .

ANN is studied according to the algorithm of back propagation in this sequence (Fig. 2).

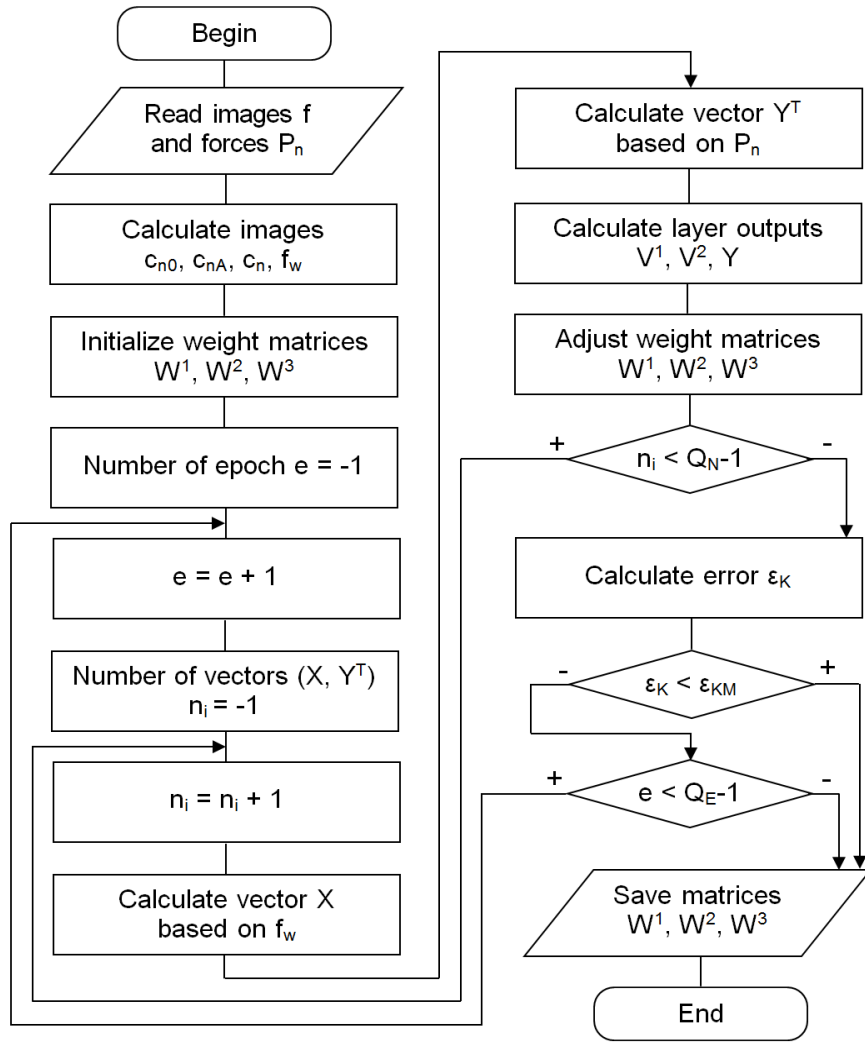


Figure 2: Scheme of ANN training algorithm

1. Initialization. The initial weights W for all layers are set equal to small random values in the range $[-\Delta W, .. \Delta W]$ (for example, $\Delta W = 0.1$) using a uniformly distributed random variable. For example, for the hidden layer V^1 , the initial values of the weights are calculated by the formula:

$$W_{i,k1}^1 = 2(R_{nd} - 0.5) \cdot \Delta W, \quad (4)$$

where R_{nd} is uniformly distributed in the range from 0 to 1 random variable, $i = 0, \dots, Q_X$; $k_1 = 0, \dots, Q_{V1}$.

In weight matrices W , rows correspond to the elements from which the connections come, and columns correspond to the elements to which the connections go.

2. Normalization (scaling) of the values of all X and Y^T vectors in the range from 0 to 1. Normalization of vector values is due to the fact that the ANN uses sigmoid activation functions of neurons whose output signals are in the range from 0 to 1.

3. Direct propagation is to find the output vector Y based on the input X by the following formulas:

Layer 1:

$$Net_{k1} = \sum_{i=0}^{Q_X} X_i \cdot W_{i,k1}^1, V_{k1}^1 = \frac{1}{1 + \exp(-Net_{k1})}, \quad (5)$$

where $k_1 = 0, \dots, Q_{V1}$.

Layer 2:

$$Net_{k2} = \sum_{k1=0}^{Q_{V1}} V_{k1}^1 \cdot W_{k1,k2}^2, V_{k2}^2 = \frac{1}{1 + \exp(-Net_{k2})}, \quad (6)$$

where $k_2 = 0, \dots, Q_{V2}$.

Layer 3:

$$Net_j = \sum_{k2=0}^{Q_{V2}} V_{k2}^2 \cdot W_{k2,j}^3, Y_j = \frac{1}{1 + \exp(-Net_j)}, \quad (7)$$

where $j = 0, \dots, Q_Y$.

As a result of direct propagation, the mean squared error (mse) is calculated (for all vectors of the training set) by the formula:

$$\varepsilon_K = \frac{1}{QN \cdot (QY + 1)} \sum_{n=0}^{QN-1} \sum_{j=0}^{QY} (Y_{j,n} - Y_{j,n}^T)^2 \quad (8)$$

If the value of the mean square error ε_K is less than the specified value, the network training process ends.

4. Back propagation is the correction of weights through the difference signals D .

Layer 3:

$$D_j^3 = Y_j(1 - Y_j)(Y_j^T - Y_j), W_{k2,j}^{3(e)} = W_{k2,j}^{3(e-1)} + \eta_Y D_j^3 V_{k2}^2, \quad (9)$$

where $j = 0, \dots, Q_Y$; e is epoch number; because the sigmoid activation function of neurons is used, the difference of vectors $(Y^T - Y)$ is multiplied by the derivative of the sigmoid function: $Y(1 - Y)$.

Layer 2:

$$D_{k2}^2 = \sum_{j=0}^{Q_Y} V_{k2}^2 (1 - V_{k2}^2) (D_j^3 \cdot W_{k2,j}^3), W_{k1,k2}^{2(e)} = W_{k1,k2}^{2(e-1)} + \eta_{L2} D_{k2}^2 V_{k1}^1, \quad (10)$$

where $k_2 = 0, \dots, Q_{V2}$.

Layer 1:

$$D_{k1}^1 = \sum_{k2=0}^{Q_{V2}} V_{k1}^1 (1 - V_{k1}^1) (D_{k2}^2 \cdot W_{k1,k2}^2), W_{i,k1}^{1(e)} = W_{i,k1}^{1(e-1)} + \eta_{L1} D_{k1}^1 X_i, \quad (11)$$

where $k_1 = 0, \dots, Q_{V1}$; $\eta_Y, \eta_{L2}, \eta_{L1}$ – training standards (norms) (e.g. 0.1).

The mode of ANN testing consists in direct signal propagation, i.e. in calculation by means of formulas (5-7) on the basis of input moiré images of input signals X and output Y , as well as predicted values of forces P_{nN} at ANN outputs. In the test mode, the predicted values of forces P_{nN} are compared with the exact (pre-known) values of forces P_n .

The prediction mode is similar to the test mode, but in this mode only the predicted values of forces P_{nN} are calculated, and their exact values P_n are unknown.

2.3. Software implementation of moiré image analysis using artificial neural networks

The program "p_Moire_Neuro" for the analysis of X-ray moiré images was created on the basis of an algorithm (Fig. 2) in Python [13-15] using the cloud platform Google Colab in a web shell Jupyter Notebook [16]. The program provides reading of initial images f in various graphic formats (in particular in tiff and jpg formats). The values of the correct output signals of the ANN, namely the set of forces P_n for the training set are read from text files. The values of the forces predicted by means of ANN are displayed in text and graphic formats.

The program imported a number of libraries, including the "numpy" library for mathematical calculations, the "matplotlib" library for graphing and image visualization, the "pandas" library for processing text file data, the "scipy" and "cv2" libraries for contouring and image scaling. Low-pass filtering is performed by the function "gaussian_filter" of the submodule "ndimage" of the "scipy" library. Uniformly distributed random variables were generated by the "random.uniform" function of the "numpy" library.

The "tensorflow" library for working with neural networks was not used to create the ANN model, and the functions of the "numpy" library were mainly used to process signals in the ANN. This solution partially reduces the speed of the program when training ANN, but provides more flexibility in choosing the structure and modes of operation of the neural network.

3. Results of neural network analysis of X-ray moiré images

ANN training was performed by the program "p_Moire_Neuro" on the basis of a training set, which contained 15 calculated moiré images f with known values of forces P_n [4-5], which caused their formation (Fig. 3). In the image file name, the first digit means the form of force distribution (S1 – with a minimum in the center, S2 – uniform distribution, S3 – with a maximum in the center), the next number means the sum of forces, and the set of numbers after "PL" means forces P_n (Fig. 4).

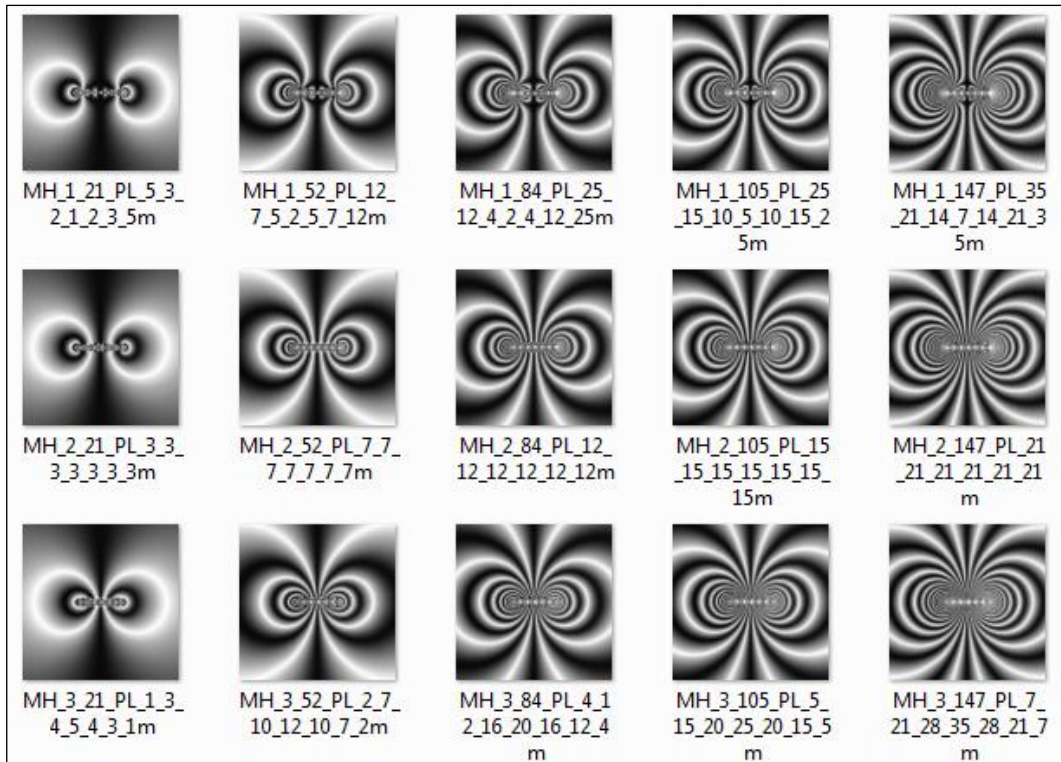


Figure 3: Image of the training set

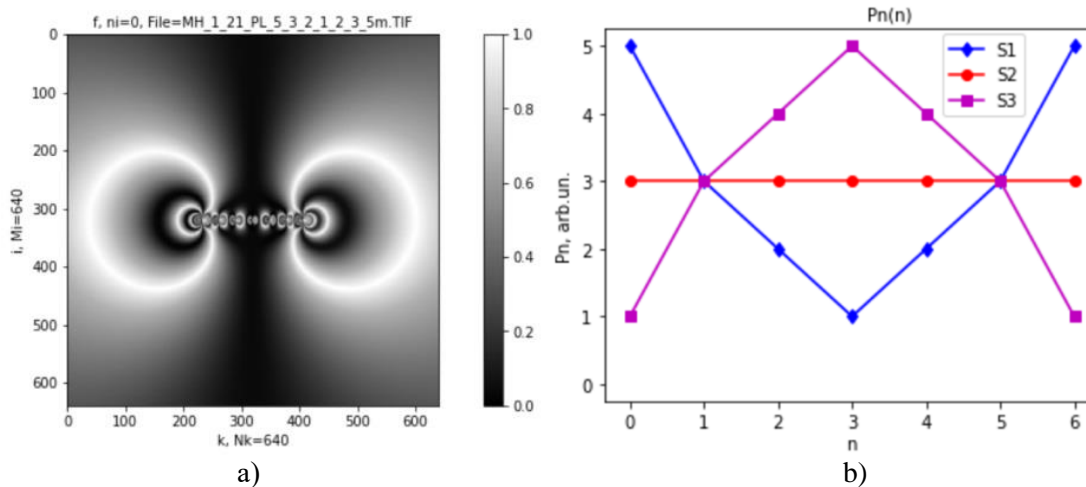


Figure 4: The calculated image f of the training set, which is formed by the action of the forces $P_n = (5, 3, 2, 1, 2, 3, 5)$ (a) and the three main shapes (S1, S2, S3) of the distribution of forces (b)

In this work, the values of P_n forces were measured in relative units, but it is possible to convert them into absolute (N) [4-5]. On the basis of the initial images f their contours are calculated, logarithmization, zooming and filtering are performed (Fig. 5 – Fig. 7). The contours of the images were calculated by the Sobel method in terms of height and width, and the image size f was reduced from the initial ($M_i \times N_k$ pixels) to the size of the filtered image of the contours f_w ($M_{iw} \times N_{kw}$ pixels).

To prevent retraining in addition to training set (Fig. 3) also formed a test set of images with known values of forces (Fig. 8). Distributions of test set forces have shapes similar to the distributions of training set forces, but quantitatively different from them.

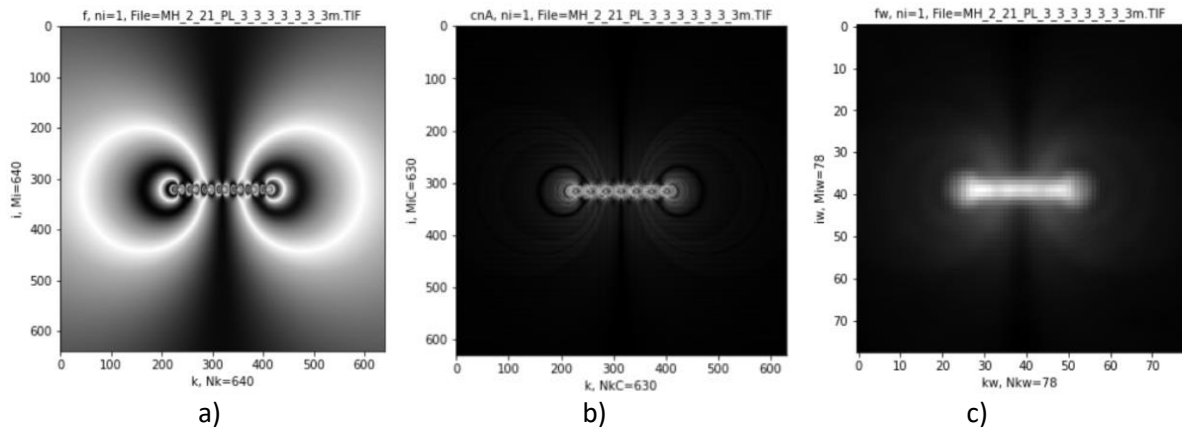


Figure 5: Calculated image f of the training set, which is formed by the action of the forces $P_n = (3, 3, 3, 3, 3, 3)$ (a); calculated on the basis of image f the contours c_{nA} after logarithm (b); calculated on the basis of c_{nA} the filtered image f_w on a reduced scale (c)

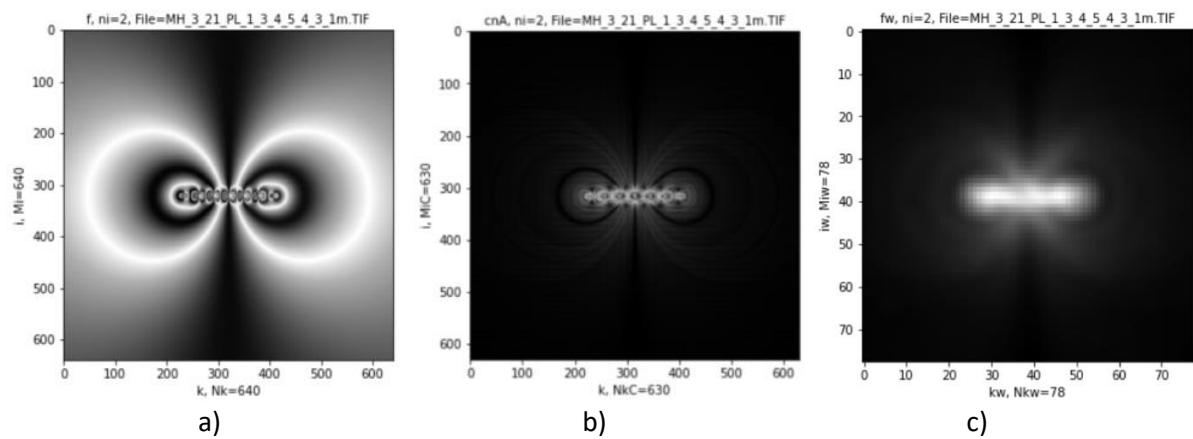


Figure 6: Calculated image f of the training set, which is formed by the action of the forces $P_n = (1, 3, 4, 5, 4, 3, 1)$ (a); calculated on the basis of image f the contours c_{nA} after logarithm (b); calculated on the basis of c_{nA} the filtered image f_w on a reduced scale (c)

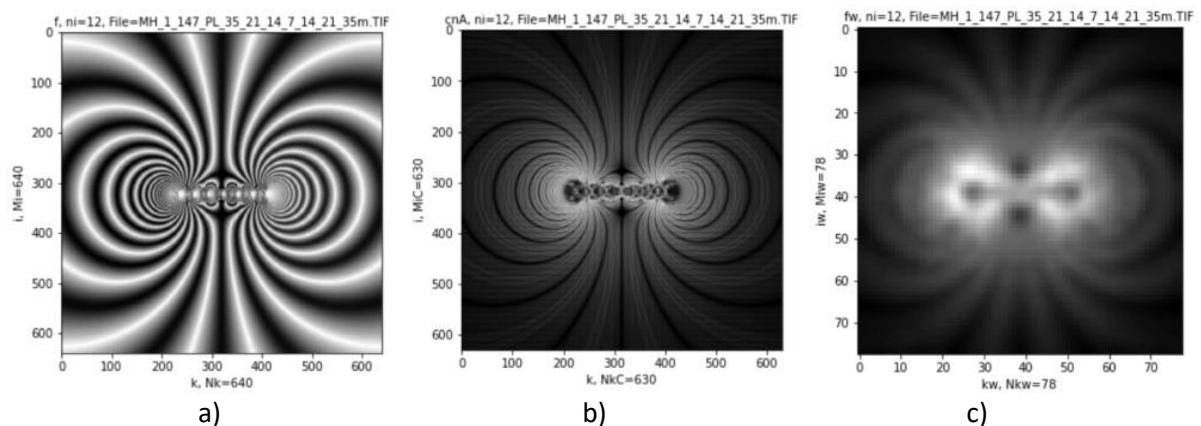


Figure 7: Calculated image f of the training set, which is formed by the action of the forces $P_n = (35, 21, 14, 7, 14, 21, 35)$ (a); calculated on the basis of image f the contours c_{nA} after logarithm (b); calculated on the basis of c_{nA} the filtered image f_w on a reduced scale (c)

The peculiarity of the test sample is that it was not taken into account in the training process to change the weight of the ANN, and was used only to calculate the training error for the test set (error was calculated similarly to the error for the training set). ANN training was performed for a specified number of epochs, but was interrupted with increasing error for the test set (Fig. 9).

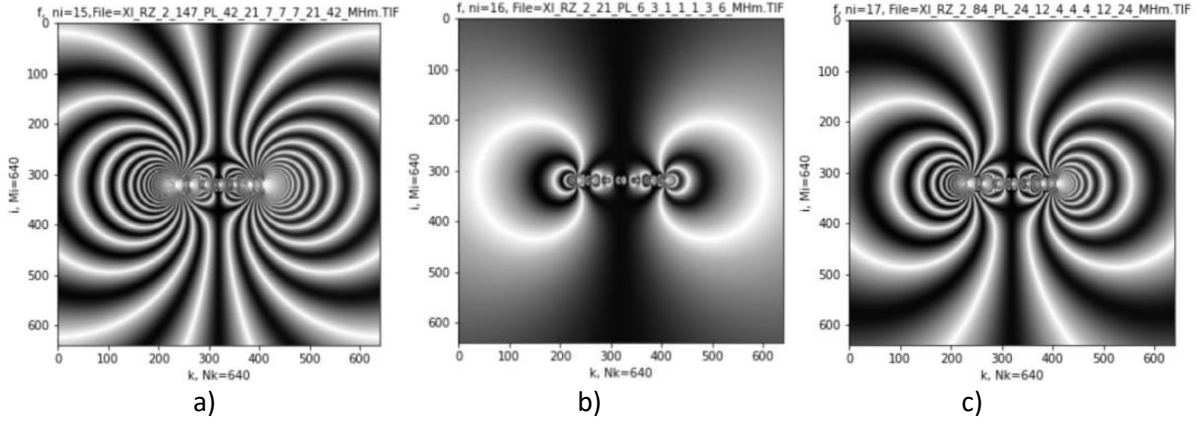


Figure 8: Calculated images f of the test set, which are formed by the action of the forces: a) $P_n = (42, 21, 7, 7, 7, 21, 42)$; b) $P_n = (6, 3, 1, 1, 1, 3, 6)$; c) $P_n = (24, 12, 4, 4, 4, 12, 24)$

Training of ANN was carried out during $Q_E = 5000$ epochs. For moiré images f with size $M_i \times N_k = 640 \times 640$ pixels, filtered images of contours f_w with dimensions $M_{iw} \times N_{kw} = 78 \times 78$ pixels are calculated, so the number of ANN inputs is equal to $(Q_X + 1) = 6084$. The number of neurons in the first hidden layer is set equal to $(Q_{V1} + 1) = 64$, the number of neurons in the second hidden layer is $(Q_{V2} + 1) = 32$. This number of neurons (Q_{V1}, Q_{V2}) was used, because with more neurons in the hidden layers the value of the training error ϵ_K (8) does not decrease, but significantly increases the training time; at the same time, with fewer neurons (Q_{V1}, Q_{V2}), the training error ϵ_K increases.

The values of training norms are set equal to $\eta_Y = 0.09$, $\eta_{L2} = 0.09$, $\eta_{L1} = 0.09$, because at lower values of norms the training time increases significantly without changing the training error ϵ_K , and at higher values of norms the error ϵ_K increases.

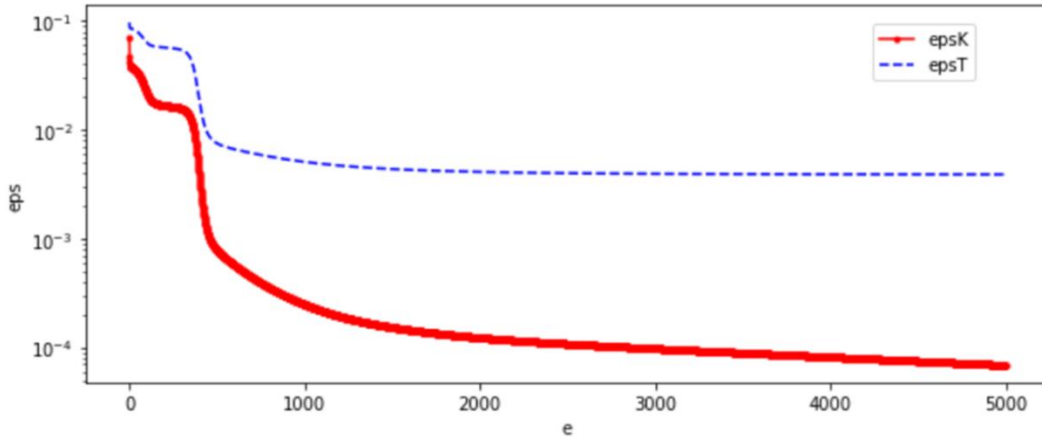


Figure 9: Dependence of training error on the number of epoch e ; ϵ_K (epsK) is training error for the training set, ϵ_T (epsT) is training error for the test set; the minimum error for the training set $\epsilon_{KM} = 6.9 \cdot 10^{-5}$; training time $t_{TR} = 1391.9$ s.

As a result of ANN training, the values of the predicted P_{nN} forces compared to the correct P_n for all images of the training set were almost identical, as evidenced by the small values of the mean square error eps (Fig. 10). For images of the test set, the distribution of forces for which differs from the distribution of forces of the training set, a satisfactory agreement of the calculated forces P_{nN} with the theoretical P_n is obtained (Fig. 11).

This can be explained by the fact that, thanks to the training of ANN, it became able to establish the relationship between the intensity distribution of moiré images and the distribution of forces that led to their formation. After training, ANN was used to calculate the values of forces on the basis of both calculated (Fig. 12) and experimental moiré images (Fig. 13).

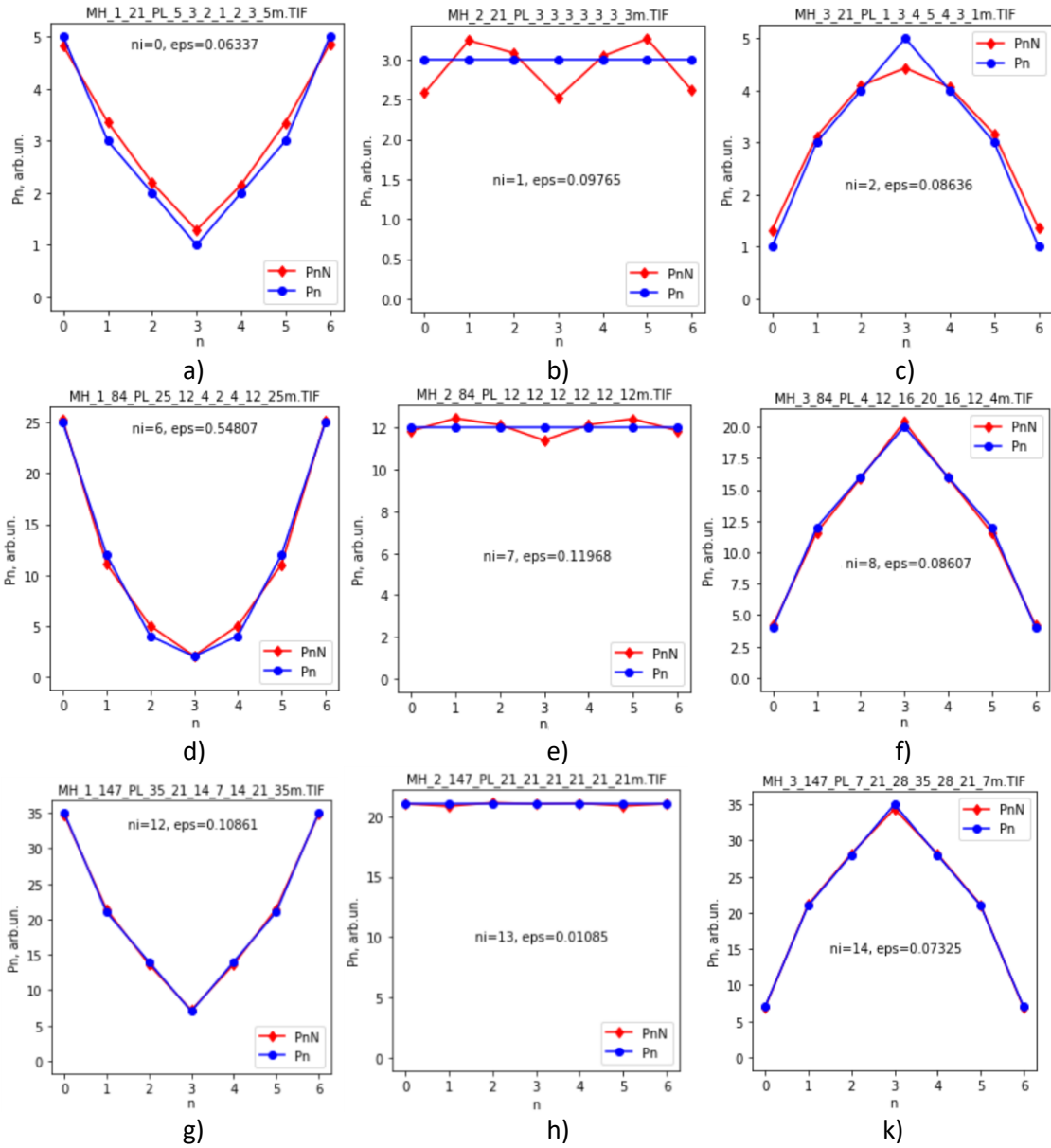


Figure 10: Examples of calculated P_{nN} and theoretical P_n forces for the training set:

- a) $P_n = (5, 3, 2, 1, 2, 3, 5)$; b) $P_n = (3, 3, 3, 3, 3, 3, 3)$; c) $P_n = (1, 3, 4, 5, 4, 3, 1)$;
d) $P_n = (25, 12, 4, 2, 4, 12, 25)$; e) $P_n = (12, 12, 12, 12, 12, 12, 12)$; f) $P_n = (4, 12, 16, 20, 16, 12, 4)$;
g) $P_n = (35, 21, 14, 7, 14, 21, 35)$; h) $P_n = (21, 21, 21, 21, 21, 21, 21)$; i) $P_n = (7, 21, 28, 35, 28, 21, 7)$

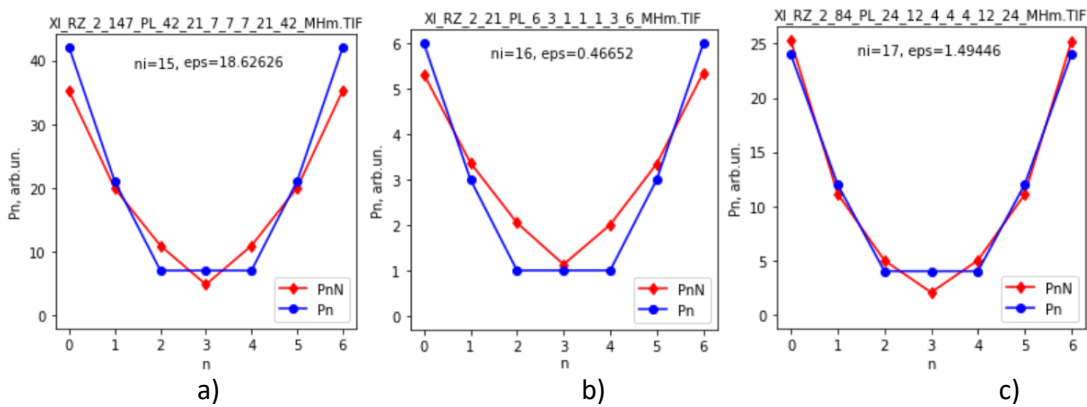


Figure 11: Examples of calculated P_{nN} and theoretical P_n forces for the test set:

- a) $P_n = (42, 21, 7, 7, 7, 21, 42)$; b) $P_n = (6, 3, 1, 1, 1, 3, 6)$; c) $P_n = (24, 12, 4, 4, 4, 12, 24)$

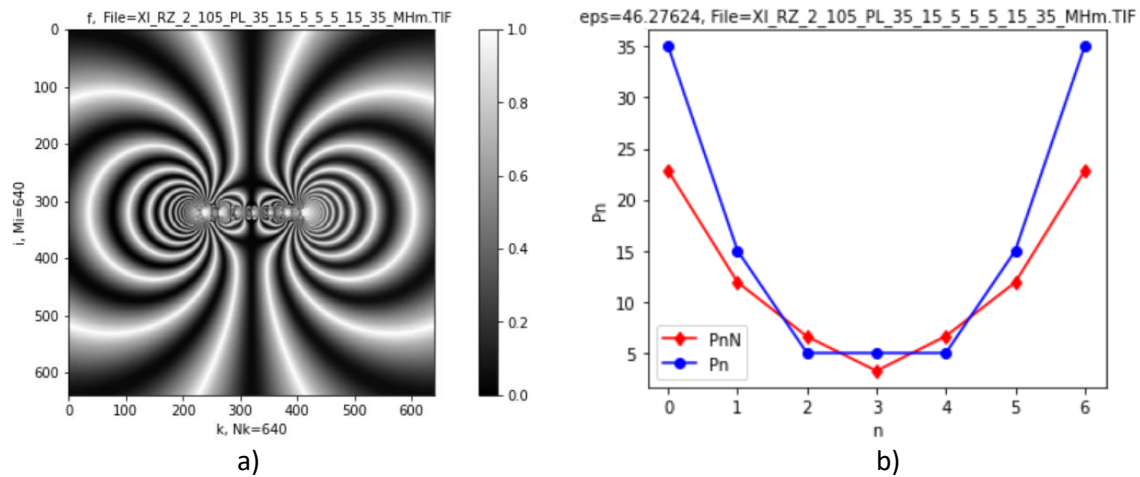


Figure 12: Calculated moiré image (a) and the results of the calculation of P_{nN} forces using ANN in comparison with the theoretical values of $P_n = (35, 15, 5, 5, 5, 15, 35)$ (b)

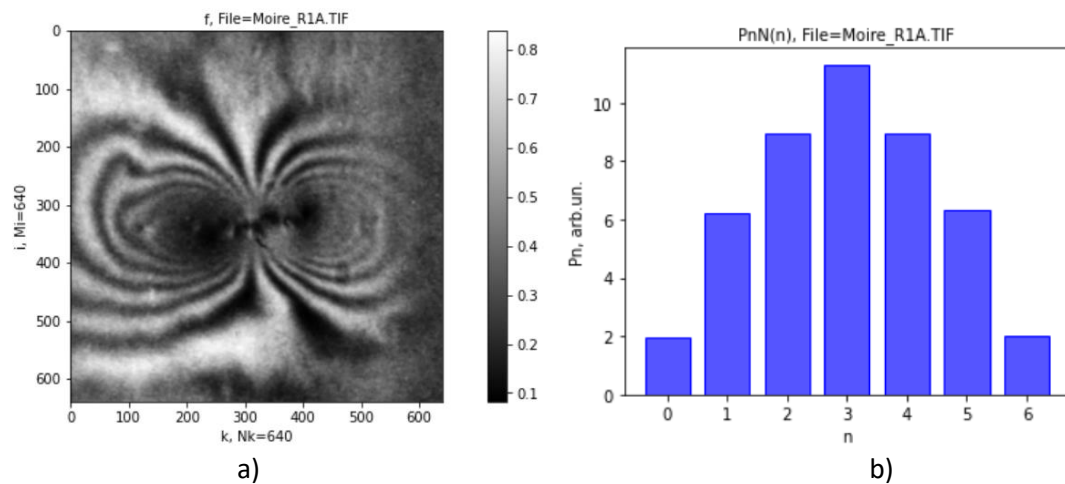


Figure 13: Experimental moiré image [5] (a) and the results of the calculation of forces $P_{nN} = (1.98, 6.22, 8.96, 11.31, 8.94, 6.35, 2.01)$ using ANN (b)

In the case of experimental moiré images, the obtained values of forces are consistent with the data of other research methods [2-5].

4. Conclusion

A method for analyzing digital X-ray moiré images using artificial neural networks has been developed. Image analysis consisted in solving the inverse problem, namely in calculating the values of forces whose action on the surface of the studied crystal led to the formation of a moiré image. A multilayer perceptron was used as an ANN, the input signals of which were the contours of the moiré image after logarithmization, scaling and filtering. Due to this image processing, the training time of the ANN is reduced and the training error remains low. The output signals of the ANN were the values of the set of forces, the action of which generated a moiré image. ANN training was performed by the back propagation method. The training set consisted of a series of calculated moiré images with known values of forces. The artificial neural network was developed in Python in the cloud platform Google Colab. The results of testing the developed program showed high accuracy of restoring the values of the forces in the analysis of calculated and experimental moiré images.

The scientific novelty of the work is the use as input signals of the artificial neural network not directly the pixel intensities of the moiré image, but the values of reduced, logarithmic and filtered image contours, which reduces training time and prevents retraining of the artificial neural network.

An important advantage of the developed method of analysis of moiré images using artificial neural networks is the ability to restore forces with their arbitrary geometric placement on the surface of the studied crystal. To analyze new types of moiré images, it is enough to train the developed artificial neural network on the basis of data from a new training set.

5. Acknowledgements

We are grateful to S. M. Novikov, I. V. Yaremchuk (Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine) for their help in obtaining the X-ray moiré images used in our study, as well as for useful advice on research topic.

This publication has been supported of the Integrated Infrastructure Operational Program for the project "Creation of a Digital Biobank to Support the Systemic Public Research Infrastructure" (ITMS: 313011AFG4) co-financed by the European Regional Development Fund.

6. References

- [1] H. Yoshioka, Y. Kadono, T. Kim, et al. Imaging evaluation of the cartilage in rheumatoid arthritis patients with an X-ray phase imaging apparatus based on Talbot-Lau interferometry, *Sci. Rep.* 10 (2020) 6561-1-6561-9. doi: <https://doi.org/10.1038/s41598-020-63155-9>.
- [2] M. Kageyama, et al., X-ray phase-imaging scanner with tiled bent gratings for large-field-of-view nondestructive testing, *NDT & E International*, 105 (2019) 19-24. doi: [10.1016/j.ndteint.2019.04.007](https://doi.org/10.1016/j.ndteint.2019.04.007).
- [3] Momose Lab. Optics of X-ray interferometry, 2021. URL: <http://mml.tagen.tohoku.ac.jp/en/research/xray-interferometer>.
- [4] I. M. Fodchuk, S. M. Novikov, I. V. Yaremchuk, Direct and inverse problems in X-ray three-crystal triple Laue case interferometry, *Appl. Optics.*, 55 12 (2016) B120 – B125. doi: [10.1364/AO.55.00B120](https://doi.org/10.1364/AO.55.00B120).
- [5] I. M. Fodchuk, S. V. Balovsyak, S. M. Novikov, I. V. Yanchuk, V. F. Romankevych, Reconstruction of spatial distribution of strains in crystals using the energy spectrum of X-ray Moiré patterns, *Ukr. J. Phys. Opt.*, Vol. 21, No. 3 (2020) 141-151. doi: [10.3116/16091833/21/3/141/2020](https://doi.org/10.3116/16091833/21/3/141/2020).
- [6] N. K. Manaswi, *Deep Learning with Applications Using Python*, Apress, India, 2018. doi: [10.1007/978-1-4842-3516-4](https://doi.org/10.1007/978-1-4842-3516-4).
- [7] E. R. Ranschaert. S. Morozov, P. R. Algra, *Artificial Intelligence in Medical Imaging. Opportunities, Applications and Risks*, Springer Nature Switzerland, 2019. doi: [10.1007/978-3-319-94878-2](https://doi.org/10.1007/978-3-319-94878-2).
- [8] O. Berezsky, S. Verbovy, O. Pitsun, "Hybrid Intelligent Information Technology for Biomedical Image Processing". *IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*, Lviv (2018): 420-423. doi: [10.1109/STC-CSIT.2018.8526711](https://doi.org/10.1109/STC-CSIT.2018.8526711).
- [9] Tensorflow. Open source machine learning platform, 2021. URL: <https://www.tensorflow.org>.
- [10] S. Krigg, *Computer Vision Metrics. Survey, Taxonomy, and Analysis*, Apress, Berkeley, CA, 2014. doi: [10.1007/978-1-4302-5930-5](https://doi.org/10.1007/978-1-4302-5930-5).
- [11] P. Russo, *Handbook of X-ray Imaging. Physics and Technology*, CRC Press, Taylor & Francis Group, 2018.
- [12] R. Gonzalez, R. Woods, *Digital image processing*, 4th edition, Pearson/ Prentice Hall, NY, 2018.
- [13] F. Nelli, *Python Data Analytics. Data Analysis and Science Using Pandas, matplotlib, and the Python Programming Language*, Apress, 2015.
- [14] O. Embarak, *Data Analysis and Visualization Using Python*, Apress, Higher Colleges of Technology, Abu Dhabi, United Arab Emirates, 2018. doi: [10.1007/978-1-4842-4109-7](https://doi.org/10.1007/978-1-4842-4109-7).
- [15] Scipy Lecture Notes. One document to learn numerics, science, and data with Python, 2021. URL: www.scipy-lectures.org.
- [16] Google Colab, 2021. URL: <https://colab.research.google.com>.