ЗМІСТ

Секція 1
Інтелектуальні інформаційні технології

Григорович В.Г. ПРОБЛЕМИ ПОБУДОВИ СЕМАНТИЧНИХ ВІДНОШЕНЬ ТА МЕТРИКИ ДЛЯ ОЦІНЮВАННЯ КОНЦЕПТІВ В ОНТОЛОГІЯХ ………... **6**

Клапата Н.І., Лучкевич М.М. СИСТЕМА ІДЕНТИФІКАЦІЇ ВІДБИТКІВ ПАЛЬЦІВ ЛЮДИНИ …..………..………... **9**

Секція 2
Моделювання та дослідження складних систем

Бурда Ю.Р. РОЗРОБЛЕННЯ ДОДАТКУ «МАСАЖНИЙ САЛОН» ……...….. **15**

Дорошенко М.В. ЦИФРОВА ОБРОБКА ЗОБРАЖЕНЬ ЗАСОБАМИ ПАКЕТУ WAVELET TOOLBOX …………………. **18**

Красиленко В.Г., Нікітович Д.В. ВДОСКОНАЛЕННЯ ТА МОДЕЛЮВАННЯ МЕТОДУ ГЕНЕРУВАННЯ ПОТОКУ МАТРИЧНИХ ПЕРЕСТАНОВОК ЗНАЧНОЇ РОЗМІРНОСТІ …………………. **23**

Krasilenko V.G., Lazarev A.A., Nikitovich D.V. DEVELOPMENT AND SIMULATION ARRAY OF DEVICES BASED ON THE FPGA FOR PARALLEL CALCULATION OF NORMALIZED EQUIVALENCES OF THE REFERENCE FILTERS WITH THE CURRENT PROCESSED FRAGMENT OF THE IMAGE ………..…. **37**

Лазурчак Л.В. СЕРЕДОВИЩЕ РОЗРОБКИ RAD STUDIO ДЛЯ СТВОРЕННЯ ВЛАСНИХ ПРОЕКТІВ …………………….. **51**

Макарова Л.М., Торопов В.М. ЙМОВІРНІСНІ МОДЕЛІ РОЗПОДІЛУ ЧАСУ НАПРАЦЮВАННЯ МІЖ ВІДМОВАМИ БАНКОМАТІВ ……………..….. **53**

Пелещак Р.М., Даньків О.О., Кузик О.В., Михавчак Г.В. МОДЕЛЮВАННЯ ПЕРЕБУДОВИ ЗОННОЇ СТРУКТУРИ НАПІВПРОВІДНИКА ПІД ВПЛИВОМ ЛАЗЕРНОГО ОПРОМІНЕННЯ …….. **58**

*Приходько С.Б., Книрік К.О.*НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ РОЗРОБКИ МОБІЛЬНИХ ЗАСТОСУНКІВ НА ПОЧАТКОВОМУ ЕТАПІ ПРОЕКТУВАННЯ …………… **64**

Приходько С.Б., Приходько Н.В., Місюра А.О. ПОБУДОВА НЕЛІНІЙНОЇ РЕГРЕСІЙНОЇ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ PHP-ЗАСТОСУНКІВ З ВІДКРИТИМ КОДОМ ЗА МЕТРИКАМИ ДІАГРАМИ КЛАСІВ ……………….…………... **67**

Сікора О.В., Жидик В.Б. ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ УДОСКОНАЛЕННЯ ПРОЦЕСУ КОНДУКТИВНОГО СУШІННЯ ПОЛІГРАФІЧНИХ МАТЕРІАЛІВ ………..… **71**

Ших Н.В., Шаклеіна Н.В. АНАЛІЗ ПОПУЛЯРНИХ СЕРЕДОВИЩ КРОС-ПЛАТФОРМНОЇ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ ……………... **74**

Секція 3
Інформаційні технології соціокомунікаційних систем та мереж

Васьків Н.Ф. РОЗРОБЛЕННЯ ВЕБ-ПЛАТФОРМИ «FREELANCE» …………... **79**
Мартинов С.В. РОЗРОБЛЕННЯ
МОБІЛЬНОГО ДОДАТКУ «АВТООРГАНАЙЗЕР» ………………………… **72**
Моржин А.Ю. РОЗРОБЛЕННЯ КЛІЄНТСЬКОЇ
ЧАСТИНИ СТРІМІНГОВОГО ВІДЕО-СЕРВІСУ ……………………………… **85**
Собків Р.А. ІГРОВИЙ ДОДАТОК "WHAT IS THIS SONG?" ………………..… **88**

Секція 4
Інформаційно-комунікативні технології в освіті та наукових дослідженнях

Вдовичин Т.Я. ОСОБЛИВОСТІ ЗАСТОСУВАННЯ
ІКТ УЧАСНИКАМИ ОСВІТНЬОГО ПРОЦЕСУ ЗВО …………………………… **91**
Воронка Н.В. ОСОБЛИВОСТІ ЗАСТОСУВАННЯ ДИФЕРЕНЦІЙОВАНОГО
НАВЧАННЯ НА УРОЦІ ІНФОРМАТИКИ В ПОЧАТКОВІЙ ШКОЛІ ………. **93**
Гальчишак М.Р. ОСОБЛИВОСТІ МИСЛЕННЄВОЇ
ДІЯЛЬНОСТІ МОЛОДШИХ ШКОЛЯРІВ НА УРОЦІ
ІНФОРМАТИКИ В ПОЧАТКОВІЙ ШКОЛІ …………..……………………… **97**
Гарбич-Мошора О.Р. ПЕРСПЕКТИВИ
ВИКОРИСТАННЯ MEGAPOLIS.DOCNET ………….……………………….. **102**
Григорович А.Г., Сосяк Р.М., Хевпа Д.Я. СИСТЕМА
ДИСТАНЦІЙНОГО НАВЧАННЯ ОСНОВАМ ПРОГРАМУВАННЯ ……….... **104**
Гринько В.О. РЕАЛІЗАЦІЯ НАВЧАЛЬНО-ДОСЛІДНИЦЬКОГО
ПРОЕКТУ НА ОСНОВІ ВИКОРИСТАННЯ
ЦИФРОВИХ ТЕХНОЛОГІЙ ……………………………………………..….. **108**
Дорошенко М.В., Шаповаловський А.О. ДИСТАНЦІЙНЕ ВИВЧЕННЯ
ДИСЦИПЛІНИ «МЕТОДИ ОБ’ЄКТНО-ОРІЄНТОВАНОГО
ПРОГРАМУВАННЯ» НА ОСНОВІ ПЛАТФОРМИ EFRONT ……………..... **112**
Здобиляк У.М., Василиків І.Б. ЗАСТОСУВАННЯ PLICKERS
НА УРОКАХ МАТЕМАТИКИ В ПОЧАТКОВІЙ ШКОЛІ …………………..… **115**
Кобильник Т.П. СТАТИСТИЧНЕ СЕРЕДОВИЩЕ R ЯК ЗАСІБ
НАВЧАННЯ ДИСЦИПЛІНИ «ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ» …… **118**
Когут У.П. ВПРОВАДЖЕННЯ КВЕСТ-ТЕХНОЛОГІЇ В ОСВІТНІЙ ПРОЦЕС **120**
Кутняк О.А. СТВОРЕННЯ МУЛЬТИМЕДІЙНИХ
ДИДАКТИЧНИХ МАТЕРІАЛІВ ЗАСОБАМИ LEARNINGAPPS ……………... **123**
Мойко О.С. ОСОБЛИВОСТІ ВИКОРИСТАННЯ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ В СИСТЕМІ ВИЩОЇ ШКОЛИ ………... **125**
Сікора О.В., Когут У.П., Вдовичин Т.Я. ПРОЕКТУВАННЯ СИСТЕМИ
ДИСТАНЦІЙНОГО НАВЧАННЯ СТУДЕНТІВ ………………………………… **130**
ВІДОМОСТІ ПРО АВТОРІВ ……………………………………………………… **133**

16. Красиленко В.Г. Моделювання протоколів узгодження секретного матричного ключа для криптографічних перетворень та систем матричного типу / В.Г. Красиленко, Д.В. Нікітович // Системи обробки інформації. – 2017. – Вип. 3 (149). – С 151-157.

17. Красиленко В.Г. "Моделювання багатокрокових та багатоступеневих протоколів узгодження секретних матричних ключів" / В.Г. Красиленко, Д.В. Нікітович // Комп`ютерно-інтегровані технології: освіта, наука, виробництво: науковий журнал. – Луцьк: ЛНТУ, 2017. – Вип. 26. – С. 111-120. – Режим доступу: http://ki.lutsk-ntu.com.ua/node/134/section/27.

18. Красиленко В.Г. Моделювання процесів генерування матричних ключів / В.Г. Красиленко, Д.В. Нікітович // «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2018): Збірник тез доповідей IV Міжнародної науково-практичної конференції, 17-18 травня 2018 року.–Черкаси: ЧДТУ, 2018. – С. 32-35. – Режим доступу: https://chdtu.edu.ua/itont-2018/materiali-konferentsiji.

DEVELOPMENT AND SIMULATION ARRAY OF DEVICES BASED ON THE FPGA FOR PARALLEL CALCULATION OF NORMALIZED EQUIVALENCES OF THE REFERENCE FILTERS WITH THE CURRENT PROCESSED FRAGMENT OF THE IMAGE

Krasilenko V.G., Lazarev A.A., Nikitovich D.V.

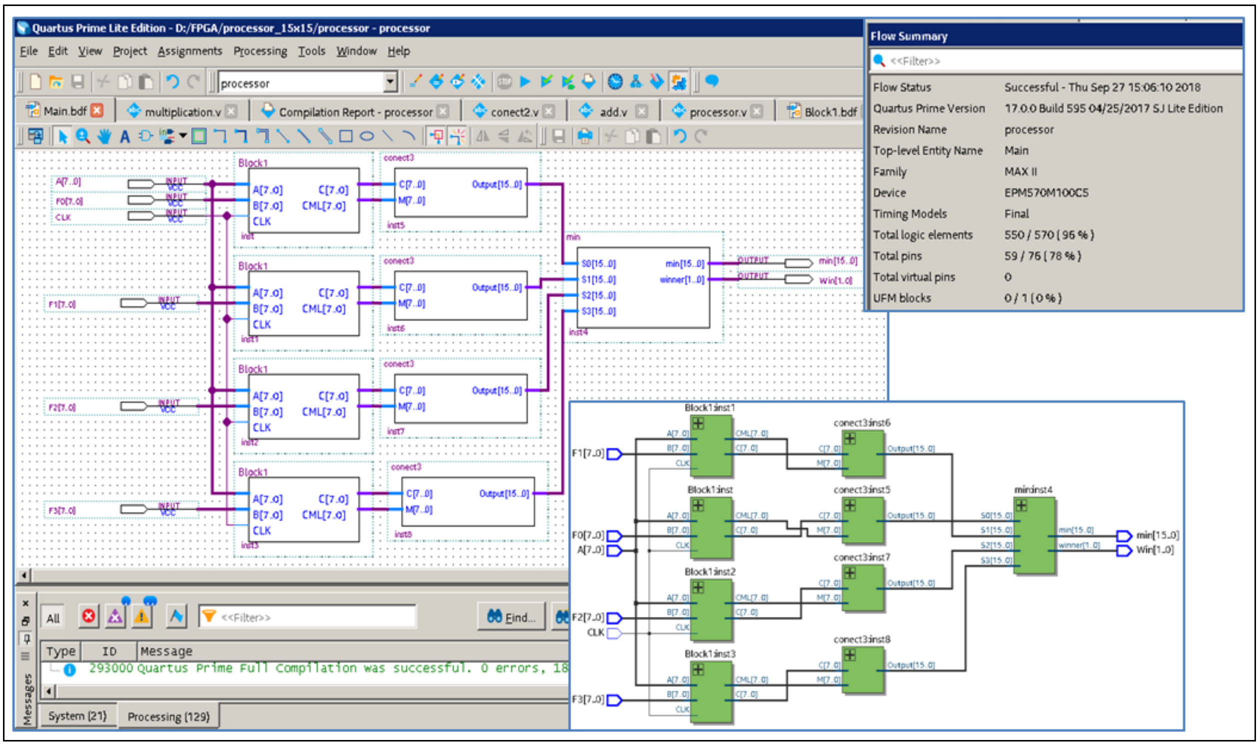
Vinnytsia National Technical University;

krasvg@i.ua

Introduction, purpose and objectives of the work. The basis of most known methods, algorithms and means, including models of neural networks (NNs), for the recognition and clustering of images in the biometric, machine vision systems is to compare two different images of the same object or its fragments [1, 2]. Discriminant measure of compared reference and current fragments is often a mutual two-dimensional correlation function. To improve accuracy and probability indicators with strong correlation obstacle-damaged image, it is desirable to use methods of combining images based on mutual equivalently two-dimensional spatial functions, nonlinear transformations of adaptive-correlation weighting [3]. The NNs are also widely used for modeling associative memory, pattern recognition. Models of equivalence (EM) of auto-associative memory (AAM) and hetero-associative memory (HAM) and multiport AAM (MAAM) and multiport hetero-associative memory (MHAM) were proposed in papers [4-9] and their modification in paper [10-13]. The simulation results of such models [5-7, 10-11] confirmed that the EM has advantages. These EM HAM studies have shown that these models allow the recognition of vectors with 4096 components and a significant percentage (up to 30%) of damage, at a network power that is 3 to 4 times higher than the number of neurons [10-11]. For of analysis and recognition should be solved the problem of clustering of different objects. This previous clustering allows organizing proper automated grouping processed data, to cluster analysis, to evaluate on the basis of many signs each cluster and improve subsequent learning procedures and classification. Knowing the significant advantages of EM for creating on their basis improved neural networks [5-8], MAAM, MHAM [7, 10, 11] it was suggested about the possibility of modifying EM and MHAM for parallel cluster analysis of images. Hardware implementations of these models, including equivalentors with spatial and time integration, spatially non-invariant and non-invariant models and their implementations were considered and examine in papers [1, 2, 7-9]. At the same time an urgent task is to study a more general, spatially invariant (SI) equivalence models (SI EMs) that is more invariant to spatial displacements and the possibilities of its application for image clustering [11, 13]. In addition these models are very closely related to operations of images convolution. And the latter are basic operations in the promising paradigms of convolutional neural networks (CNN) with deep leaning [14] and our self-learning equivalence-convolutional NNs (SLE_CNNs) [15-17]. The use of the aforementioned models and their effective implementations to create new bio-induced self-learning methods and extract patterns from the training set and processed images for multilevel image bitmaps was considered in [15]. In paper [16] we showed that the self-learning concept works with directly multi-level images without processing the bitmaps. Such equivalently neural paradigms are very perspective for processing, clustering, recognition, storing large size and strongly correlated and highly noised images. They are also very promising for solving the problem of creating machine uncontrolled learning. And since the basic operational functional nodes of EM are such vector-matrix or matrix-tensor procedures with continuous-logical operations as: normalized operations "equivalence", "non-equivalence", "auto-equivalence", we consider in this paper new conceptual approaches to the design of scale arrays of equivalentors (Es) or complementary non-equivalentors (N-Es) [17]. For all known convolutional neural networks, as for our SLE_CNNs, it is necessary to calculate the convolution of the current fragment of the image in each layer with a large number of templates that are used, which are a set of standards that are selected or formed during the learning process. But, as studies show, large images require a large number of filters to process images, and the size of the filters can also be large. Therefore, the problem of increasing the computing performance of SLE_CNN and their NCs as basic nodes is acute. It should be noted that the accuracy of calculations, especially for large filter sizes and a large dynamic range (8 bits) of halftone images, is necessary for making the right decisions for building maps and selecting neuron-winners. Therefore, intensified work aimed at creating specialized neural accelerators. Unlike most papers, where compute the function of comparing two 2D arrays and using the operations of multiplication and addition-accumulation, in our works we use functions of normalized equivalence in which there is no multiplication operation. But as our experiments show, equivalent models also allow the construction of equivalently convolutional structures and self-learning systems. Therefore, in this paper, using **our approaches to designing** of Es (N-Es) and in order to increase the accuracy of calculations, we consider the based **FPGA digital structures** of the calculators for processing 2D arrays. Our approach is based on the use of digital methods for implementing the required operations.

Presenting main material. For constructing of EMs the base binary operations of neurobiological (NBL) "equivalence" $(\sim)$ and "non-equivalence" $(\nsim)$ were used of such types, but here we use only one type of "non-equivalence": $a \overset{+}{\nsim} b = |a-b|$, where $a, b \in C_u = [0,1]$, $a = 1-a, b = 1-b$ and the equivalence function is: $eq = a \overset{+}{\sim} b = 1-|a-b|$ Normalized equivalence of two matrices $A = \{a_{ij}\}_{I\times J}$ and $B = \{b_{ij}\}_{I\times J} \in [0,1]^{I\times J}$ is determined in the following way: $A \underset{n}{\sim} B = \sum_{i=1}^{I}\sum_{j=1}^{J}\frac{(a_{ij} \sim b_{ij})}{I \cdot J}$, and correspondingly normalized non-equivalence: $A \underset{n}{\nsim} B = \sum_{i=1}^{I}\sum_{j=1}^{J}\frac{(a_{ij} \nsim b_{ij})}{I \times J}$. Operations $\left(\tilde{n}\right)$ and $\left(\overset{\nsim}{n}\right)$ are measures of similarity (equivalence) and difference (non-equivalence, distance) of matrices, which are connected with Hamming distance [5, 9]. Thus, by components operations $(\sim)$ and $(\nsim)$ of scalar NBL are generalized on matrix case and NBL logic becomes matrix NBL, i. e. (MNBL). Without loosing community we can consider carrying set (for scalar case) $C_u = [0,1]$ and $C_u^N = [0,1]^N$ (for N-dimension (vector, matrix) case). Normalized equivalence and nonequivalence are more general new complementary metrics in matrix space R. The variants of operations of equivalence and nonequivalence depend on different types of operations of t-norms and s-norms used in them and integrated operations of crossing and joining up in fuzzy logic. Depending on type, variants of equivalent algebra (EA) [5, 8, 9], as a new algebro-logical instrument for creation of equivalental theory of NNs on the basis of matrix NBL. The weighing coefficients of synapse connections matrix of equivalence models are determined through the normalized equivalence function, namely [4, 8]: $T_{ij}^{0} = \frac{1}{M}\sum_{m=1}^{M}(S_m^i \sim S_m^j) = f(\vec{S}^i, \vec{S}^j)$ or $T_{ij}^{\beta} = \frac{1}{M}\sum_{M=1}^{M}(\,|\beta| \sim S_m^i \sim S_m^j) = f(\vec{S}^i, \vec{S}^j)$ where β_m is vectors equivalence coefficient, and also it is the normalized equivalence f of vectors, that $\beta_m = f\left(\vec{X}, \overrightarrow{S^m}\right) = \frac{1}{N}\sum_{i=1}^{N}(S_i^m \sim X_i^{im})$. In addition, it is possible to show many other formulas, which are used in the equivalence models paradigm and based on calculations of the normalized equivalence or nonequivalence of vectors or matrices [5, 7, 8]. But in most general case optoelectronic complementally dual neuron (equivalentor/nonequivalentor), including the normalized components of vectors, has analog homopolar encoded components. For neural networks using the functions of normalized equivalences of matrices and tensors, generalized equivalence models have also been developed. They use spatially dependent normalized equivalence functions (SD_NEF) [1, 5], which are defined as: $\tilde{e}(A,B)=\dfrac{A\tilde{*}B}{I\times J}=\dfrac{1}{I\times J}\sum_{i=1}^{I}\sum_{j=1}^{J}(a_{\varsigma+i,\eta+j}\sim b_{y}),$ where $\tilde{\mathbf{e}}=\left[e_{\varsigma,\eta}\right]\in[0,1]^{(N-I+1)\times(M-J+1)}$, and symbol $(\tilde{*})$ indicates a spatial convolution, but with an element-wise operation of not multiplication, but "equivalence". Therefore interpretation method for spatially invariant case requires the calculation of spatial features convolution-type $\mathbf{E}^{m}=\mathbf{W}^{m}\overset{t}{\tilde{\otimes}}\mathbf{X}$, where $E_{k,l}^{m}=1-mean\left(\left|\overrightarrow{submatrix\left(\mathbf{X},k,k+r_{0}-1,l,l+r_{0}-1\right)-\mathbf{W}^{m}}\right|\right)$, nonlinear processing by the expression $EN_{k,l}^{m}=G\left(a,E_{k,l}^{m}\right)=0,5\left[1+\left(2E_{k,l}^{m}-1\right)^{a}\right]$ and comparing each other to determine the winners for indexing expressions: $MAX_{k,l}=\max_{index\,m}\left(EN_{k,l}^{m=0},EN_{k,l}^{m=1}...EN_{k,l}^{M-1}\right)$ and $EV_{k,l}^{m}=f_{nonlinear}^{activ}\left(EN_{k,l}^{m},MAX_{k,l}\right)$. From the above formulas it follows that it is necessary to calculate the average value of the component-wise difference of two matrices. Similarly, normalized nonequivalent functions for all filters are calculated. For implementation of new proposed subclass of neural computers, MAAM [4, 6], MHAM [7, 10], and SLE_CNN [15-17], we need certain new or modified known devices [17, 18] capable of calculating normalized spatial equivalence functions (NSEqFs) with the necessary speed and performance. Such specialized devices were previously called "image equivalentor" [4, 5, 8, 9, 12], (IE). In one particular case, the image equivalentor is essentially a doubled correlator or a doubled convolver. But there are many types of equivalents due to the large variety of equivalences. Thus, in the generalized SI EM [11, 13] and SLE_CNNs [15-16], for the input image S_{inp}, learning array-matrix A, which is a set of K*L reference images, it is necessary to calculate element-wise equivalence convolution and nonlinear transformations of elements in the first and second steps of the iteration algorithm. As metrics, we use generalized normalized vector equivalence functions. Research of the generalized SI EM confirmed their advantages and improved characteristics. Method of clustering based on the use of MAAM and MHAM and simultaneous calculation of the corresponding distances between all cluster neurons and all training vectors gives good convergence and high speed [10, 11, 13]. The model describes an iterative learning process that consists in computing the optimal set of weight vectors for a cluster neuron using the training matrix. We specify the number of templates and their size. The calculation of optimal patterns is formed by an iterative procedure based on extracting similarity and features in fragments of objects that are in the image or in their learning set. The works [15-16] show great promise of the SLE_CNNs for self-learning-recognition of images, including multilevel and color images. But for their work in real time, taking into account the large requirements for performance and the amount of calculations, it is necessary to have appropriate high-performance and energy-efficient image processors with parallel principles of operation and multi-inputs-multi-outputs, whose design was partially considered in papers [10, 11, 13].

Structure, listings, simulation results. The structural diagram of four parallel-working neuron-equivalents, which calculate the nonequivalence of the input current fragment with four reference standards (filters), is shown in Fig.1. Values of pixels of the filters (reference standards) in byte format are fed successively to the four inputs F1 [7..0], F2 [7..0], F3 [7..0], F4 [7..0], and pixel values of a selected current image fragment is fed sequentially to the fifth input A [7..0]. The sizes of filters and fragments can be arbitrary, but we designed for a size of 15x15, since for smaller sizes, the problem of speeding up the calculations is not so acute. The output of the scheme is the port Win [1..0], on which the winner's filter number is indicated, with which the current fragment has the greatest match, that is, their normalized nonequivalence is minimal. On the second output port min [15..0], the nonequivalence value between the winner and the current fragment is formed. This value is necessary for use in multilayer self-learning equivalence convolutional neural structures. Inside, the circuit contains four nodes Block1: inst1 - inst4 of the definition of the normalized nonequivalence of the compared fragment and filters, the interconnection circuit conect3: inst5 - inst8 and the winner selection node min: inst4. The circuit is implemented on Altera FPGA chip from the MAXII family EPM570M100C5. 550 logical elements (96% of 570 available) and 59 conclusions (78% of 76 available) were used. Clock frequency is 50 MHz, supply voltage 2.5V, power consumption 50mW.



Quartus Prime Lite Edition - D:/FPGA/processor_15x15/processor - processor

Flow Summary

Flow Status	Successful - Thu Sep 27 15:06:10 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	processor
Top-level Entity Name	Main
Family	MAX II
Device	EPM570M100C5
Timing Models	Final
Total logic elements	550 / 570 (96 %)
Total pins	59 / 76 (78 %)
Total virtual pins	0
UFM blocks	0 / 1 (0 %)

```
Type  ID     Message
      293000 Quartus Prime Full Compilation was successful. 0 errors, 18
```

Fig. 1. Block diagram of four parallel neuron-equivalentors

The processor: inst node finds the sum of the modules of the difference in pixel values of the fragment being processed and the filter. Rationing is implemented on three nodes: multiplication multiplacation: inst2, division conect: inst3, summation add:inst4. Thus, we find the value of the normalized nonequivalence. On the output through the node conect2: inst5, 8 higher and 8 lower order bits of the result are fed.

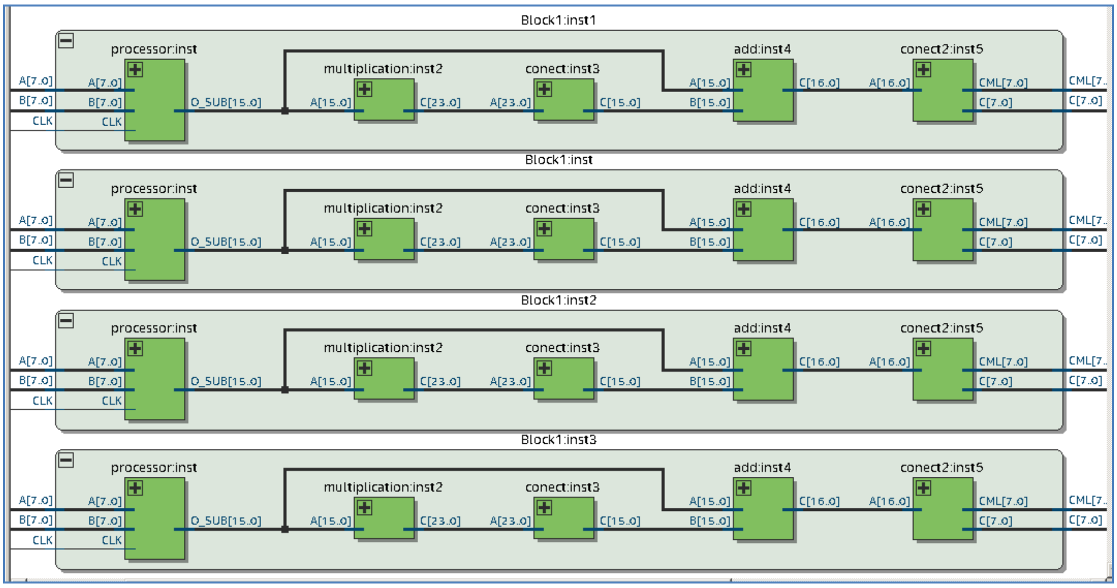


Fig. 2. Functional diagram of four non-equivalence calculation units

The functional diagram of the processor: inst node is shown in Fig. 3. The functional diagram of the min: inst4 winner selection node is shown in Fig. 4. These nodes are designed by the Verilog language. Listing of the design of these nodes is shown in Fig. 5.

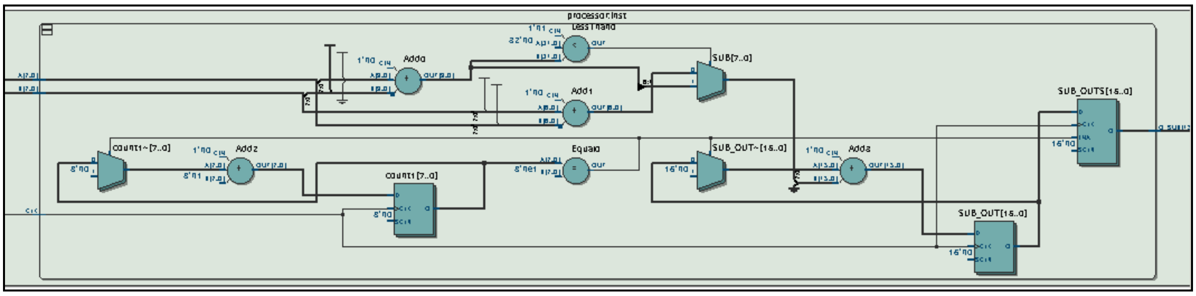


Fig. 3. Functional diagram of the processor: inst node

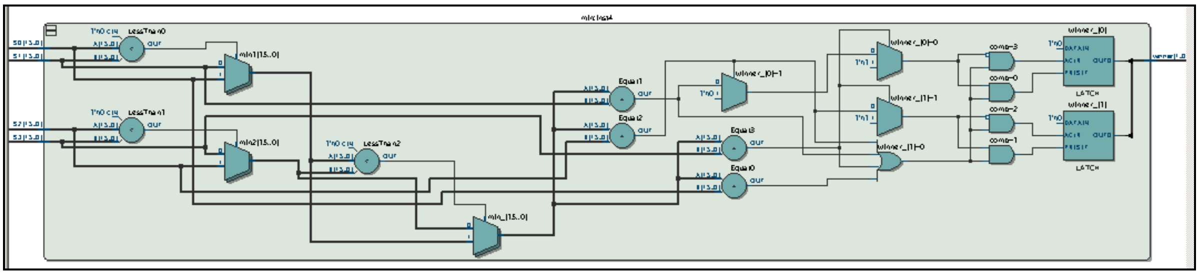
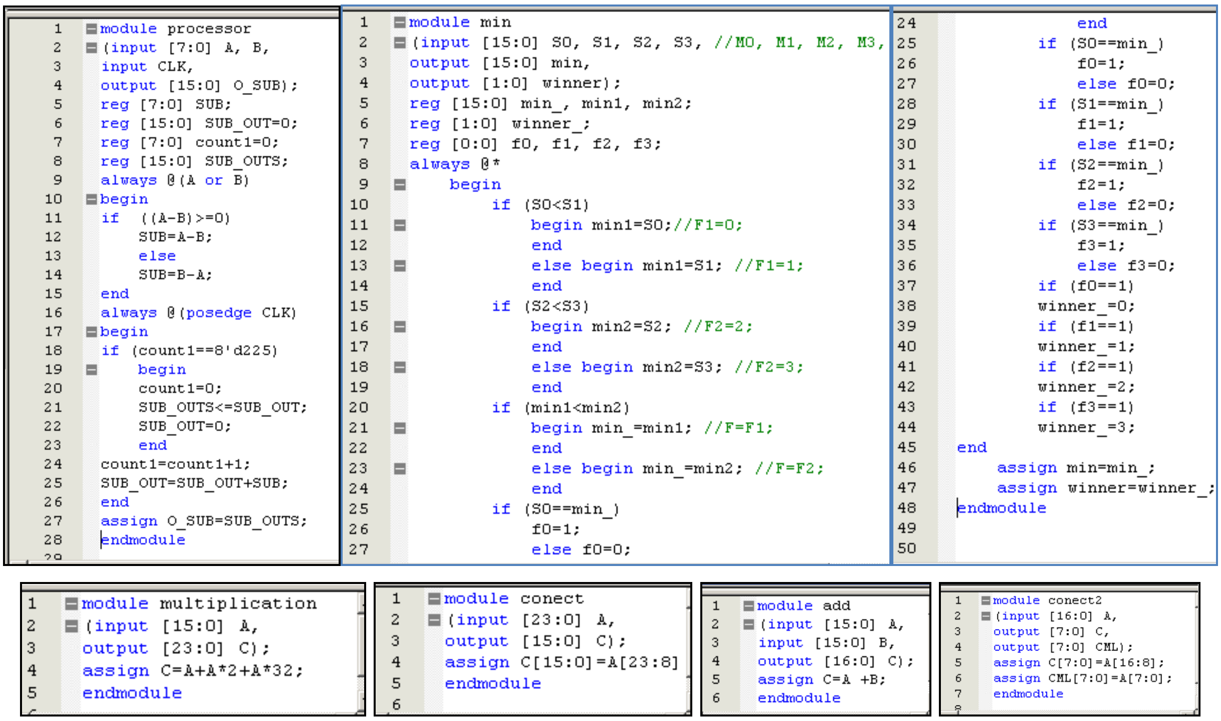


Fig. 4. Functional diagram of the min: inst4 winner selection node

Figures 6, 7 show the time diagrams of the work of the developed neuron-calculator. In fig. 6, and it can be seen that in the first period the values of all pixels of the current fragment are 25, and the values of all pixels are the same in each of the four filters and are 9, 8, 175 and 99, respectively in the first, second, third and fourth.



```
module processor
(input [7:0] A, B,
 input CLK,
 output [15:0] O_SUB);
 reg [7:0] SUB;
 reg [15:0] SUB_OUT=0;
 reg [7:0] count1=0;
 reg [15:0] SUB_OUTS;
 always @(A or B)
begin
 if  ((A-B)>=0)
     SUB=A-B;
     else
     SUB=B-A;
 end
 always @(posedge CLK)
begin
 if (count1==8'd225)
     begin
     count1=0;
     SUB_OUTS<=SUB_OUT;
     SUB_OUT=0;
     end
 count1=count1+1;
 SUB_OUT=SUB_OUT+SUB;
 end
 assign O_SUB=SUB_OUTS;
 endmodule
```

```
module min
(input [15:0] S0, S1, S2, S3, //M0, M1, M2, M3,
 output [15:0] min,
 output [1:0] winner);
 reg [15:0] min_, min1, min2;
 reg [1:0] winner_;
 reg [0:0] f0, f1, f2, f3;
 always @*
     begin
           if (S0<S1)
                begin min1=S0;//F1=0;
                end
                else begin min1=S1; //F1=1;
                end
           if (S2<S3)
                begin min2=S2; //F2=2;
                end
                else begin min2=S3; //F2=3;
                end
           if (min1<min2)
                begin min_=min1; //F=F1;
                end
                else begin min_=min2; //F=F2;
                end
           if (S0==min_)
                f0=1;
                else f0=0;
           if (S1==min_)
                f1=1;
                else f1=0;
           if (S2==min_)
                f2=1;
                else f2=0;
           if (S3==min_)
                f3=1;
                else f3=0;
           if (f0==1)
           winner_=0;
           if (f1==1)
           winner_=1;
           if (f2==1)
           winner_=2;
           if (f3==1)
           winner_=3;
end
     assign min=min_;
     assign winner=winner_;
endmodule
```

```
module multiplication
(input [15:0] A,
 output [23:0] C);
 assign C=A+A*2+A*32;
 endmodule
```

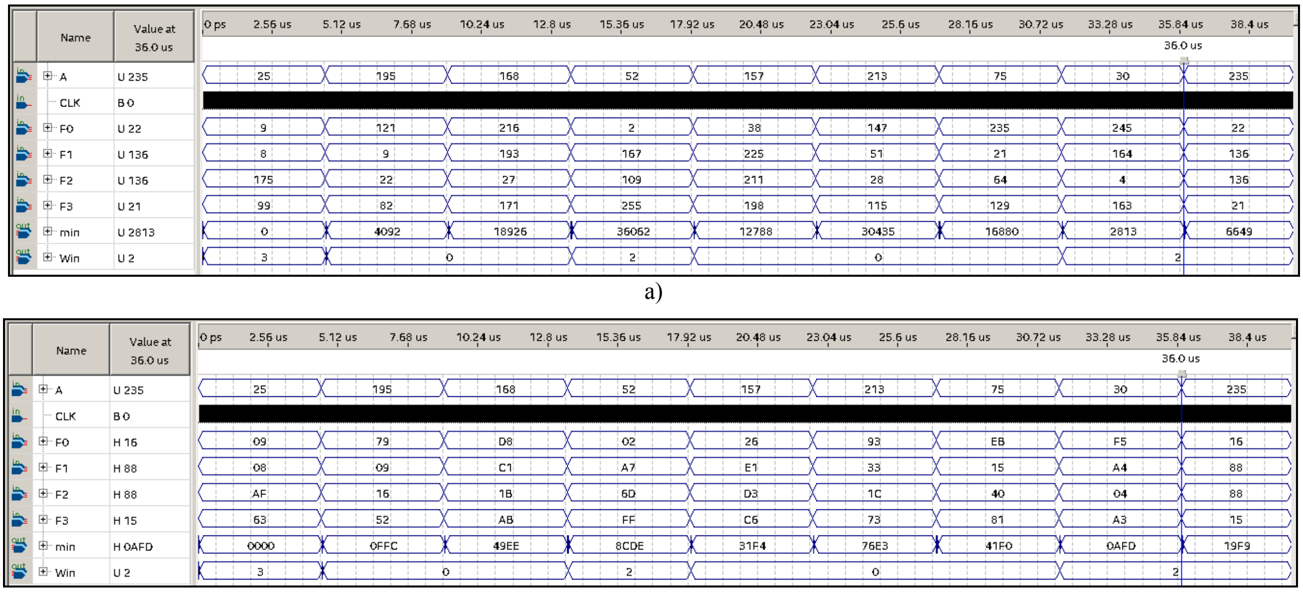
```
module conect
(input [23:0] A,
 output [15:0] C);
 assign C[15:0]=A[23:8]
 endmodule
```

```
module add
(input [15:0] A,
 input [15:0] B,
 output [16:0] C);
 assign C=A +B;
 endmodule
```

```
module conect2
(input [16:0] A,
 output [7:0] C,
 output [7:0] CML);
 assign C[7:0]=A[16:8];
 assign CML[7:0]=A[7:0];
 endmodule
```

Fig. 5. Unit design listings: processor (left); min winner selection node (center and right), multiplication node and others

Therefore, the modulus of the difference in pixel values is the same within each compared pair (filter fragment) for all pixels and will be equal to: 16, 17, 150, 74, respectively. After summing up for 225 cycles of these modules for each filter, the sum of modules of element-wise (pixel-by-pixel) differences is formed, which needs to be normalized by dividing by 225 (the number of pixels). We propose replacing the complex division operation by 225 by dividing the increased amount of modules with the calculated required coefficient by 256. For example, to divide by 225 we use the multiplication of the sum of the modules by 3 and the addition of the sum of the modules multiplied by 32 (actually shifted by 5 digits). The result obtained is divided into two parts, in which the upper 8 digits correspond to the integer part of the result, and the younger ones - the fractional part of the result. Therefore, the result of processing, which appears at the output in the second cycle, shows that for the winner filter 0, for which the minimum pixel difference was 16 and was the smallest, the result is 16 bit binary code. Since in time diagrams, displaying a code of this length is inconvenient, the result is represented as a decimal number, actually multiplied by 256. Therefore, we get the number 4092, which can be seen in Fig. 6a, which, after dividing by 256, should be taken as the number 15.98. In Fig. 6b, to represent the integer and fractional parts when using a hexadecimal coding system, we see the displayed result 0FFC, which has the integer part 0F (in decimal form 15) and the fractional part FC / 256, which corresponds to 0.98 in decimal form. Similarly, the neuron-calculator works in the following periods. Fig. 7 shows the timing diagrams of the functioning of the circuit for the case when the pixel values of the filter are constant, and the pixel values of the current fragment are arbitrary. The results also confirm the normal operation and computations with digital accuracy of the required nonequivalence values to make the right decision regarding the choice of the winner.



Name	Value at 36.0 us									
A	U 235	25	195	168	52	157	213	75	30	235
CLK	B 0									
F0	U 22	9	121	216	2	38	147	235	245	22
F1	U 136	8	9	193	167	225	51	21	164	136
F2	U 136	175	22	27	109	211	28	64	4	136
F3	U 21	99	82	171	255	198	115	129	163	21
min	U 2813	0	4092	18926	36062	12788	30435	16880	2813	6649
Win	U 2	3	0		2	0			2	

a)

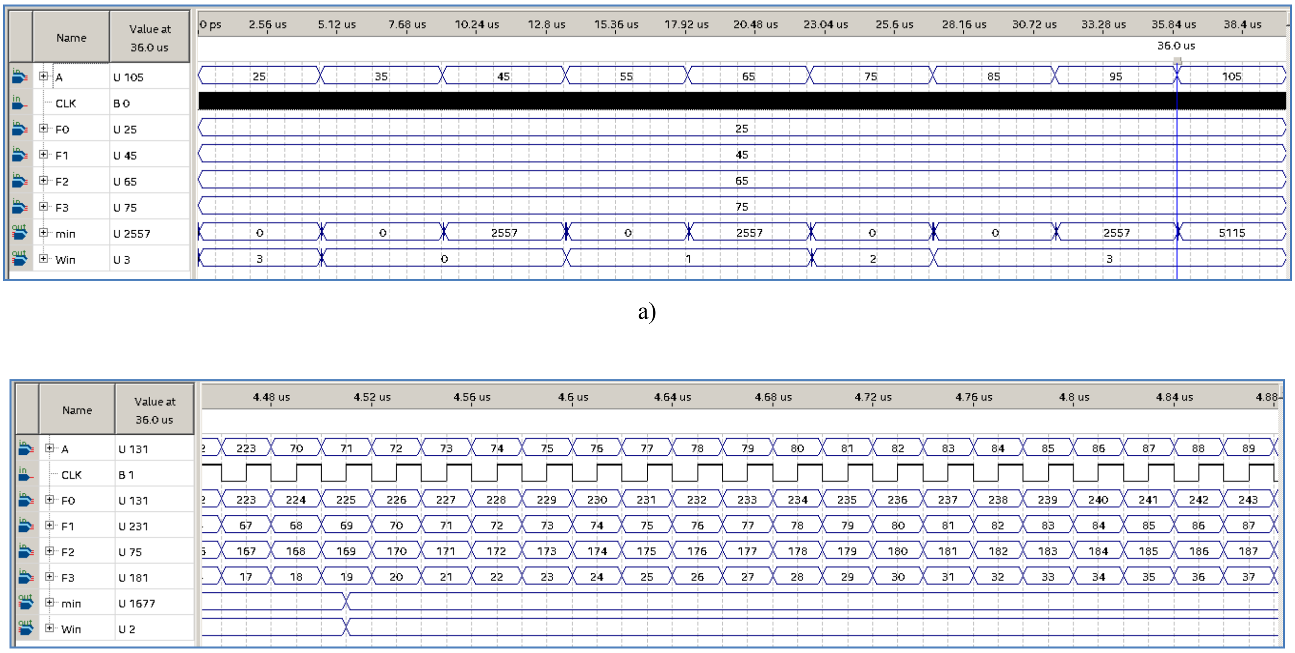
Name	Value at 36.0 us									
A	U 235	25	195	168	52	157	213	75	30	235
CLK	B 0									
F0	H 16	09	79	D8	02	26	93	EB	F5	16
F1	H 88	08	09	C1	A7	E1	33	15	A4	88
F2	H 88	AF	16	1B	6D	D3	1C	40	04	88
F3	H 15	63	52	AB	FF	C6	73	81	A3	15
min	H 0AFD	0000	0FFC	49EE	8CDE	31F4	76E3	41F0	0AFD	19F9
Win	U 2	3	0		2	0			2	

b)

Fig. 6. The timing diagrams of the simulation results of the circuit of four parallel-working neuron-equivalentors for constant values of signals inside the filters and a fragment: a) the values of the signals are shown in decimal code; b) shows the signal values in the hex code; the comparison time of the current fragment and the reference is 4.5 μs

Given the limitations on the size of the article, here we do not show the design and simulation results for other filter sizes (3x3, 5x5, 7x7, 9x9, etc.). Processing time is proportional to the total number of pixels in the filter. We also investigated and modeled the neuron-calculator node using a more powerful Altera FPGA chip EP3C16F484 Cyclone III family, which has 15408 logical elements, RAM 516096 bits, 346 I/O pins. Taking into account the complexity of the one neuron-calculator and the resources of this chip, it would be possible to implement 64 neuron-calculators, but given the availability of available pins, it is possible to implement only 32 neuron-calculators. For performance evaluation, for filter size 15x15 = 225, clock frequency 225MHz, (4.44ns clock cycle), total processing time 225x4.44 = 1µs, we define the number of operations performed per cycle as 2.22 compare-subtract-accumulate operations (lower modest estimate) that gives for each NCs 500 operations during filter processing, and 32 channels operating in parallel, respectively, 500x32 = 16000 operations per 1µs. Thus, the performance of the proposed node will be $16x10^9$ operations per second, and its energy efficiency, taking into account the power consumption of 0.2 watts, is estimated at $8x10^{10}$ operations per second per watt.

Conclusions. We show the results of design and modeling the proposed new FPGA-implementations of neuron-calculators as hardware accelerators of self-learning equivalently-convolutional neural networks (SLE_CNNs). Simulation results show that processing time in such circuits does not exceed units of microseconds, and for some variants 500 nanoseconds. Circuits are simple, have low supply voltage (2.5V), low power consumption (50mW), digital accuracy, integrated construction, satisfy the problem of interconnections and cascading. Signals at the output of such neurons can be both can be digital with increased accuracy and also with two additional complement outputs, that indicate winning neurons and their activity. The presence of digital outputs encoding the normalized equivalence of the matrices being compared and represented by the whole and fractional parts allows determining the neuron-winner with increased accuracy and displaying its intensity. Thus, a simplified matching of neuron-calculators node with other digital nodes is implemented, and a simple selection of the corresponding memory elements into which the calculated results are written, which form the maps for their subsequent convolutions with the image.



a)

b)

Fig. 7. Time diagrams of simulation results for the circuit of four parallel-working neuron-equivalents: a) for constant values of the signals of the standards and different signals of the fragments at the same intensity of the pixel values; b) for different values of the signals within the standards and the fragment in cycles within the processing cycle. The comparison time of the current fragment and the reference is 4.5 μs. Time of one clock cycle is 20ns

Research confirms the possibility of creating 32 components array of neuron-calculators on 8 Altera FPGA chips EPM570M100C5 MAXII family. And in each chip it is possible to implement four NCs for a window size (filter) of 15x15. And for smaller sizes, their number can be increased, but the efficiency of using the resources of the scheme is limited by the number of pins. The processing time of the current image fragment does not exceed 5μs. Calculations show that when using an Altera FPGA chip EP3C16F484 Cyclone III family, it is possible to implement 32 neuron-calculators in the one chip. For the chip for 2.5V and clock frequency 200MHz the power consumption will be at the level of 200mW, and the calculation time for filters with a size of 15x15 will be 1μs. The performance of such a node will be 16x10^{9} operations per second, and its energy efficiency 8x10^{10} operations per second per watt. The obtained results confirm the correctness of the chosen concept and the possibility of creating NCs and MIMO structures on their basis.

References:

1. Krasilenko V.G., Nikolskyy A.I., Bozniak Y.A., "Recognition algorithms of multilevel images of multicharacter identification objects based on nonlinear equivalent metrics and analysis of experimental data," Proc. of SPIE Vol. 4731, pp. 154-163 (2002).

2. Krasilenko V.G., Nikolskyy A.I., Bozniak Y.A., "Recognition Algorithms of Images of Multi-Character Identification Objects Based jn Nonlinear Equivalent Metrics and Analysis of Experimental Dat Using Designed Software," Proceedings of Eleventh All-Ukrainian International Conference, pp. 107-110 (2012).

3. Krasilenko V.G., Boznyak Y.A., Berozov G.N., "Modelling and comparative analysis of correlation and mutual alignment equivalent images," Science and learning process: scientific and methodical. Proceedings Scientific Conference of the VSEI Entrepreneurship University "Ukraine", pp. 68-70 (2009).

4. Krasilenko V.G., Magas A.T., "Multiport optical associative memory based on matrix-matrix equivalentors," Proc. of SPIE Vol. 3055, pp. 137 - 146 (1997).

5. Krasilenko V.G., Saletsky F.M., Yatskovsky V.I., Konate K. "Continuous logic equivalence models of Hamming neural network architectures with adaptive-correlated weighting," Proc. of SPIE Vol. 3402, pp. 398-408 (1998).

6. Krasilenko V.G., Nikolskyy A.I., Yatskovskaya R.A., Yatskovsky V.I. "The concept models and implementations of multiport neural net associative memory for 2D patterns," Proc. of SPIE Vol. 8055, 80550T (2011).

7. Krasilenko V.G., Lazarev A., Grabovlyak S. "Design and simulation of a multiport neural network heteroassociative memory for optical pattern recognitions," Proc. of SPIE Vol. 8398, 83980N-1 (2012).

8. Krasilenko V.G., Nikolskyy A. I. "Optical pattern recognition algorithms based on neural-logic equivalent models and demonstration of their prospects and possible implementations," Proc. of SPIE Vol. 4387, pp. 247-260 (2001).

9. Krasilenko V.G., Kolesnitsky O.K., Boguhvalsky A.K. "Application of non-linear correlation functions and equivalence models in advanced neuronets," Proc. of SPIE Vol. 3317, pp. 211-223 (1997).

10. Krasilenko V.G. , Alexander A. Lazarev, Sveta K. Grabovlyak, Diana V. Nikitovich, "Using a multi-port architecture of neural-net associative memory based on the equivalency paradigm for parallel cluster image analysis and self-learning," Proceedings of SPIE Vol. 8662, 86620S (2013).

11. Krasilenko V.G., Lazarev A.A., Nikitovich D.V. "Experimental research of methods for clustering and selecting image fragments using spatial invariant equivalent models," Proceedings of SPIE Vol. 9286, 928650 (2014).

12. Krasilenko V.G., Nikolskyy A.I., Flavitskaya, J.A., "The Structures of Optical Neural Nets Based on New Matrix_Tensor Equivalently Models (MTEMs) and Results of Modeling," Optical Memory and Neural Networks (Information Optics) Vol. 19 (1), 31–38 (2010).

13. Krasilenko V.G., Nikitovich D.V. "Researching of clustering methods for selecting and grouping similar patches using two-dimensional nonlinear space-invariant models and functions of normalized equivalence," VII Ukrainian-Polish scientific and practical conference Electronics and information technologies (ELIT-2015). – Lviv: Ivan Franko National University of Lviv, pp. 129-134 (2015).

14. Shafiee A., et al. "ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars," 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), Seoul, 2016, pp. 14-26. doi: 10.1109/ISCA.2016.12.

15. Krasilenko V.G., Lazarev A.A., Nikitovich D.V. "Modeling and possible implementation of self-learning equivalence-convolutional neural structures for auto-

encoding-decoding and clusterization of images," Proceedings of SPIE Vol. 10453, 104532N (2017).

16. Krasilenko V.G., Lazarev A.A., Nikitovich D.V. "Modeling of biologically motivated self-learning equivalent-convolutional recurrent-multilayer neural structures (BLM_SL_EC_RMNS) for image fragments clustering and recognition," Proc. SPIE 10609, MIPPR 2017: Pattern Recognition and Computer Vision, 106091D (8 March 2018); doi: 10.1117/12.2285797; https://doi.org/10.1117/12.2285797 .

17. Krasilenko V.G., Lazarev A.A., Nikitovich D.V., "Design and simulation of optoelectronic neuron equivalentors as hardware accelerators of self-learning equivalent convolutional neural structures (SLECNS)," Proceedings of SPIE Vol. 10689, 106890C (2018).

18. Krasilenko V.G., Lazarєv O.O. Sheremeta O.P. Designing of array neuron-equivalentors with a quasi-universal activation function for creating a self-learning equivalent- convolutional neural structures, Information Processing Systems, 2019, Vol. 1(156), pp. 82-91. https://doi.org/10.30748/soi.2019.156.11

СЕРЕДОВИЩЕ РОЗРОБКИ RAD STUDIO ДЛЯ СТВОРЕННЯ ВЛАСНИХ ПРОЕКТІВ

Лазурчак Л.В.

Дрогобицький державний педагогічний університет імені Івана Франка

Homyshak59@ukr.net

Інтегроване середовище розробки RAD Studio дозволяє писати і редагувати код швидко і раціонально, завдяки використанню сучасних засобів об'єктно-орієнтованого програмування в поєднанні з надійним програмним забезпеченням. У ньому передбачено автоматичне доповнення коду на основі бібліотек, що використовуються.