The object of research in this work is ensemble classifiers with stacking, intended for the classification of objects in images with the presence of small sets of labeled data for training. To improve the quality of classification at the first stage of such a classifier, it is necessary to place more primary classifiers that differ in heterogeneous structured processing. However, the number of known neural networks with appropriate characteristics is limited. One approach to solving this problem is to build analogs of known neural networks that make classification errors on other images compared to the base network. The disadvantage of the known methods for constructing such analogs is the need to perform additional floating-point operations. The current paper proposes and investigates a new method to form analogs through random cyclic shifts of rows or columns of input images. This has made it possible to completely eliminate additional floating-point operations. The effectiveness of using this method is explained by the structured processing of input images in basic neural networks. The use of analogs obtained by the proposed method does not impose additional restrictions in practice. This is because the heterogeneity of structured processing in basic neural networks is a typical requirement for them in an ensemble classifier with stacking.

The simulation for the CIFAR-10 data set demonstrated that the proposed technique for constructing analogs allows for a comparative quality of classification by the ensemble classifier. Using MLP-Mixer analogs provided an improvement of 4.6 %, and CCT analogs – 5.9 %

Keywords: multilayer perceptron, neural network, ensemble classifier, weighting coefficients, classification of objects in images

UDC 681.3.07: 004.8
DOI: 10.15587/1729-4061.2024.299734

REDUCING THE VOLUME OF COMPUTATIONS WHEN BUILDING ANALOGS OF NEURAL NETWORKS FOR THE FIRST STAGE OF AN ENSEMBLE CLASSIFIER WITH STACKING

Oleg Galchonkov
Corresponding author
PhD, Associate Professor*
E-mail: o.n.galchenkov@gmail.com
Oleksii Baranov
Software Engineer
Oracle Corporation
Oracle World Headquarters
Oracle Way 2300, Austin, Texas, United States, 78741
Petr Chervonenko
PhD, Associate Professor*
Oksana Babilunga
PhD, Associate Professor*
*Department of Information Systems
Institute of Computer Systems
Odesa Polytechnic National University
Shevchenko ave., 1, Odesa, Ukraine, 65044

Received date 21.01.2024
Accepted date 25.03.2024
Published date 30.04.2024

How to Cite: *Galchonkov, O., Baranov, O., Chervonenko, P., Babilunga, O. (2024). Reducing the volume of computations when building analogue of neural networks for the first stage of an ensemble classifier with stacking. Eastern-European Journal of Enterprise Technologies, 2 (9 (128)), 27–35. doi: https://doi.org/10.15587/1729-4061.2024.299734*

1. Introduction

Neural networks for categorizing objects in images are becoming increasingly widespread. However, new application areas are characterized primarily by a small amount of initial labeled data for training neural networks. For example, the BloodMNIST microscope image set of blood samples [1] contains a total of 17,092 images. A similar dataset for blood testing for malaria parasites [2] contains 27,588 images. The set of CT images for training neural networks to classify COVID-19 contains 6752 images of the lungs of 4154 patients [3, 4]. At the same time, the main trend in the development of neural networks for classifying objects in images is the construction of very large networks with tens and hundreds of millions of weight coefficients, which are trained on very large data sets. For example, a neural network with the ViT-Huge (VisionTransformer) architecture [5] contains more than 632 million customizable weighting coefficients and was trained on the JFT-300M dataset (internal Google dataset) [6], containing 303 million images.

Even if such a neural network is retrained within the framework of transfer learning on a small data set, it will not show high performance with a very large volume of calculations to perform classification. Therefore, for small data sets, special neural networks are designed with a relatively small amount of computation, acceptable in practice. Such networks must be fully trained from initial conditions to steady state on these small data sets. When designing such neural networks, various approaches are used that can be combined and complement each other. One of the most effective techniques for combining seems to be the construction of ensemble classifiers with stacking [7].

Such classifiers have two stages. At the first stage, several classifiers work in parallel; the output signals of these classifiers are combined by an additional neural network at the second stage. This ensures a high degree of parallelizability of calculations. In addition, classifiers of the first stage

can be trained separately, and then used to train the second stage. The basis for the effectiveness of ensemble classifiers with stacking is the use of heterogeneous neural networks with structured processing at the first stage. With approximately the same classification quality, such neural networks make errors on different images. However, at present, the number of neural networks suitable for the first stage is limited [8]. This places a limit on the resulting improvement in classification quality of the entire ensemble classifier. To increase the number of neural networks at the first stage, it is necessary either to develop neural networks with a new processing architecture, or to build analogs of already known networks [9].

Such analogs should have the same processing structure as the basic neural networks but make errors on different sets of images. Work [9] describes one of the methods for constructing analogs by rotating the original images. Due to the structured nature of image processing in the original neural network, the analog makes classification errors on another set of images. However, rotating images requires performing four floating-point multiplications and two floating-point addition operations, plus an integer operation for each pixel in the image. Therefore, it seems relevant to devise methods for obtaining analogs that require less computation.

2. Literature review and problem statement

The main types of neural network architectures for object recognition in images are convolutional neural networks (CNN), transformer-based architectures, and multilayer perceptron (MLP)-based architectures. Convolutional neural networks show good results [10, 11], but their processing is not structured; the entire image is processed at once. Therefore, constructing analogs based on them for use in the first stage of an ensemble classifier does not lead to an improvement in the quality of classification [9]. Transformer-based architectures were developed later than convolutional neural networks, are characterized by a high degree of structure and show better results [5]. However, most of these architectures are designed to be implemented on supercomputers and are trained on very large data sets. A study of the effectiveness of the classical architecture based on a transformer [5] showed that on small data sets it provides high quality classification of objects in images. However, the volume of calculations for such a transformer is many times greater than the volume of calculations for an ensemble classifier that has the same classification quality [8]. Therefore, to ensure a high ratio between the quality of classification and the volume of calculations, special modifications of transformers are built. The main idea of such modifications is not to process the entire image at once but to do it in parts and use one or another mechanism for mixing the processed fragments. Processing in parts provides structure to the architecture and reduces the amount of computation. Shuffling image fragments makes it possible to take into account the relationships between all image fragments.

In [12], it was proposed to use convolution and self-attention in simplified modifications of the transformer. That made it possible to take into account both local and global dependences between fragments of the input image with a relatively small volume of computation. In work [13] (CCT – Compact Convolutional Transformer), the main modification of the transformer consisted of additional processing of each fragment of the input image by a convolutional neural network. That made it possible to radically reduce the volume of calculations while maintaining high classification quality. Additional processing of image fragments using CNN provides significant convenience when working with new data sets, allowing one to achieve high quality classification [14]. The disadvantage of the architectures proposed in [12–14] is the reduction in the structure of processing due to the use of CNNs. This makes it difficult to obtain effective analogs. In work [15] (EANet – external attention MLP), by analogy with a transformer, modification was carried out by replacing standard attention modules that require a large volume of computation with external attention modules and the use of additional MLP modules. Simplification of the transformer architecture and the introduction of the Fourier transform into it for mixing image fragments also made it possible to significantly improve the ratio of the quality of classification of objects in images and the volume of calculations (Fnet – Transformer encoder with a standard unparameterized Fourier Transform) [16]. However, building a large number of analogs based on this approach does not seem promising. In work [17] (SwinTr – Hierarchical Vision Transformer using Shifted Windows), the transformer architecture was simplified by calculating self-attention not over the entire image but only inside the windows. Shuffling is achieved due to the fact that these windows partially overlap. In principle, based on this approach, it is possible to build a number of analogs using different combinations of windows, but this requires separate research.

The classifier architecture was radically simplified and structured in [18], in which MLP-Mixer (MLP Architecture for Vision) was proposed. The input image is divided into a lattice of fragments. The architecture involves a sequential connection of modules, each of which contains two groups of single-layer perceptrons. The first group processes each of the columns of the fragment lattice separately. The second group processes the results of the first group for each of the rows separately. A complete lattice of processed image fragments is transmitted between modules. At the output of the classifier, as in all architectures, there is a multilayer perceptron that generates the resulting output signals of the classifier. Work [19] proposed the gMLP (MLP with gating) architecture, which uses a similar approach, but instead of columns and rows, channel projections and spatial projections with multiplicative gating are used. The disadvantage of the architectures proposed in [18, 19] is the minimalism of construction, which does not imply internal variations that would allow the construction of their analogs. To improve the mechanism for mixing image fragments used in MLP-Mixer, work [20] introduced axis shift modules, which shift fragments vertically and horizontally inside fragment modules. As processing moves closer from the source image to the output vector, the size of the fragment modules decreases. This allows for better handling of local dependences. This approach to building an architecture potentially makes it possible to obtain different variations in block sizes and types of shifts for constructing analogs. However, it requires additional research. In works [18–20], classifier architectures use static mechanisms for mixing image fragments, which does not take into account the contents of these fragments. In [21], the DynaMixer architecture (Vision MLP Architecture with Dynamic Mixing) with dynamic integration of information using the dynamic formation of mixing matrices is proposed. However, de-

spite the additional reduction in dimensions, the formation of mixing matrices increases the overall volume of calculations. Dynamic accounting of information contained in input images is also implemented in the DSM (Dynamic Spectrum Mixer for Visual Recognition) architecture [22]. The input image is converted into the frequency domain using a two-dimensional cosine transform. Only powerful components are retained for further processing, the rest are discarded. This provides a significant reduction in the volume of computation. Weighting coefficients are formed by a special dynamic generator, which takes into account which frequency components are left for the processed image. It should be noted that architectures with dynamic formation of the processing area make it difficult to build analogs of such classifiers since there is no fixed processing structure. In [23], the MDMLP (multi-dimensional MLP) architecture was proposed, where a number of modifications of MLP-Mixer were suggested. First, the original image is split into overlapping fragments. Secondly, in addition to mixing by height and width, mixing by channels has been introduced. Thirdly, an attention module built on the basis of MLP was introduced into the architecture. All that made it possible to increase the quality of classification on small data sets compared to MLP-Mixer. At the same time, additional mechanisms for mixing and overlapping fragments reduce the structure of processing, which makes these architectures less efficient when used as the first stage of an ensemble classifier with stacking. In [24], the column and row processing structure used in MLP-Mixer is replaced with processing of helical structures with different helix sizes. That made it possible to increase the quality of categorization. However, this method of mixing image fragments reduces the structure of processing, which makes it difficult to build effective analogs that make errors on images other than the same ones as the underlying network. Combining the idea of orthogonalizing convolutional layers by transferring processing to the frequency domain with mixing fragments vertically and horizontally in MLP-Mixer [18] made it possible to design an effective SplitMixer architecture (a simple and lightweight isotropic MLP-like architecture) [25]. It has a high ratio of classification accuracy to the required volume of calculations. However, the use of convolutional layers reduces the structure of the architecture and complicates the construction of analogs for the first stage of the ensemble classifier. The simplest technique for mixing image fragments is proposed in the Spatial-Shift MLP architecture [26]. Fragments are combined into groups. The groups are superimposed on each other in a layered structure and then in this structure the layers are shifted along two axes relative to each other. Potentially, the construction of analogs of such an architecture is possible due to various options for constructing groups and options for shifts. However, communication between groups requires additional floating-point operations. Providing a variety of classification results with analogs will lead to an increase in the volume of calculations.

Most of the considered approaches to constructing new architectures involve performing a certain volume of floating-point operations, which will lead not to a decrease but to an increase in the volume of calculations when designing analogs of neural networks. The simplest and most studied solution was proposed in [9]. Analogs are obtained by rotating the input images at different angles. Due to the structured processing in basic neural networks, analogs make errors in other images. Joint processing of the results of the first-stage classifiers at the second stage of the ensemble classifier allows one to increase the resulting classification quality. However, rotating images requires a significant volume of floating-point operations. It seems promising to study the possibility to design analogs by using only shifts, which do not require floating point operations at all.

3. The aim and objectives of the study

The goal of our work is to devise a technique for constructing analogs of neural networks applying shift operations for use at the first stage of an ensemble classifier with stacking in the task of classifying objects in images. This will make it possible to design analogs of basic neural networks without additional floating-point operations. Adding such analogs at the first stage will improve the resulting classification quality of the ensemble classifier.

To achieve the goal, the following tasks were set:

– to determine the number of matching errors between analogs and the basic neural network using the example of MLP-Mixer and CCT;

– to study the dependence of the resulting classification quality of an ensemble classifier with stacking on the number of neural network analogs using the example of MLP-Mixer, CCT and the CIFAR-10 data set for different shift parameters;

– to investigate the effectiveness of using analogs obtained using image shifts at the first stage of the ensemble classifier, in comparison with analogs obtained using image rotations.

4. The study materials and methods

The object of research in this work is ensemble classifiers with stacking, designed to classify objects in images in the presence of small sets of labeled data for training. The subject of research is analogs of basic neural networks of the first stage of an ensemble classifier. The CIFAR-10 dataset [27], containing 50,000 color images for training and 10,000 for testing, was taken as a data set. Image size is 32x32 pixels. The images belong to 10 classes: airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, trucks. Examples of images are shown in Fig. 1 [28].

The generalized architecture of the ensemble classifier is shown in Fig. 2 [9]. It consists of two stages. At the first stage, the input image is processed in parallel by M basic neural networks and L their analogs. Analogs differ from basic neural networks only in the presence of transformations q_1, ..., q_L at their input. The output of each neural network is a support vector for all k classes of objects that may be in the input image x:

$$P_i(x) = \left(p_{i1}(x), p_{i2}(x), ..., p_{ik}(x)\right), \quad (1)$$

where i is the classifier number,

$p_{ij}(x)$ – support by the i-th classifier that the object in the image x belongs to the j-th class.

At the second stage, each of the support vectors is normalized separately and fed to the multilayer perceptron (MLP):

$$a_i(x) = \max_j \left\{\left|p_{ij}(x)\right|\right\}, \quad (2)$$

$$p_{ij}^n(x) = p_{ij}(x) / a_i(x), \quad (3)$$

where i is the classifier number, i=1,...,M+L.

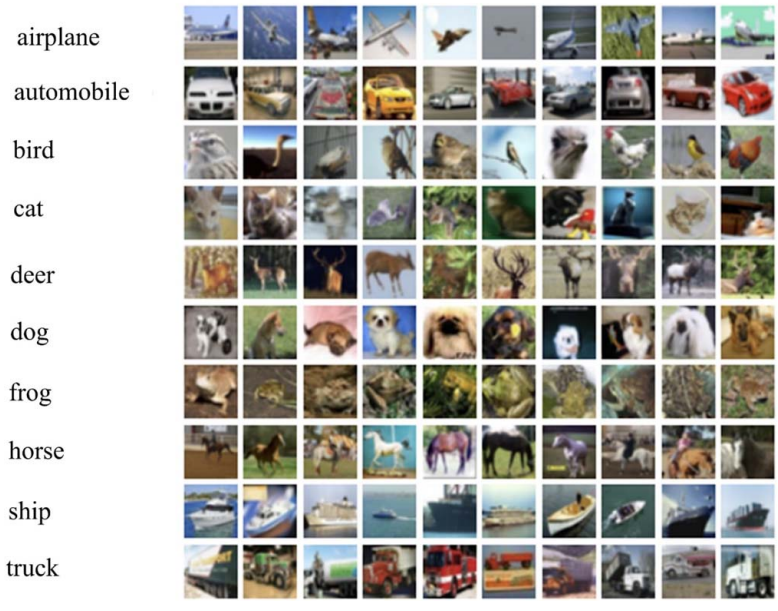

Fig. 1. Sample images from the CIFAR-10 set

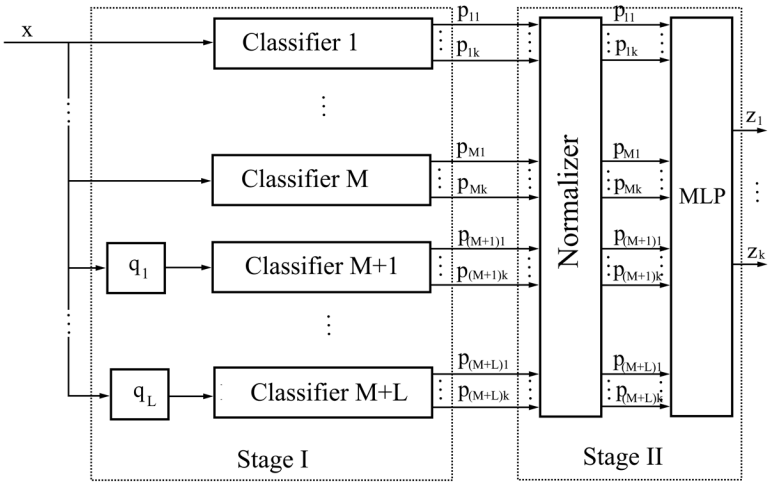

Fig. 2. A generalized architecture of an ensemble classifier with staking

Multilayer perceptron (MLP) [29] contains three layers:

– input layer of dimensionally $k\times(M+L)$;

– hidden layer of dimensionally $(k-1)\times(M+L)$, activation function Relu;

– output layer of dimensionality k, softmax activation function.

The class of the object in the current image x is determined by the channel number at the MLP output with the maximum value:

$$class(x)=\arg\left\{\max_j\left(z_j(x)\right)\right\}, \quad (4)$$

where $z_j(x)$ is the signal at the j-th output of MLP for the input image x, $j=1,...,k$.

Transformations q are cyclic shifts of rows or columns of pixels in the input image in accordance with the shift vectors:

$$e_r=(e_{r1},e_{r2},...,e_{r32}), \quad (5)$$

where e_{rs} are integers in the range from 0 to D, generated by a random number generator with a uniform distribution.

Each row (or column) of the input image is cyclically shifted by e_{rs} positions. If $e_{rs}=0$, then the line is not shifted. For each of the analogs, its own shift vector is formed, which remains unchanged during training and testing.

The dependence of the efficiency of using analogs depending on the value of D, as well as the differences when shifting along rows and columns, is subject to research.

In addition, the efficiency of using sparse shift vectors with $D=1$ was studied:

$$e_f=(e_{f1},e_{f2},...,e_{f32}), \quad (6)$$

where the elements of this vector are calculated using two independent vectors e_r^1 and e_r^2 by logical multiplication of the corresponding components:

$$e_{fi}=\begin{cases}e_{ri}^1 & \text{if } e_{ri}^2=1,\\ 0 & \text{if } e_{ri}^2=0.\end{cases} \quad (7)$$

The number of unities in the vector e_f is a random variable. The corresponding distribution obtained over 10,000 implementations is shown in Fig. 3.

For comparability with the results of work [9], the same neural networks were taken as basic first-stage classifiers. These are networks based on MLP architecture: MLP-Mixer [18], EANet [15], and gMLP [19]. As well as networks based on transformers: CCT [13], Fnet [16], and SwinTr [17]. The programs of these networks and their detailed descriptions are given in [30]. They are written in python using the Keras, TensorFlow, and Addons libraries. The initial values for the weighting coefficients were set randomly according to Xavier's initialization [31]. Training was performed on the training data for 50 epochs. The results are given in Table 1 [8].

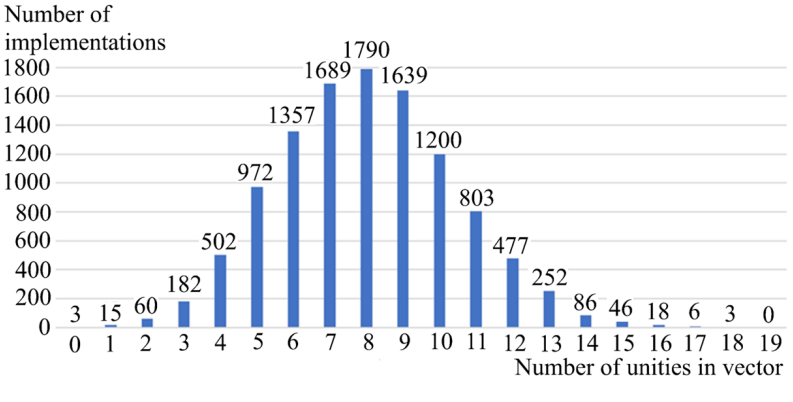

Fig. 3. Distribution of the number of unities in a sparse shift vector

To investigate the effectiveness of using analogs built using shifts, analogs were built for MLP-Mixer and CCT.

They were trained in the same way as basic neural networks.

Table 1

Parameters of classifiers of the first stage after training

Neural network	Quality of classification	Number of weights
CCT	0.8021	408139
EANet	0.6788	355530
FNet	0.7572	582410
SwinTr	0.7128	151386
MLP-Mixer	0.7674	219658
gMLP	0.7405	862218

MLP training was carried out using already trained first-stage classifiers for 10 epochs. 100 implementations were performed, and the best result was selected from them.

Classification quality was defined as the ratio of correctly recognized objects in test images to the total number of test images equal to 10,000.

All calculations were performed on the Colab platform using the A-100 GPU [32].

5. Results of investigating the effectiveness of a technique for constructing analogs of neural networks using shift operations

5. 1. Examining the number of matching errors between analogs and the basic neural network

Table 2 gives the number of matching errors for the basic neural network MLP-Mixer (designation 0_1) and ten implementations of its analog, differing in their random shift vector along the rows, with D=1. The first digit in the designation is the D value, the second is the implementation number. Data in Tables 2–4 are obtained from the results of classification of data from the training data set (50,000 images) after 50 epochs of training on the same data set.

Along the diagonal in the Table 2 is the number of errors made by the neural network itself and its analogs. Off the diagonal, the number of identical errors made by the neural network/analogs, indicated vertically and horizontally, is indicated. Table 2 shows that the number of errors made by analogs is comparable to the number of errors of the base neural network and may even be lower. The number of matching errors made by different networks is 57–65 % of the number of errors of the basic neural network.

Table 2

Number of matching errors

Neural network	0_1	1_1	1_2	1_3	1_4	1_5	1_6	1_7	1_8	1_9	1_10
0_1	8785	5229	5392	5117	5311	5292	5161	5433	5365	4967	5172
1_1	5229	8722	5425	5224	5257	5323	5407	5445	5429	5020	5381
1_2	5392	5425	8865	5133	5377	5327	5293	5539	5368	5097	5318
1_3	5117	5224	5133	8295	5179	5211	5123	5405	5317	4970	5285
1_4	5311	5257	5377	5179	8758	5435	5227	5699	5442	5177	5349
1_5	5292	5323	5327	5211	5435	8656	5277	5521	5327	5064	5315
1_6	5161	5407	5293	5123	5227	5277	8598	5546	5416	5031	5347
1_7	5433	5445	5539	5405	5699	5521	5546	9154	5742	5327	5612
1_8	5365	5429	5368	5317	5442	5327	5416	5742	9587	5172	5571
1_9	4967	5020	5097	4970	5177	5064	5031	5327	5172	7958	5074
1_10	5172	5381	5318	5285	5349	5315	5347	5612	5571	5074	8942

Table 3 gives the number of matching errors made by the basic neural network MLP-Mixer (0_1) and a group of its analogs. Table 4 – CCT (0_1) and a group of its analogs.

Table 3

Number of matching errors in the neural network group for MLP-Mixer

Neural network group	0_1	0_1 1_1	0_1 1_1 1_2	0_1 1_1 1_2 1_3	0_1 1_1 1_2 1_3 1_4	0_1 1_1 1_2 1_3 1_4 1_5	0_1 1_1 1_2 1_3 1_4 1_5 1_6	0_1 1_1 1_2 1_3 1_4 1_5 1_6 1_7
Number of matching errors made by all networks in the group	8785	5229	892	140	25	2	1	0

Table 4

Number of matching errors in the neural network group for CCT

Neural network group	0_1	0_1 1_1	0_1 1_1 1_2	0_1 1_1 1_2 1_3	0_1 1_1 1_2 1_3 1_4	0_1 1_1 1_2 1_3 1_4 1_5
Number of matching errors made by all networks in the group	7023	3541	558	69	8	0

Tables 3, 4 demonstrate that analogs make a significant number of their mistakes on different images. In a group consisting of the base MLP-Mixer neural network and seven analogs, there are no matching errors that all eight neural networks make. And for CCT, such a group contains only 6 networks.

5. 2. Investigating dependence of the resulting classification quality on the number of analogs of the neural network

This study was conducted on a complete ensemble classifier architecture. The base neural network and its analogs were pre-trained for 50 epochs on a training data set. Then their weighting coefficients remained constant and MLP was trained for 10 epochs also on the training data set. The MLP training was repeated 100 times, and the best set of weights was selected. After this, all weighting coefficients of the ensemble classifier remained constant and the quality of classification by the ensemble classifier of objects was checked on images from the test dataset (10,000 images).

Table 5 gives the quality of classification by the ensemble classifier of objects on images from the test data set for various values of the D range for the formation of shift vectors. At the first stage of the ensemble classifier, the basic neural network MLP-Mixer and N of its analogs were used. The cyclic shift of the input images was carried out line by line.

Table 5 demonstrates that increasing the range of shifts leads to a significant deterioration in the quality of classification. For the range D=1, the quality of classification increases with the number of analogs used up to N=9. When N=10, the same value is obtained. At the same time, the increase in classification quality increased from 0.7674 (without analogs, Table 1) to 0.7965 (for 9 analogs) or by 3.8 %. This corresponds to a reduction in the number of classification errors by 291 or 12.5 %.

Table 5

Dependence of classification quality on the number of analogs and on the range of shifts in the formation of shift vectors

No.	D=1	D=2	D=3	D=4	D=5	D=6
1	0.7828	0.7790	0.7818	0.7783	0.7755	0.7766
2	0.7856	0.7829	0.7829	0.7805	0.7794	0.7765
3	0.7910	0.7896	0.7840	0.7816	0.7823	0.7776
4	0.7920	0.7917	0.7863	0.7845	0.7833	0.7803
5	0.7936	0.7905	0.7852	0.7835	0.7841	0.7772
6	0.7953	0.7907	0.7842	0.7851	0.7824	0.7773
7	0.7957	0.7915	0.7859	0.7850	0.7840	0.7775
8	0.7954	0.7902	0.7861	0.7865	0.7838	0.7772
9	0.7965	0.7913	0.7860	0.7868	0.7841	0.7775
10	0.7965	0.7918	0.7873	0.7867	0.7835	0.7778

Similar data are given in Table 6, which compares row and column shifts for the range D=1 and the use of sparse shift vectors (DP – column shift, DL – row shift).

The data in Table 6 show that the use of sparse shift vectors and row shifts provides some advantage in classification quality and allows us to obtain a quality of 0.8029 when using 12 analogs. This represents a 4.6 % improvement in quality or a 15 % reduction in classification errors.

Table 6

Dependence of classification quality on the number of analogs for MLP-Mixer

No.	D=1 Line shift	D=1 Column shift	DP Column shift	DL Line shift
1	0.7828	0.7870	0.7828	0.7792
2	0.7856	0.7900	0.7902	0.7881
3	0.7910	0.7936	0.7943	0.7912
4	0.7920	0.7942	0.7949	0.7931
5	0.7936	0.7951	0.7969	0.7972
6	0.7953	0.7956	0.7987	0.8000
7	0.7957	0.7985	0.7995	0.8006
8	0.7954	0.7989	0.8004	0.8002
9	0.7965	0.7985	0.8012	0.8019
10	0.7965	0.7992	0.8015	0.8010
11	0.7975	0.7981	0.8011	0.8023
12	0.7972	0.7982	0.8007	0.8029

Table 7 gives similar data for the CCT basic classifier.

A single classifier has a classification quality on the test data set of 0.8021 or 1979 errors (Table 1). The data in Table 7 show that by increasing the number of analogs used to 11, it is possible to increase the quality of classification to 0.8492 (an increase of 5.9 %) and, accordingly, reduce the number of errors to 1508 (a decrease of 23.8 %). The data in Table 7 also demonstrate the advantage of using sparse shift vectors and row shifts.

Table 7

Dependence of classification quality on the number of analogs for CCT

No.	D=1 Line shift	D=1 Column shift	DP Column shift	DL Line shift
1	0.8242	0.8254	0.8223	0.8201
2	0.8295	0.8292	0.8286	0.8289
3	0.8340	0.8330	0.8342	0.8346
4	0.8372	0.8335	0.8352	0.8363
5	0.8381	0.8369	0.8377	0.8395
6	0.8400	0.8384	0.8413	0.8407
7	0.8414	0.8381	0.8424	0.8444
8	0.8420	0.8393	0.8417	0.8470
9	0.8428	0.8407	0.8446	0.8474
10	0.8425	0.8403	0.8439	0.8483
11	0.8434	0.8403	0.8435	0.8492
12	0.8422	0.8409	0.8449	0.8491

5. 3. Investigating the effectiveness of using analogs obtained using image shifts and rotations

Training was carried out on a set of training images. The resulting classification quality was checked on a test set of images with constant weight coefficients of the entire ensemble classifier.

The results of comparing the effectiveness of using analogs obtained using image rotations and shifts are given in Table 8. For comparability of results, the basic classifiers and data on analogs with rotation are taken from [9]. For shifts, only sparse shift vectors and row shifts were used. The selection of the best shift vectors was not carried out; once-generated vectors with random values were used. The change in the number of errors was calculated for ensemble classifiers with analogs at the first stage compared to analog classifiers with the same set of basic neural networks at the first stage, but without analogs.

The data in Table 8 show that adding MLP-Mixer analogs obtained by rotating images always led to some improvement in the classification quality of the entire ensemble classifier. In contrast, the use of MLP-Mixer analogs obtained by the shift method for some sets of base classifiers gave better results (–12 % errors) than rotation (–11 %), and for others – worse results (+0.7 % errors).

Table 8

Quality of classification of the ensemble classifier

First stage classifiers	Quality of classification	Number of errors	Change in the number of errors
1	2	3	4
CCT+EANet	0.8184	1816	0 %
CCT+EANet+2 analogs MLT-Mixer with rotation	0.8390	1610	–11 %
CCT+EANet+2 analogs MLT-Mixer with rotation	0.8402	1598	–12 %

Continuation of Table 8

1	2	3	4
CCT+EANet+2 analogs CCT with rotation	0.8429	1571	–13 %
CCT+EANet+MLT-Mixer	0.8401	1599	0 %
CCT+EANet+MLT-Mixer+2 analogs MLT-Mixer with rotation	0.8425	1575	–1.5 %
CCT+EANet+MLT-Mixer+2 analogs MLT-Mixer with rotation	0.8436	1564	–2.2 %
CCT+EANet+MLT-Mixer+2 analogs CCT with rotation	0.8482	1518	–5 %
CCT+EANet+MLT-Mixer+FNet	0.8445	1555	0 %
CCT+EANet+MLT-Mixer+FNet+2 analogs MLT-Mixer with rotation	0.8458	1542	–0.8 %
CCT+EANet+MLT-Mixer+FNet+2 analogs MLT-Mixer with shift	0.8458	1542	–0.8 %
CCT+EANet+MLT-Mixer+FNet+2 analogs CCT with shift	0.8496	1504	–3.3 %
CCT+EANet+MLT-Mixer+FNet+gMLP	0.8457	1543	0 %
CCT+EANet+MLT-Mixer+FNet+gMLP+2 analogs MLT-Mixer with rotation	0.8471	1529	–1 %
CCT+EANet+MLT-Mixer+FNet+gMLP+2 analogs MLT-Mixer with shift	0.8457	1543	0 %
CCT+EANet+MLT-Mixer+FNet+gMLP+2 analogs CCT with shift	0.8514	1486	–3.7 %
CCT+EANet+MLT-Mixer+FNet+gMLP+SwinTr	0.8468	1 532	0 %
CCT+EANet+MLT-Mixer+FNet+gMLP+SwinTr +2 analogs MLT-Mixer with rotation	0.8482	1518	–1 %
CCT+EANet+MLT-Mixer+FNet+gMLP+SwinTr+2 analogs MLT-Mixer with shift	0.8457	1543	+0.7 %
CCT+EANet+MLT-Mixer+FNet+gMLP+SwinTr+2 analogs CCT with shift	0.8515	1485	–3.1 %

The use of CCT analogs always provided an improvement in classification quality, both compared to the situation without the use of analogs and compared to the use of MLP-Mixer analogs obtained by rotation and shift. The range of error reduction resulting from the addition of sheared CCT analogs was from 3.1 % to 13 %.

6. Discussion of results of investigating the use of analogs of basic classifiers in ensemble classifiers

A distinctive feature of the proposed method, in contrast to well-known works, is the complete absence of additional floating point arithmetic operations for constructing analogs of basic neural networks. Analogs are obtained only by cyclically shifting some rows or columns of the input image.

Increasing the number of neural networks with structured processing of input images at the first stage of the ensemble classifier is possible in two ways. The first is the development of new architectures that differ from the known ones. Thus, in works [12–17], the architecture of transformers was taken as a basis and modifications were made to improve the ratio of classification quality to the volume of computation on small data sets. In [18], a new architecture based on MLP with a very high degree of structured processing was proposed. Its further modifications are reported in [18–26]. The disadvantage of this approach is that the number of known architectures that can be used at the first stage of the ensemble classifier is currently limited. New architectures are developed quite rarely, and their number will not increase significantly in the near future. The second approach, which complements the first, is the development of methods for constructing analogs of basic neural networks that would have the properties of basic ones but would make classification errors on other input images. The only known work in this area is [9], in which the use of rotations of input images at different fixed angles was proposed to construct analogs. The disadvantages of this method are the need to perform a significant volume of floating-point operations to rotate images and the limited number of angles by which it is advisable to rotate images. Our paper proposes a method that eliminates both of these disadvantages. To obtain analogs, it is proposed to use cyclic shifts of columns or rows of input images, that is, the need for additional floating-point operations has disappeared. The number of vectors specifying shifts is practically unlimited, that is, the number of generated analogs can be very large. The comparison (Table 8) showed that with the indicated advantages, the proposed method provides comparable and even better classification quality compared to using rotations of input images to form analogs.

Our study showed that shifting the columns and rows of input images makes it possible to obtain analogs of the basic neural networks of an ensemble classifier, the use of which at the first stage improves the quality of classification. This is explained by the fact that to use the first stage of an ensemble classifier with stacking, neural networks with structured processing of input signals are required. The structured processing leads to the fact that the cyclic shift of rows and columns changes the errors that the neural network makes. Combining the results of neural networks that make errors on different sets of images makes it possible to improve the resulting classification quality at the second stage.

As follows from Table 2, a large number of analogs obtained using shifts allows the selection of analogs with the best classification quality. This is an additional advantage of the proposed method for forming analogs.

As follows from Tables 5–7, the best results are provided by sparse shift vectors with a shift range of D=1. On average, such shift vectors require only 8 cyclic shifts of rows or columns per position (Fig. 3).

A comparison of the analogs of the neural networks MLP-Mixer and CCT showed that their analogs are significantly different. This suggests that in the practical implementation of ensemble classifiers with stacking, it is necessary to select analogs that provide the highest resulting classification quality. Comparison of Tables 3, 4 reveals that CCT analogs are highly diverse. The group of CCT and five analogs no longer has the same errors that all first-stage classifiers make. For MLP-Mixer, such a group should already contain 7 analogs. From Tables 6, 7, it is clear that increasing the number of MLP-Mixer analogs at the first stage to 12 allows increasing the resulting classification quality by 4.6 %. At the same time, for CCT this increase is 5.9 %. Adding two CCT analogs to different numbers of basic first-stage classifiers (Table 8) provided higher quality by 1–3.7 % compared to two MLP-Mixer analogs.

The limitations of the proposed approach that we can note are the need for additional research to select analogs of base classifiers that provide the greatest increase in the resulting classification quality of the entire ensemble classifier.

The disadvantage of the proposed approach is that with an increase in the number of analogs at the first stage, the increase in the resulting classification quality decreases and almost completely stops in the example under consideration when the number of analogs is 11–12 (Tables 6, 7). Overcoming this shortcoming requires additional research and the construction of new neural networks with structured processing. One of the options for developing this approach is to introduce dynamic cyclic shifts directly into the architecture of basic neural networks. This direction seems promising for finer adjustment of analogs during the learning process in order to improve the resulting quality of classification by an ensemble classifier.

7. Conclusions

1. Using the example of the basic neural networks MLP-Mixer, CCT, and the CIFAR-10 data set, it has been shown that the proposed method for generating analogs provides significantly different sets of images in which the basic neural network and its analogs make errors. This leads to an increase in the resulting classification quality of the ensemble classifier with stacking when adding analogs to the first stage.

2. When using MLP-Mixer analogs built according to the proposed method at the first stage of the ensemble classifier, increasing the number of analogs from 0 to 12 ensured an increase in the classification quality of the ensemble classifier from 0.7674 to 0.8029, that is, by 4.6 %. Accordingly, the number of errors decreased from 2326 to 1971, or by 15 %. For CCT analogs, the increase in classification quality was from 0.8021 to 0.8492, or 5.9 %. The number of errors decreased from 1979 to 1508, or by 23.8 %.

3. Comparison with analogs built by the image rotation method showed that analogs of MLP-Mixer, built by the proposed method, give a better resulting classification quality by 1–0.7 % only with a small number of base neural networks at the first stage. At the same time, CCT analogs built using shifts provided an advantage over MLP-Mixer analogs (with rotation) for all studied combinations of basic neural networks by 2–3.5 %.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study and the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

The manuscript has associated data in the data warehouse. The references are given in the paper.

Use of artificial intelligence

The authors used the technologies of artificial intelligence within the acceptable framework for providing reliable verification of data, described in the chapter on the research methodology.

References

1. Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B. et al. (2023). MedMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification. Scientific Data, 10 (1). https://doi.org/10.1038/s41597-022-01721-8
2. Islam, Md. R., Nahiduzzaman, Md., Goni, Md. O. F., Sayeed, A., Anower, Md. S., Ahsan, M., Haider, J. (2022). Explainable Transformer-Based Deep Learning Model for the Detection of Malaria Parasites from Blood Cell Images. Sensors, 22 (12), 4358. https://doi.org/10.3390/s22124358
3. Yang, X., He, X., Zhao, J., Zhang, Y., Zhang, S., Xie, P. (2020). COVID-CT-dataset: A CT scan dataset about COVID-19. arXiv. https://doi.org/10.48550/arXiv.2003.13865
4. Zhang, K., Liu, X., Shen, J., Li, Z., Sang, Y., Wu, X. et al. (2020). Clinically Applicable AI System for Accurate Diagnosis, Quantitative Measurements, and Prognosis of COVID-19 Pneumonia Using Computed Tomography. Cell, 182 (5), 1360. h https://doi.org/10.1016/j.cell.2020.08.029
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T. et al. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv. https://doi.org/10.48550/arXiv.2010.11929

6. Sun, C., Shrivastava, A., Singh, S., Gupta, A. (2017). Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. 2017 IEEE International Conference on Computer Vision (ICCV). https://doi.org/10.1109/iccv.2017.97
7. Aggarwal, C. C., Sathe, S. (2017). Outlier Ensembles. An Introduction. Springer International Publishing AG 2017, 276. https://doi.org/10.1007/978-3-319-54765-7
8. Galchonkov, O., Babych, M., Zasidko, A., Poberezhnyi, S. (2022). Using a neural network in the second stage of the ensemble classifier to improve the quality of classification of objects in images. Eastern-European Journal of Enterprise Technologies, 3 (9 (117)), 15–21. https://doi.org/10.15587/1729-4061.2022.258187
9. Galchonkov, O., Baranov, O., Babych, M., Kuvaieva, V., Babych, Y. (2023). Improving the quality of object classification in images by ensemble classifiers with stacking. Eastern-European Journal of Enterprise Technologies, 3 (9 (123)), 70–77. https://doi.org/10.15587/1729-4061.2023.279372
10. Krizhevsky, A., Sutskever, I., Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60 (6), 84–90. https://doi.org/10.1145/3065386
11. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2016.90
12. Li, K., Wang, Y., Zhang, J., Gao, P., Song, G., Liu, Y. et al. (2023). UniFormer: Unifying Convolution and Self-Attention for Visual Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45 (10), 1–18. https://doi.org/10.1109/tpami.2023.3282631
13. Hassani, A., Walton, S., Shah, N.,Abuduweili, A., Li, J., Shi, H. (2021). Escaping the Big Data Paradigm with Compact Transformers. arXiv. https://doi.org/10.48550/arXiv.2104.05704
14. Gao, A. K. (2023). More for Less: Compact Convolutional Transformers Enable Robust Medical Image Classification with Limited Data. arXiv. https://doi.org/10.48550/arXiv.2307.00213
15. Guo, M.-H., Liu, Z.-N., Mu, T.-J., Hu, S.-M. (2022). Beyond Self-Attention: External Attention Using Two Linear Layers for Visual Tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45 (5), 1–13. https://doi.org/10.1109/tpami.2022.3211006
16. Lee-Thorp, J., Ainslie, J., Eckstein, I., Ontanon, S. (2022). FNet: Mixing Tokens with Fourier Transforms. Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. https://doi.org/10.18653/v1/2022.naacl-main.319
17. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z. et al. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. 2021 IEEE/CVF International Conference on Computer Vision (ICCV). https://doi.org/10.1109/iccv48922.2021.00986
18. Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T. (2021). MLP-Mixer: An all-MLP Architecture for Vision. arXiv. https://doi.org/10.48550/arXiv.2105.01601
19. Liu, H., Dai, Z., So, D. R., Le, Q. V. (2021). Pay Attention to MLPs. arXiv. https://doi.org/10.48550/arXiv.2105.08050
20. Lian, D., Yu, Z., Sun, X., Gao, S. (2021). AS-MLP: An Axial Shifted MLP Architecture for Vision. arXiv. https://doi.org/10.48550/arXiv.2107.08391
21. Wang, Z., Jiang, W., Zhu, Y., Yuan, L., Song, Y., Liu, W. (2022). DynaMixer: A Vision MLP Architecture with Dynamic Mixing. arXiv. https://doi.org/10.48550/arXiv.2201.12083
22. Hu, Z., Yu, T. (2023). Dynamic Spectrum Mixer for Visual Recognition. arXiv. https://doi.org/10.48550/arXiv.2309.06721
23. Lv, T., Bai, C., Wang, C. (2022). MDMLP: Image Classification from Scratch on Small Datasets with MLP. arXiv. https://doi.org/10.48550/arXiv.2205.14477
24. Chen, S., Xie, E., Ge, C., Chen, R., Liang, D., Luo, P. (2023). CycleMLP: A MLP-Like Architecture for Dense Visual Predictions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45 (12), 14284–14300. https://doi.org/10.1109/tpami.2023.3303397
25. Borji, A., Lin, S. (2022). SplitMixer: Fat Trimmed From MLP-like Models. arXiv. https://doi.org/10.48550/arXiv.2207.10255
26. Yu, T., Li, X., Cai, Y., Sun, M., Li, P. (2022). S2-MLP: Spatial-Shift MLP Architecture for Vision. 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). https://doi.org/10.1109/wacv51458.2022.00367
27. The CIFAR-10 dataset. Available at: https://www.cs.toronto.edu/~kriz/cifar.html
28. Primery izobrazheniy i annotatsiy. Available at: https://docs.ultralytics.com/ru/datasets/classify/cifar10/#sample-images-and-annotations
29. Brownlee, J. (2019). Better Deep Learning. Available at: https://machinelearningmastery.com/better-deep-learning/
30. Code examples. Computer vision. Keras. Available at: https://keras.io/examples/vision/
31. Brownlee, J. (2021). Weight Initialization for Deep Learning Neural Networks. Available at: https://machinelearningmastery.com/weight-initialization-for-deep-learning-neural-networks/
32. Colab. Available at: https://colab.research.google.com/notebooks/welcome.ipynb