UDC 681.3.07: 004.8
DOI: 10.15587/1729-4061.2023.279372

IMPROVING THE QUALITY OF OBJECT CLASSIFICATION IN IMAGES BY ENSEMBLE CLASSIFIERS 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, USA, 78741
Mykola Babych
PhD, BI Engineer (FE Developer)
Digitally Inspired LTD
Bell Court, 2, Leapale Lane, Guildford,
United Kingdom, GU14LY
Varvara Kuvaieva
PhD, Associate Professor*
Yuliia Babych
PhD, Associate Professor
Department of Design Information Technologies and Design
Institute of Digital Technologies, Design and Transport**
*Department of Information Systems
Institute of Computer Systems**
**Odessa Polytechnic National University
Shevchenko ave., 1, Odessa, Ukraine, 65044

The object of research is the process of classifying objects in images. The quality of classification refers to the ratio of correctly recognized objects to the number of images. One of the options for improving the quality of classification is to increase the depth of neural networks used. The main difficulties along the way are the difficulty of training such neural networks and a large amount of computing that makes it difficult to use them on conventional computers in real time. An alternative way to improve the quality of classification is to increase the width of the neural networks used, by constructing ensemble classifiers with staking. However, they require the use of classifiers at the first stage with different structured processing of input images, characterized by high quality classification and relatively low volume of calculations. The number of known such architectures is limited. Therefore, the problem arises of increasing the number of classifiers at the first stage of the ensemble classifier by modifying known architectures. It is proposed to use blocks of rotation of images at different angles relative to the center of the image. It is shown that as a result of structured image processing by the starting classifier, processing of rotated image leads to redistribution of errors on image set. This effect makes it possible to increase the number of classifiers in the first stage of the ensemble classifier. Numerical experiments have shown that adding two analogs of the MLP-Mixer algorithm to known configurations of ensemble classifiers reduced the error from 1 to 11 % when working with the CIFAR-10 dataset. Similarly, for CCT, the error reduction was between 2.1 and 10 %. In addition, it has been shown that increasing the MLP-Mixer configuration in width gives better results than increasing in depth. A prerequisite for the success of using the proposed approach in practice is the structured image processing by the starting classifier

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

Received date 13.03.2023
Accepted date 16.05.2023
Published date 30.06.2023

How to Cite: *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. doi: https://doi.org/10.15587/1729-4061.2023.279372*

1. Introduction

Neural networks are becoming more and more widespread in all areas of human life. This is facilitated by the constant increase in the efficiency of their work. However, in most cases, this is achieved through the use of deeper neural networks, more powerful supercomputers, and more extensive data sets. The next step after achieving high performance is to reduce computational costs with the same or even better quality of work and ensure that these neural networks work on small data sets. Two different approaches are used to this end. The first of these is thinning or compression [1]. With the help of these methods, it is possible to achieve a reduction in computational costs by 30–50 times, without degrading the quality of neural networks [2] or even improving [3]. In addition, modifications of neural networks are carried out in the direction of reducing the amount of memory used [4]. The same area includes methods of low-rank factorization [5] and knowledge distillation [6]. However, due to the fact that the training of neural networks is random, the resulting architecture is not homogeneous. This leads to a complication of implementation on parallel computing architectures.

Within the framework of the second approach, new modifications of neural network architectures are immediately developed. Due to their deterministic regularity, they are easily parallelized and provide uniform loading of parallel processors. The highest results in pattern recognition in images are shown by the architecture of a neural network called Transformer (ViT) [7]. The most powerful variant of this architecture, ViT-Huge, contains more than 632 million configurable weights. Within the framework of the second approach, a large number of modifications of this architecture have been implemented, characterized by high quality of pattern recognition and significantly less computation. In [8], a modification of the architecture with overlapping patches (areas into which the original image in ViT is divided) is proposed. This makes it possible to get a gain by taking into account the local image structures located at the patch boundaries. In [9], an additional positional self-attention was introduced into the architecture of the transformer, implemented using a convolution layer. This improved the characteristics of the transformer in terms of flexibility in the arrangement of objects in the image. In [10], the architecture of the transformer is modified by using a convolution to extract low-level features by building a multi-level cross-attention mechanism. In [11], convolutional projections are built into the tokenization process instead of linear projections. In [12], the mechanism of self-attention with quadratic complexity is replaced by the mechanism of external attention with linear complexity. Extensive numerical experiments have shown the same or even higher quality of work than with the use of internal attention, with significantly lower computational costs.

The considered approaches to the construction of the architecture of neural networks involve increasing the capabilities of the neural network by increasing its depth. Better results are obtained by using a larger number of series-connected blocks. Additional opportunities for improving performance are opened by combining neural networks into ensembles [13]. At the same time, the use of stacking seems to be the most promising. This architectural technique makes it possible to increase the capabilities of the neural network by increasing its width. At the same time, the possibilities for parallelizing calculations to implement them on multiprocessor computers naturally increase. The greatest effect from the use of stacking is achieved when neural networks with different architectures are combined and process input signals in different ways. The more efficient the merged neural networks are and the more their processing algorithms differ, the better the results of the ensemble of these neural networks. Therefore, in order to improve the ratio of the qualitative indicators of neural networks and the volume of calculations for their implementation, it is relevant to study the possibilities of increasing the efficiency of ensemble classifiers.

2. Literature review and problem statement

In [12], it is proposed to use a neural network at the second stage of an ensemble classifier with stacking. Moreover, it is not the output numbers with the maximum value of the classifiers of the first stage that are fed to its input, but their full output vectors. Such an architectural solution makes it possible to significantly improve the accuracy of the classification, in comparison with the methods of majority voting. Comparison of such an architecture with a classic transformer [7] shows its advantage not only in terms of classification accuracy but also a much smaller amount of calculations. However, the number of currently known architectures that can be used in the first stage is limited. And this significantly limits the possibility of further improving the quality of the classification of objects in the images by an ensemble classifier with stacking.

It is known [13] that one of the main conditions for the effective operation of an ensemble classifier with stacking in the second stage is a variety of architectures and algorithms for processing input signals in classifiers of the first stage. In addition, they require high quality classification and a small amount of calculations. Therefore, in [12], for the first stage, the following modifications of the transformer with small amounts of calculations were chosen in the first place.

CCT (Compact Convolutional Transformer) [14] is a transformer supplemented with blocks of convolutional neural networks and using a reduced patch size. In [14], the main efforts are made to ensure the effective operation of the neural network on small data sets. To do this, the size of patches has been significantly reduced, the method of sequential integration of information from the encoder has been used, and a convolution with small steps is used in the tokenization process.

EANet (External Attention Transformer) [15] is a transformer where the self-attention block with quadratic complexity is replaced by an external attention block with linear complexity and additional multilayer perceptron (MLP) blocks are used. Extensive numerical experiments have shown the same or even higher quality of work than with the use of internal attention, with significantly lower computational costs.

FNet [16] is a transformer where the self-attention block is replaced by a simpler shuffling of tokens using a fast Fourier transform.

SwinTr (Swin Transformer) [17] is a transformer where self-attention is calculated not for the entire image but only inside each of the shifted windows. This provides additional flexibility when processing objects at different scales.

The main advantages of the classifiers proposed in [14–17] are the structuring of processing and the difference between processing. However, the main attention in these works is aimed at improving the quality of classification by increasing the depth of the neural network used. The possibilities of improving the quality of classification by further increasing the width are not considered.

Along with the modifications of the transformer, another direction in the construction of the architecture of classifiers is known – MLP-like [18, 19]. They do not contain attention blocks and convolutions and process the input images using specially structured multilayer perceptrons (MLPs). The main idea of the processing is that spatial relationships are not searched for all at once throughout the image. First, vertical connections are searched, then horizontally. Such classifiers demonstrate the quality of classification and the amount of calculations similar to those of transformer modifications. However, [18, 19] also did not consider the possibility of further improving the quality of classification by increasing the width of the neural network. Since the signal processing algorithms in them differ significantly from transformers, it seems appropriate to include them in the first stage of the ensemble classifier. In [12], the following classifiers were included in the first stage.

MLP-Mixer [18]. Each of the blocks of this classifier contains two groups of perceptrons. The first group processes separately each of the vertical rows of patches into which the input image is divided, the second group processes the results of the work of the first group, combined into horizontal rows. This makes it possible to classify objects in a full image with a relatively small amount of calculations.

gMLP [19]. The idea of image processing in these classifiers is similar to that in MLP-Mixer but implemented in a different way. Channel projections and interchannel (spatial) projections are used, supplemented by multiplicative strobing.

Based on the ideology of constructing an ensemble classifier with stacking, it follows that the more diverse classifiers in the first stage, the higher the quality of classification. However, the number of known suitable architectures is limited, so it is necessary to study the possibility of constructing their analogs for use in the ensemble classifier.

3. The aim and objectives of the study

The aim of this work is to improve the quality of classification of objects in images by ensemble classifiers with stacking by adding effective analogs of well-known architectures to the first stage.

To accomplish the aim, the following tasks have been set:

– to investigate the efficiency of rotation of input images using the example of MLP-Mixer;

– to compare the efficiency of increasing the number of blocks in MLP-Mixer and increasing the number of MLP-Mixer analogs in the ensemble classifier;

– to investigate the effectiveness of adding analogs of classifiers to the first stage of the ensemble classifier.

4. The study materials and methods

The object of research in this work is ensemble classifiers with stacking. The main hypothesis of the study is that the rotation of the input images during structured processing leads to a change in the numbers of images on which the classifier makes an error. This should make it possible to build analogs of known classification algorithms and, due to this, increase the number of suitable classifiers in the first stage of the ensemble classifier. This approach does not imply any additional simplifications and assumptions, which makes the results obtained universal.

The generalized architecture of the ensemble classifier is shown in Fig. 1. It uses in the first stage M classifiers and L of their analogs.

For the comparability of the results with work [12], the same classifiers were taken for the first stage: CCT, EANet, FNet, SwinTr, MLP-Mixer, gMLP. In addition, at the first stage, analogs of the best classifiers from each of the subgroups – CCT and MLP-Mixer – were used. Since convolutional neural networks (CNN) [20] show comparable results as single classifiers, CNN analogs were also used. The algorithm of CNN differs significantly from the two groups of classifiers considered in that the processing is not structured by channels and between them, but the entire image is processed at once. The inputs of the main classifiers receive directly images x. Analogs differ from the main classifiers only in that their inputs receive images x in the form turned at q_i degrees. Naturally, they are also trained on rotated images.

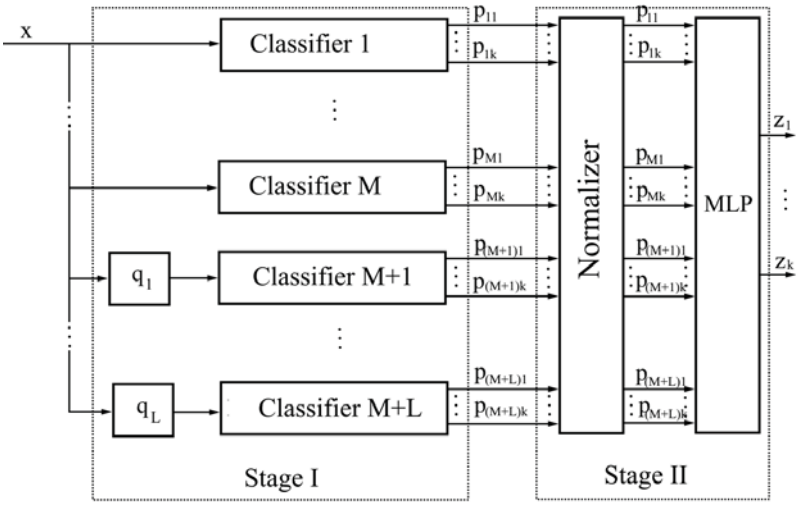

Fig. 1. Generalized architecture of the ensemble classifier

From the first stage to the second, the full output vectors of each of the classifiers are received:

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

where i is the number of the classifier,

k is the number of data classes,

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

In the second stage, these signals are normalized and then fed to the multilayer perceptron (MLP). Normalization is carried out for each of the classifiers separately:

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

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

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

$p_{ij}(x)$ is the j-th output of the i-th classifier.

A multilayer perceptron (MLP) contains 3 layers [21]:

– input layer, dimensionality k×(M+L);

– hidden layer, dimensionality (k–1)×(M+L), Relu activation function;

– output layer, 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 $z_j(x)$ are the signals at the output of the MLP with the image at the input x.

Numerical experiments were conducted on the CIFAR-10 dataset [22], which contains 50,000 color images for training, 32×32 pixels, with objects of 10 classes, such as airplane, steamer, car, horse, etc., and 10,000 similar images for test-

ing. The images were rotated relative to the center of the image. The pixel values of the rotated image were recalculated only within the boundaries of a circle with a radius of 15.5 centered in the center of the image if we take the distances between pixels as 1. Pixels outside this circle in the corners of the images remained unchanged. Examples of an image without rotation, with a rotation of +3 degrees and with a rotation of 10 degrees (a counterclockwise rotation is taken as a positive direction) are shown, respectively, in Fig. 2, *a*–*c*.

a *b* *c*

Fig. 2. Image of a car from the CIFAR-10 set: *a* — without rotation; *b* — rotation +3 degrees; *c* — rotation −10 degrees

The programs of the classifiers of the first stage are written in Python using the libraries Num.py, Tensorflow, Keras, Addons, and are taken from [23]. Training of classifiers of the first stage was carried out in 50 epochs, the results are given in Table 1 [12].

MLP training was carried out over 4 epochs. In MLP training, the classifiers of the first stage were not trained and had constant weights achieved as a result of training. For each configuration of the ensemble classifier, MLP training was performed 100 times. The initial conditions were set randomly each time according to Xavier's initialization [24]. As a result, the maximum value of the classification quality was used.

Table 1

Parameters of classifiers of the first stage after training

Neural network	Classification quality	Number of weights
CNN	0.7687	343306
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

All classifiers were trained on training data. The resulting classification quality was determined on the basis of data processing for testing as the ratio of correctly recognized objects to the total number of test images.

All calculations were performed on the Colab cloud platform [25].

5. Results of the study of the ensemble classifier

5. 1. Investigation of the efficiency of rotation of input images on the example of MLP-Mixer

Fig. 3 shows the architecture of the MLP-Mixer base layer [18]. The image is represented as a collection of channels. At the beginning, the channels are divided into fragments and inter-channel processing is performed, then for each channel separately. If the image at the input of the layer is rotated relative to its center, the redistribution of the pixels of the original image between the channels and between the fragments of the channels changes. This will lead to a change in the processing results.

Tables 2, 3 give the number of coincident errors at the output of MLP-Mixer when feeding images from the training dataset without rotation and with different rotations (rotations are indicated in degrees). MLP-Mixer contained 4 base layers.

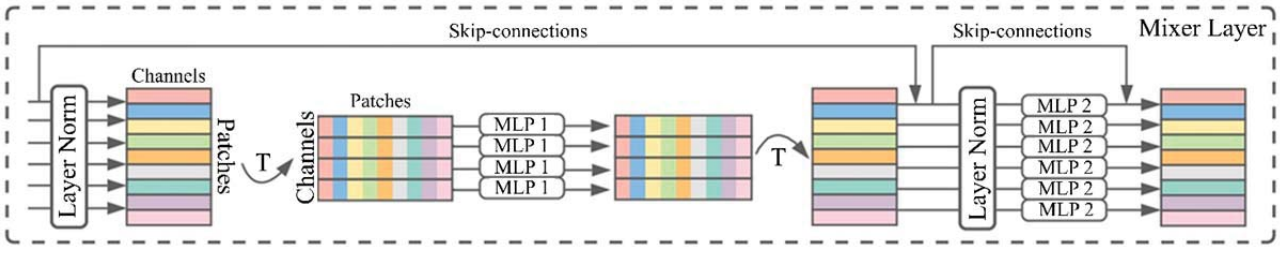

Fig. 3. Base layer architecture MLP-Mixer [18]

Table 2

The number of coincident errors

MLP-Mixer and rotation	No rotation	Rotation +3°	Rotation +5°	Rotation +10°	Rotation +15°	Rotation +20°	Rotation +25°	Rotation +30°
No rotation	8314	5103	5419	5734	4374	6261	6217	6395
Rotation +3°	5103	10427	6722	6778	4915	7659	7792	8133
Rotation +5°	5419	6722	11805	8043	5254	8984	9045	9297
Rotation +10°	5734	6778	8043	14090	5386	11148	11171	11286
Rotation +15°	4374	4915	5254	5386	7649	5844	5764	6012
Rotation +20°	6261	7659	8984	11148	5844	22825	18685	18852
Rotation +25°	6217	7792	9045	11171	5764	18685	26481	22425
Rotation +30°	6395	8133	9297	11286	6012	18852	22425	29209

Table 3

The number of coincident errors

MLP-Mixer and rotation	No rotation	Rotation –3°	Rotation –5°	Rotation –10°	Rotation –15°	Rotation –20°	Rotation –25°	Rotation–30°
No rotation	8314	4814	4517	5733	5091	5139	5122	5040
Rotation –3°	4814	8387	4813	5964	5295	5380	5405	5395
Rotation –5°	4517	4813	7943	5546	4969	5026	5117	5099
Rotation –10°	5733	5964	5546	13919	6817	7108	7243	7226
Rotation –15°	5091	5295	4969	6817	9685	6375	6289	6255
Rotation –20°	5139	5380	5026	7108	6375	10439	6819	6776
Rotation –25°	5122	5405	5117	7243	6289	6819	11010	7152
Rotation –30°	5040	5395	5099	7226	6255	6776	7152	11578

Tables 2, 3 show that the number of coincident errors between the variant without rotation of the input images and the variants with their rotation is significantly less than the total number of errors in the variant without rotation. However, as the rotation angle increases, the total number of errors and the number of coincident errors between variants with different rotation angles increase significantly. Moreover, for positive rotation angles, this increase is greater than for negative angles. This is observed due to the asymmetry of the location of objects relative to the center of the image.

Table 4 gives the number of coincident errors present simultaneously in all variants, depending on the set of variants with different angles of rotation of the images at the input.

Table 4

The number of coincident errors

MLP-Mixer+analogs with rotation	Number of coincident errors
MLP-Mixer	8314
MLP-Mixer, Rotation –3°	4814
MLP-Mixer, Rotation –3°, Rotation +3°	981
MLP-Mixer, Rotation –3°, Rotation +3°, Rotation –5°	146
MLP-Mixer, Rotation –3°, Rotation +3°, Rotation –5°, Rotation +5°	29
MLP-Mixer, Rotation –3°, Rotation +3°, Rotation –5°, Rotation +5°, Rotation–10°	2
MLP-Mixer, Rotation –3°, Rotation +3°, Rotation –5°, Rotation +5°, Rotation–10°, Rotation +10°	1
MLP-Mixer, Rotation –3°, Rotation +3°, Rotation –5°, Rotation +5°, Rotation –10°, Rotation +10°, Rotation –15 °	0

From the data given in Tables 2–4, it follows that the rotation of the images fed to the input of the classifier with significantly structured processing leads to significant differences in errors.

5. 2. Comparison of the efficiency of increasing the depth of the MLP-Mixer and the width of the ensemble classifier

The quality of the classification can be increased either in depth, increasing the number of blocks connected in series, or in width. In the latter case, the number of classifiers at the first stage of the ensemble classifier increases. For comparison, classifiers containing MLP-Mixer blocks consisting of 4 base layers connected in series, shown in Fig. 3, were used. At the first stage, the ensemble classifier contained MLP-Mixer with one block of layers and no rotations of the input image, and the same classifiers, but with different angles of rotation of the input images. With the same number of blocks, the variant with an increase in depth and the variant with an increase in width have almost the same amount of calculations. The results of the comparison are given in Table 5.

The spread of values for MLP-Mixer is explained by the fact that due to the large amount of calculations, each configuration was trained once, starting with random values for weight coefficients, with a stop at the highest possible result.

In the ensemble classifier, the second stage was trained 100 times and the best result was chosen.

Table 5

Comparison of depth and width build-up

Number of blocks	MLP-mixer with build-up in depth	MLP-mixer + analogs (wide-width extension)	
	Classification accuracy	Classification accuracy (improvement)	Rotation angles of input images for different analogs in degrees
1	0.7674	0.7674	0
2	0.7814	0.7881 (+0,9 %)	0 +(–3)
3	0.7868	0.7920 (+0,7 %)	0+(–3)+(+3)
4	0.7633	0.7934 (+3,9 %)	0+(–3)+(+3)+ +(–5)
5	0.7740	0.7951 (+2,7 %)	0+(–3)+(+3)+ +(–5)+(+5)
6	0.7831	0.7957 (+1,6 %)	0+(–3)+(+3)+ +(–5)+(+5)+ +(–10)
7	0.7719	0.7955 (+3,1 %)	0+(–3)+(+3)+ +(–5)+(+5)+ +(–10)+(+10)
8	0.7824	0.7952 (+1,6 %)	0+(–3)+(+3)+ +(5)+(+5)+ +(–10)+ +(+10)+(–15)

From the results given in Table 5, it can be seen that the width extension for the MLP-Mixer is a more preferable option than the depth extension. However, it should be noted that the addition of classifiers 7 and 8 with large angles of rotation of the input images to the ensemble classifier led not to an improvement but to a deterioration in the results.

5. 3. Investigation of the effectiveness of adding analogs of classifiers to the first stage of the ensemble classifier

Sets from [12] were used as initial sets of classifiers of the first stage of the ensemble classifier. Analogs of MLP-Mixer, CCT, and CNN classifiers were added to them. The results are shown in Table 6. In each cell of the results, the first number is the accuracy of the classification, the second number is the number of errors on the test set, the third is the percentage of increasing or decreasing the number of errors in relation to the set without adding analogs.

Table 6

Results of adding analogs to the first stage

The main classifiers of the first stage	Classification quality without the use of analogs	Analogs to be added					
		MLP-Mixer Rotation –3°	MLP-Mixer Rotation –3°, Rotation +3°	CCT Rotation –3°	CCT Rotation –3°, Rotation +3°	CNN Rotation –3°	CNN Rotation –3°, Rotation +3°
CCT+EAT	0.8184	0.8365	0.8390	0.8354	0.8374	0.7815	0.8024
	1816	1635	1610	1646	1626	2185	1976
	0 %	–10 %	–11 %	–9 %	–10 %	+20 %	+9 %
CCT+EAT+ +MLP-Mixer	0.8401	0.8421	0.8425	0.8434	0.8460	0.7893	0.8056
	1599	1579	1575	1566	1540	2107	1944
	0 %	–1,3 %	–1,5 %	–2,1 %	–3,7 %	+32 %	+22 %
CCT+EAT+ +MLP-Mixer+ +FNet	0.8445	0.8461	0.8458	0.8465	0.8493	0.7894	0.8062
	1555	1539	1542	1535	1507	2106	1938
	0 %	–1 %	–0,8 %	–1,3 %	–3,1 %	+35 %	+25 %
CCT+EAT+ +MLP-Mixer+ +FNet+gMLP	0.8457	0.8473	0.8471	0.8490	0.8496	0.7913	0.8067
	1543	1527	1529	1510	1504	2087	1933
	0 %	–1 %	–1 %	–2,1 %	–2,5 %	+35 %	+25 %
CCT+EAT+ +MLP-Mixer+ +FNet+ +gMLP+ +SwinTr	0.8468	0.8473	0.8482	0.8491	0.8500	0.7889	0.8034
	1 532	1527	1518	1509	1500	2111	1966
	0 %	–0,33 %	–1 %	–1,5 %	–2,1 %	+38 %	+28 %

From the results of Table 6, it can be seen that the addition of analogs of the MLP-Mixer classifier made it possible to reduce the number of errors by 1–11 %, depending on the initial set of classifiers of the first stage. The reduction in the number of errors for analogs of the CCT classifier was in the range of 2.1 %–10 %. The use of analogs of the CNN convolutional neural network led to an increase in errors from 9 % to 38 %.

6. Discussion of results of investigating the possibility of improving the quality of the classification of objects in images by ensemble classifiers

Our study showed that feeding to classifiers with structured processing of images rotated at different angles leads to various errors in the classification of objects in images. Table 4 shows that with a sufficiently large number of classifiers with different angles of rotation of the input images, the number of errors made simultaneously by all classifiers can be reduced to zero. This is due to the structuring of image processing in the first-stage classifiers used. Using this processing property makes it possible to build analogs of well-known classifier architectures and use them in the first stage of the ensemble classifier. As shown by the results of Table 6, it makes it possible to improve the resulting quality of classification. Thus, when two analogs of the MLP-Mixer algorithm were added to the known architecture, an error reduction of 1 % to 11 % was observed. The addition of two analogs of the CCT algorithm made it possible to reduce the number of errors from 2.1 % to 10 %. At the same time, if the image processing is not structured in the first-stage classifier and the entire image is processed at once, image rotation leads to a significant deterioration in the quality of classification of the ensemble classifier. The results in Table 6 show that for CNN, the number of errors increased by 9 % to 38 %.

However, as follows from Tables 2, 3, the increasing total number of errors when increasing the angle of rotation of the image does not allow full use of this property of analogs. The results of the experiments given in Table 5 showed that in ensemble classifiers, for the implementation of additional classifiers of the first stage, it is preferable to use small angles of rotation.

Of particular interest is the comparison of the effectiveness of increasing the classifier in depth and width. In [14–19], only an increase in depth of neural networks is used to increase the quality of classification. Results in Table 5 show that with almost the same amount of calculations, increasing the width of the MLP-Mixer classifier provided a higher quality of classification in the range from 0.7 % to 3.9 %. In addition to directly improving the quality of classification, the ensemble classifier has a number of advantages. First of all, it is the facilitation and acceleration of learning. It is easier to train a few simple classifiers than one, but very deep. In addition, the computation capacity of the ensemble classifier is easier to parallelize for multi-core and multiprocessor architectures and provides higher performance when implemented in real time.

As a limitation of the study, it should be noted that only analogs of classifiers with structured image processing can be effectively used. And the number of useful analogs for one type of classifier is limited. Therefore, further improvement of the quality of classification of ensemble classifiers requires not only the use of analogs at the first stage but also the development of new architectures that increase the variety of classifiers of the first stage.

As a disadvantage of the proposed solution for improving the quality of classification of objects in images, one can note the limited number of analogs of known architectures that can be effectively used. It is possible to eliminate this drawback by developing new architectures of neural networks with structured processing that differs from the known ones, and by developing more advanced methods for constructing analogs of known architectures.

7. Conclusions

1. Using the example of the MLP-Mixer classifier, it is shown that the rotation of the input images during structured processing in the classifier leads to significant differences in the images on which the classifier makes recognition errors.

2. It is shown that increasing the width of classifiers by increasing the number of classifiers at the first stage of the ensemble classifier provides an improvement in the quality of classification compared to an increase in depth with almost the same amount of calculations. Using the example of the MLP-Mixer algorithm and the CIFAR-10 dataset, this improvement ranged from 0.7 % to 3.9 %.

3. In this paper, it is proposed to add analogs of well-known classifier architectures built by adding image rotation blocks to the first stage of the ensemble classifier with stacking. It is shown that this makes it possible to increase the quality of the classification of the ensemble classifier. In particular, when working with the CIFAR-10 dataset, the addition of two analogs of the MLP-Mixer algorithm to the first stage provided an error reduction of 1 % to 11 %. Similarly, for CCT, the error reduction ranged from 2.1 % to 10 %.

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 related data in the data warehouse. Links are given in the text of the paper.

References

1. Mary Shanthi Rani, M., Chitra, P., Lakshmanan, S., Kalpana Devi, M., Sangeetha, R., Nithya, S. (2022). DeepCompNet: A Novel Neural Net Model Compression Architecture. Computational Intelligence and Neuroscience, 2022, 1–13. doi: https://doi.org/10.1155/2022/2213273
2. Han, S., Mao, H., Dally, W. J. (2015). Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv. doi: https://doi.org/10.48550/arXiv.1510.00149
3. Galchonkov, O., Nevrev, A., Glava, M., Babych, M. (2020). Exploring the efficiency of the combined application of connection pruning and source data pre-processing when training a multilayer perceptron. Eastern-European Journal of Enterprise Technologies, 2 (9 (104)), 6–13. doi: https://doi.org/10.15587/1729-4061.2020.200819
4. Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv. doi: https://doi.org/10.48550/arXiv.1602.07360
5. Wu, K., Guo, Y., Zhang, C. (2020). Compressing Deep Neural Networks With Sparse Matrix Factorization. IEEE Transactions on Neural Networks and Learning Systems, 31 (10), 3828–3838. doi: https://doi.org/10.1109/tnnls.2019.2946636
6. Cheng, X., Rao, Z., Chen, Y., Zhang, Q. (2020). Explaining Knowledge Distillation by Quantifying the Knowledge. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). doi: https://doi.org/10.1109/cvpr42600.2020.01294
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T. et al. (2021). An image is worth 16x16 words: transformers for image recognition at scale. arXiv. doi: https://doi.org/10.48550/arXiv.2010.11929
8. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z. et al. (2021). Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet. 2021 IEEE/CVF International Conference on Computer Vision (ICCV). doi: https://doi.org/10.1109/iccv48922.2021.00060
9. d'Ascoli, S., Touvron, H., Leavitt, M. L., Morcos, A. S., Biroli, G., Sagun, L. (2022). ConViT: improving vision transformers with soft convolutional inductive biases. Journal of Statistical Mechanics: Theory and Experiment, 2022 (11), 114005. doi: https://doi.org/10.1088/1742-5468/ac9830
10. Yuan, K., Guo, S., Liu, Z., Zhou, A., Yu, F., Wu, W. (2021). Incorporating Convolution Designs into Visual Transformers. 2021 IEEE/CVF International Conference on Computer Vision (ICCV). doi: https://doi.org/10.1109/iccv48922.2021.00062
11. Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L. (2021). CvT: Introducing Convolutions to Vision Transformers. 2021 IEEE/CVF International Conference on Computer Vision (ICCV). doi: https://doi.org/10.1109/iccv48922.2021.00009
12. 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. doi: https://doi.org/10.15587/1729-4061.2022.258187
13. Rokach, L. (2019). Ensemble Learning. Pattern Classification Using Ensemble Methods. World Scientific Publishing Co. doi: https://doi.org/10.1142/11325
14. Hassani, A., Walton, S., Shah, N., Abuduweili, A., Li, J., Shi, H. (2021). Escaping the Big Data Paradigm with Compact Transformers. arXiv. doi: https://doi.org/10.48550/arXiv.2104.05704
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, 1–13. doi: 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. doi: 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). doi: https://doi.org/10.1109/iccv48922.2021.00986
18. Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T. et al. (2021). MLP-Mixer: An all-MLP Architecture for Vision. arXiv. doi: 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. doi: https://doi.org/10.48550/arXiv.2105.08050
20. Brownlee, J. (2019). Deep Learning for Computer Vision. Image Classification, Object Detection, and Face Recognition in Python. Available at: https://machinelearningmastery.com/deep-learning-for-computer-vision/
21. Brownlee, J. (2019). Better Deep Learning. Train Faster, Reduce Overfitting, and Make Better Predictions. Available at: https://machinelearningmastery.com/better-deep-learning/
22. Krizhevsky A. The CIFAR-10 dataset. Available at: https://www.cs.toronto.edu/~kriz/cifar.html
23. Code examples / Computer vision. Keras. Available at: https://keras.io/examples/vision/
24. Brownlee, J. (2021). Weight Initialization for Deep Learning Neural Networks. Available at: https://machinelearningmastery.com/weight-initialization-for-deep-learning-neural-networks/
25. Colab. Available at: https://colab.research.google.com/notebooks/welcome.ipynb