Convolutional neural network in the images colorization problem

M V Bulygin[1], M M Gayanova[1], A M Vulfin[1], A D Kirillova[1] and R Ch Gayanov[2]

[1]Ufa State Aviation Technical University, K. Marks str. 12, Ufa, Russia, 450008
[2]Higher School of Economics, Myasnitskaya str., 20, Moscow, Russia, 101000

e-mail: vulfin.alexey@gmail.com, kirillova.andm@gmail.com

Abstract. Object of the research are modern structures and architectures of neural networks for image processing. Goal of the work is improving the existing image processing algorithms based on the extraction and compression of features using neural networks using the colorization of black and white images as an example. The subject of the work is the algorithms of neural network image processing using heterogeneous convolutional networks in the colorization problem. The analysis of image processing algorithms with the help of neural networks is carried out, the structure of the neural network processing system for image colorization is developed, colorization algorithms are developed and implemented. To analyze the proposed algorithms, a computational experiment was conducted and conclusions were drawn about the advantages and disadvantages of each of the algorithms.
Keywords: colorization, convolutional neural networks, deep neural networks, image processing, image compression, outlining of contours.

1. Introduction

Modern neural networks (NN) show good results in a wide range of image processing tasks (Figure 1), which could not be achieved earlier by other methods. Thus, the neural network ResNet50 in the classification problem on the Imagenet set showed an accuracy of 96.43%, while the average person correctly recognizes only 94.9% of the images [1-5].

The urgency of the problem is explained by the need to reduce the computational complexity of implementing neural networks for image processing.

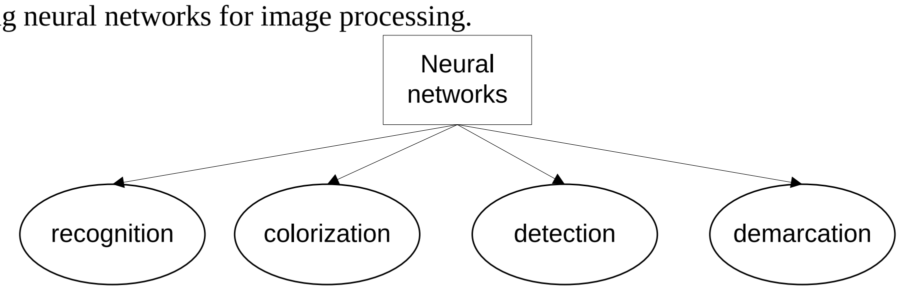

Figure 1. Tasks solved with the help of neural networks.

Goal of the work is improving the existing image processing algorithms based on the extraction and compression of features using neural networks using the colorization of black and white images as an example.

To achieve this goal it is necessary to solve the following tasks:

1. Analysis of image processing algorithms based on neural networks;
2. Development of the structure of a neural network processing system for image colorization;
3. Development of a heterogeneous neural network architecture in the problem of colorization of images;
4. Carrying out the experiment and analyzing the results.

2. Analysis of image processing algorithms based on neural networks

Image colorization is the process of adding color to a monochromatic (black and white) image or video [6]. The color space is constructed in such a way that any color is represented by a point having certain coordinates.

The problem of colorization does not have an unambiguous solution, since one gray scale corresponds to several color space points at once. For this reason, for colorization, it is necessary to use not only data about the color of the point, but also additional information. The source of such information can serve as another image (reference image), or expert opinion, or, identified in the image by a neural network an additional high-level features [7-10].

Today, colorization is in demand, for example, for color versions of black and white films. There are many methods for solving the problem of colorizing images, each of which has its own advantages and disadvantages – Table 1 [11].

Table 1. Methods of image colorization.

Method	Advantages	Disadvantages
Manual colorization	Accuracy of the colorization	Manual division into multiple zones with the color assignment; Impossibility of automatically separating the boundaries of significant areas in the presence of fuzziness or with considerable complexity
Neural network coloring based on reference points and expert data	High processing speed (5-7 s); Quite high quality of colorization due to the analysis of expert data	It is not always possible to determine the colors of the desired image points; Self-matching color for a point is a difficult task; If coloring a large number of similar images, it is necessary to specify hints points for each.
Neural network colorization based on reference points	The colorization of one image takes less than 2 minutes; The process does not require human intervention.	Low quality of colorization (photos do not turn out to be full-color, most of the pictures are painted in brown tones); The image size is limited to 1 MB.
Neural network colorization	Open source and a detailed description of the principles of its operation; It does not require large processing power and can be run in a Google Colaboratory or FloydHub environment.	Low quality of colorization of most images.

Therefore, the actuality lies in developing a neural network architecture for image colorization based on existing solutions, characterized by the organization of the input space of high-dimensional features and the reduced number of layers and neurons in the hidden layers, which allows to increase the speed of image processing and maintain the required quality of processing.

3. Development of the structure of a neural network processing system for image colorization

When carrying out a computational experiment with a neural network based on [12], it was found that after the colorization some of the images lose their clarity. To improve the process of colorization, it is necessary to apply the image with selected contours to the inputs of the neural network as a source of additional information - meta-attributes, in addition to the image itself.

The solution proposed in this work is based on [12] and uses the allocation of image contours with the help of the neural network InceptionV3 [13] to improve the colorization of images through the use of meta-features.

In the proposed solution, a hint is a color image containing information that can help a neural network when coloring (for example, a similar color photo or a photo of a person presented in the main photo, painted by a expert).

If the neural network inputs is fed by the original image, its outlines, extracted features and the uncompressed image-hint, the neural network will have too many adjustable coefficients, which will lead to a significant increase in the requirements for computing resources for training and further work of the NN in color mode. It is suggested to compress images (original monochrome and image-hints), as well as submit selected outlines in a compressed form.

Thus, the original task is divided into the following subtasks:

1. Compress the original monochrome image;
2. Extract and compress the outlines from the original image;
3. Extract the signs from the image using one of the giant neural networks;
4. Compress the image-hint;
5. Train a neural network that takes inputs to the results of solving past subtasks and receives a color image output.

Thus, a generalized structure of a heterogeneous convolutional neural network is proposed (Figure 2).

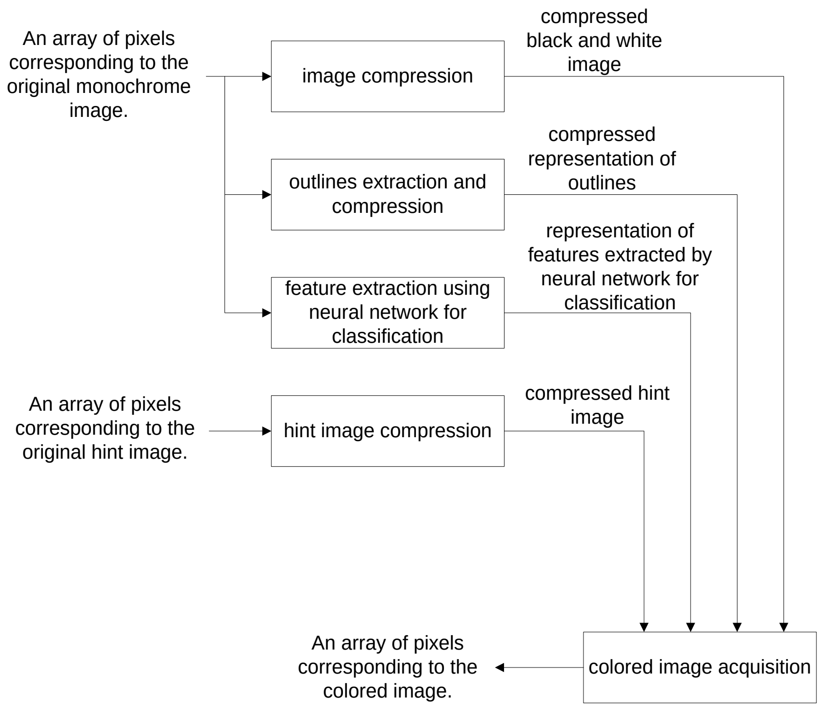

Figure 2. Generalized structure of a neural network solution.

It is important to note that the solutions obtained in solving the first four subtasks can be used to solve other problems.

3.1. Algorithms of compressing the original image

The tasks of compressing the original black-and-white image and the color hinting image are related to the tasks of information compression. It is possible to use methods that eliminate visual redundancy – information that can be deleted without compromising human perception.

A general classification and comparative analysis of image compression methods suitable for integration with subsequent neural network processing layers is shown in Figure 3 and Table 2.

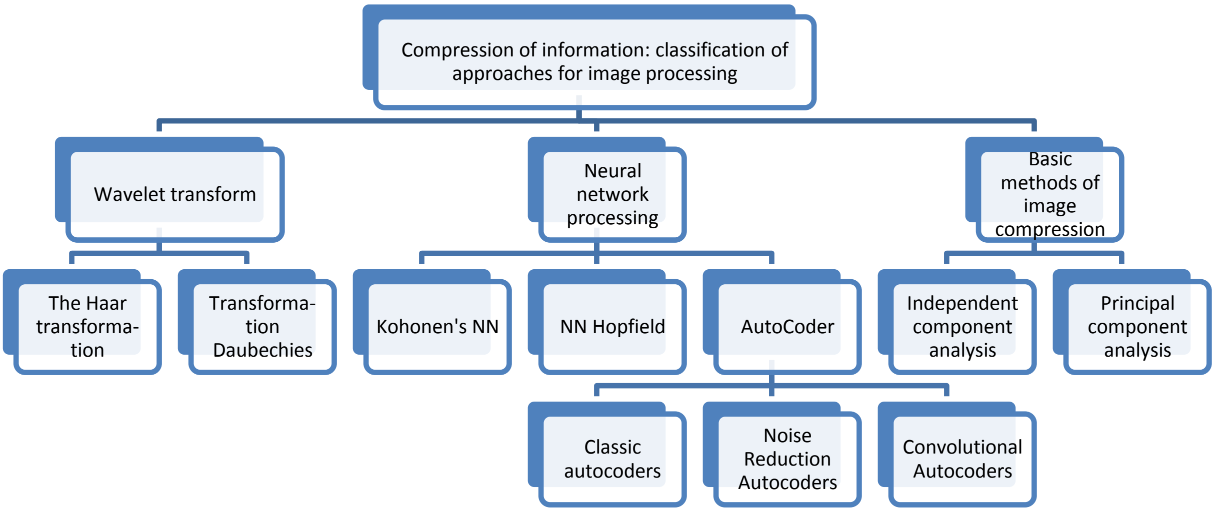

Figure 3. Classification of approaches to image compression.

When learning neural network with auto-coding, the problem of choosing the error function and optimizer arises. The most common error functions are MSE – mean squared error. Modern optimizers allow to prevent errors from reaching the local minimum, help to more evenly update the network weights and increase the speed of training. Some features can be extremely informative, but they are rare to meet. For this reason, updating the network parameters, taking into account the extent to which a typical feature represents this parameter, can make learning more effective. For this, in the Adagrad [14] Optimizer the sum of the squares of updates for each parameter is stored. The choice of the optimizer and the error function for the auto-encoder is extremely important, since this directly affects the quality and speed of the network. The empirical selection of the optimizer and the error function also seems extremely difficult, since it requires a large number of experiments that take a large amount of time. The use of the MSE error function and the optimizer Adam proved themselves in solving the colorization problem in the works of Amir Avni [9], Emil Wolner [9], Baldasar [15].

To compress the original black and white images, convolutional autocoders were used. The autocoder for image compression accepts a black and white image represented as an array. The dimensions of the original images are 512x512 pixels, so the array and the input layer of the neural network have a dimension of 512x512x1. To solve the main problem, it is necessary to compress the image up to the dimension of 128x128x1. Compression is performed using the encoder. To restore the original images in order to verify the quality of the compression, as well as the training of the encoder, it is also necessary to use a decoder.

3.2. Algorithms of selecting the image object outlines

The most popular algorithms for extracting contours are the methods of Roberts, Prewitt and Sobel, based on the use of operators. However, the resulting contour images are quite large and contain a lot of features. An autocoder could be applied to the image of contours, but data that is of value to a neural network may be lost. Also, if the filters are applied, the solution will not be homogeneous. To isolate contours and simultaneously compress them, it was decided to use an autocoder of the same structure that was used to compress the image, however, during the training of this autocoder, the outputs will be requested not for the original image but for its outlines. To extract the contours for the training sample, we use the Sobel operator, since the contours obtained by this method are the thinnest and sharpest ones.

3.3. Neural network object recognition systems

At the moment there are many neural networks for the classification of images, but the largest of them and showing consistently high results are InceptionV3, ResNet, NasNet and VGG19. The architectures of these neural networks, as well as the weights for them after training on large image databases are freely available for download. A comparative analysis of modern neural network architectures is presented in Table 3.

Table 2. Image compression approaches.

Method	Advantages and Disadvantages	Possibility of application in the colorization problem
Wavelet compression (Haar wavelet)	Areas with approximately equal brightness make up a small part of the image, zeroing of the constant part is performed	Transformations are based on the features of human perception of images;
Wavelet compression (Daubechy wavelet)	When processing with the help of neural networks, the high-frequency coefficients zeroed out at wavelet transform can carry a lot of information	Loss of features important for the neural network as the main core of the colorization system is possible.
Kohonen's Neural networks	If the number of network clusters is less than the number of different fragments of source images, then the recovery is not accurate.	When compressing arbitrary images that were not contained in the training sample, an image consisting of fragments that were in the training sample will be restored. In the problem of image colorization, the approach is not applicable.
Hopfield's Neural Networks	Application as an associative memory allows the exact reconstruction of a distorted image.	In the event of an arbitrary image submission, the image from the training sample closest to the image being fed will be restored.
Neural network autocoders	A feature is the ability to recreate the output of the same signal as the input (displays a larger space with complex connections in a space of smaller dimensions); Ability to represent diverse and complex varieties.	The most suitable are convolutional autocoders using the dropout algorithm of the convolution and sweep layers. The greatest effect when compressing images of one type, such as handwritten figures, aircraft or persons.
Noise-reduction neural network autocoders	Restore the input x not by itself, but from its noisy representation x˙. The artificial noisiness of the input data (augmentation) forces the NN to construct independent features	
Sparse neural network autocoders	Introduces a measure of dissimilarity between the distribution of attributes of input images and is added to the objective function as a regularizer	
Method	**Advantages and Disadvantages**	**Possibility of application in the colorization problem**
Conversion neural network autocoders	Built using convolutional layers in the encoder and scan layers in the decoder.	
Classical methods of dimension reduction (principal components analysis, independent components analysis)	Linear attribute systems are distinguished.	They are used when compressing images of the same type with similar characteristics.

Table 3. Features of neural networks in the task of image processing.

Neural network	Architecture	Features
VGG-19	Total number of coefficients is 144 million. Convolution with the 5x5 core is replaced by two convolutions with a 3x3 core. The saving of the number of coefficients is 22%. In case of replacing one convolutional layer 11x11 with three layers of 3x3, the savings will be 70%.	Files describing the network structure and storing its weights have a size of more than 600 MB
InceptionV3	Inception family networks are built on Inception layers and consist of layers of convolution, sweep, subsampling. Convolution layers with a 5x5 core are replaced by two 3x3 layers; The convolution layers of 3x3 are replaced by two layers of 3x1 and 1x3; The convolution architecture has been modified to avoid a sharp decrease in the dimension of the feature space.	The network achieves an accuracy of image recognition Imagenet top5 95.8%. This result is better than that shown by the person: 94.9% [13]
ResNet	ResNet is based on several initial layers with VGG-19, followed by Deep Residual Learning. ResNet uses 152 layers to predict the difference between the outputs of the last layer VGG-19 taken and the desired result.	The network contains fewer coefficients than the original VGG-19, but the ensemble of such networks set a record, the error of top5 when processing the Imagenet database was 3.57% [16].
NasNet	This neural network is created within the framework of the AutoML project for the automated creation of machine learning models. AutoML created several layers, the architecture of which has not been found before	NasNet showed results on the basis of ImageNet better than any other neural network created by man. The NasNet neural network shows classification results close to 75% accuracy, even with a small number of parameters and addition / multiplication operations, which will allow using it even in mobile devices.

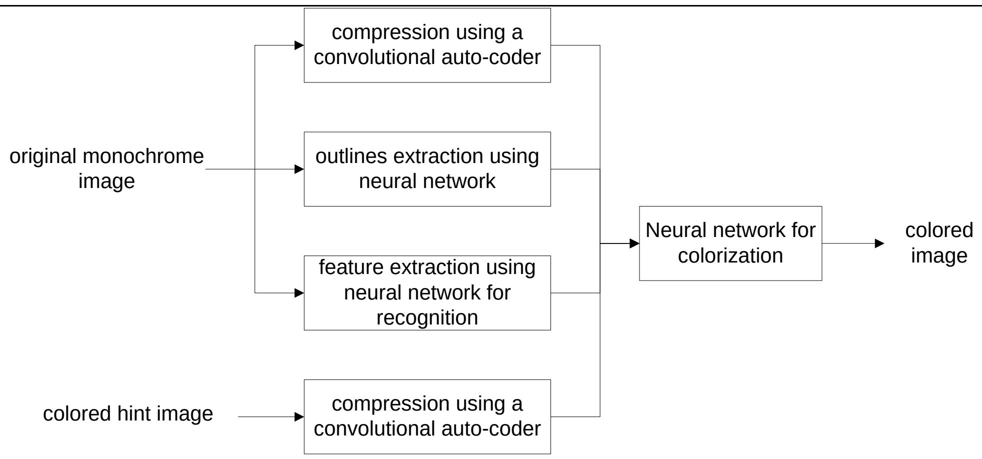

Figure 4. Neural network architecture for colorization.

To extract the features in the work, it is suggested to use the NasNet network, since it shows good classification results even with a small number of layers, and, therefore, the signs extracted with it are the most informative.

The final structure of a heterogeneous convolutional neural network for colorization is shown in Figure 4.

4. Development of algorithms for image processing using neural networks

Formation of a data set for a neural network of the selected architecture is a non-trivial task. Images from classic sets for learning neural networks, such as CIFAR-100 or STL, are too small. In Emile Wolner's decision [13], the discolored images from the Unsplash service were used to teach the neural network and its testing. These images cannot be used to learn this neural network, because the help image cannot be found. Considered the possibility of taking frames from the colorized black and white films. This idea was rejected because every film was painted by professionals in the style of the time when the film was shot and the colorization can turn unnatural. Another reason for refusing this method of obtaining data was the possible problems with copyrights. To obtain natural coloration, it was decided to search for video with a natural color transfer, and then make black and white individual frames, which will be fed as initial. As a hint, it was planned to feed frames went in the video in a few seconds. At the same time, the problem of the clarity of the original frames arose. To solve this, videos were taken that had at least 60 frames per second in the video stream. In this case, blurring when divided into frames is not so noticeable.

The number of seconds of delay between the original frame was chosen randomly in the interval from 1 to 5 to provide a different degree of similarity of frames. However, there was another problem: when training on a video containing one continuous scene, it is difficult to provide a variety of samples for training and testing. When using video collected from different scenes, there were also problems: the original frame could belong to one scene, for example, an urban landscape, and a frame-hint – another, for example, a scene shot on the sea coast. In this case, the Euclidean distance was used to select the pairs of images “original-hint” before decolorizing the original image. If it exceeded a certain threshold value, a warning was output and the frames were checked for belonging to one scene manually.

4.1. Neural network object recognition systems

Convolutional autocoders was used to compress the original black-and-white images – Table 4. The structure of the encoder is described below.

Table 4. Structure of convolutional autocoder for image compression.

Parameter	Value
Type of layers used	Convolutional, subsampling layers, layers of increasing dimension
The size of the convolution kernel	2x2
The size of the subsampling kernel	2x2
Dimension of the original image	512x512x1
Dimension of the compressed image	128x128x1
Number of learning epochs	8
Number of images in the training and validation samples	1500/500
The type of the error function (the nature of the change), Optimizer	RMS (reduction over all epochs), Adam
Activation function	ReLU - for all layers except the last one Sigmoidal - output layer
The number of weighting coefficients (total, in the autocoder)	1060356/528129

The first layer of the neural network is the input layer. The next layer is the convolution layer, this layer has 256 filters, the convolution core is 2x2 in size. Then follows the first downsampling layer, which serves to reduce the dimension. This layer has a core of dimension 2x2. At the output of this layer there are 256 signs of dimension 256x256. The next layer performs the convolution; it has a 2x2 core, as well as 128 filters. To obtain a representation of the desired dimension, a sub-sampling layer is added, having a core dimension of 2x2. The last layer of the encoder is a convolution layer with a 2x2 core, as well as a single filter. At the outputs of the last layer described, an encoded, compressed representation of the original image is removed.

The structure of the decoder has the form resembling a mirror image of the encoder structure. First, the encoded representation passes through a convolution layer, the core of which is 2x2 in size. This layer has 128 filters. Then, to increase the dimension, a layer is inserted that performs the inverse operation of the downsampling. The kernel size of this operator is 2x2. This is followed by a convolution layer, the core of which has a size of 2x2, and the number of filters is 256. Then, to obtain features of the original dimension, a dimension increase layer with a 2x2 core is used. Further, to obtain the final representation, a convolution layer with a 2x2 kernel and the number of filters equal to one is used.

Training is performed by combining the encoder and decoder into an auto-encoder. An array corresponding to the original black and white image is fed to the inputs of the auto-encoder, and the outputs require obtaining the same array. As an activation function for all layers except the last, the “ReLU” function is used. For the last layer, the sigmoidal activation function is used. The training also uses the “Adam” optimizer. The root mean square error is chosen as the error function.

The neural network was trained for eight epochs, the training sample contained 1500 images, the sample for validation had a volume of 500 images. Throughout all epochs, except the last one, a steady decrease in the error was observed, both for the training sample and during validation. The initial error value in the first epoch of learning exceeded 0.09, while by the end of the eighth epoch it was less than 0.011. The total number of coefficients for the auto-encoder is 1060356, of which 528129 are the encoder and the rest are the decoder.

The results of this neural network are shown in Figures 5 and 6.

Figure 5. Original image.

Figure 6. Image after restoration Compressing the black and white hint-image.

Convolutional autocoders was used to compress the initial color images-hints.

The results of this neural network are shown in Figures 7 and 8.

To compress the original color hint images convolutional autocoders were used. Autocoder for image compression accepts a color image as an array (RGB color space is used). The dimensions of the original images are 512x512 pixels, so the array and the input layer of the neural network have a dimension of 512x512x3. To solve the main problem, it is necessary to compress the image up to the dimension of 128x128x3. Compression is performed using the encoder. To restore the original images in order to verify the quality of the compression, as well as the training of the encoder, it is also necessary to write a decoder.

The structure of encoder is described below. The first layer of the neural network is the input layer. The next layer is the convolution layer, this layer has 768 filters, the core of the convolution is 2x2 in size. Then follows the first downsampling layer, which serves to reduce the dimension. This layer has a core of dimension 2x2. At the output of this layer there are 768 signs of dimension 256x256. The next layer performs the convolution; it has a 2x2 core, as well as 384 filters. To obtain a representation of the desired dimension, a sub-sampling layer is added, having a core dimension of 2x2. The last layer of the encoder is a convolution layer with a 2x2 core, as well as three filters. At the outputs of the last layer described, an encoded, compressed representation of the original image is removed.

Table 5. Structure of convolutional autocoder for hint image compression.

Parameter	Value
Type of layers used	Convolutional, subsampling layers, layers of increasing dimension
The size of the convolution kernel	2x2
The size of the subsampling kernel	2x2
Dimension of the original image	512x512x1
Dimension of the compressed image	128x128x1
Number of learning epochs	8 (1500)
Error (nature of change)	RMS (reduction over all epochs)
The type of the error function (the nature of the change), Optimizer	RMS (reduction over all epochs), Adam
Activation function	ReLU - for all layers except the last one Sigmoidal - output layer
The number of weighting coefficients (total, in the autocoder)	2389254/ 1194627

Figure 7. Original image.

Figure 8. Image after restoration.

The structure of the decoder has the form resembling a mirror image of the encoder structure. First, the encoded representation passes through a convolution layer, the core of which is 2x2 in size. This layer has 384 filters. Then, to increase the dimension, a layer is inserted that performs the inverse operation of the downsampling. The kernel size of this operator is 2x2. Next comes the convolution layer, the core of which has a size of 2x2, and the number of filters is 768. Then, to obtain features of the original dimension, a layer of increasing dimension with a 2x2 core is used. Further, to obtain the final representation, a convolution layer with a 2x2 core and a number of filters equal to three is used.

Training is performed by combining the encoder and decoder into an auto-encoder. An array corresponding to the original color image is fed to the inputs of the auto-encoder, and the outputs require obtaining the same array. As an activation function for all layers except the last, the “ReLU” function is used. For the last layer, the sigmoidal activation function is used. The training also uses the Adam optimizer. The root mean square error is chosen as the error function. These decisions were made after studying neural networks created by Emil Wolner [15] and Baldasar, which showed good results.

The neural network was trained for eight epochs, the training sample contained 1500 images, the sample for validation had a volume of 500 images. Throughout all epochs, except the last one, a steady decrease in the error was observed, both for the training sample and during validation. The initial error value in the first epoch of learning exceeded 0.12, while by the end of the eighth epoch it was less than

0.02. The total number of coefficients for the autocoder is 2389254, 1194627 of which are the encoder, and the rest are the decoder.

4.2. Isolating and compressing the outlines of the original image

To isolate and compress the outlines of the original black-and-white images, convolutional autocoders were used. An array corresponding to the original black-and-white image is fed at the inputs of the autocoder, and at the outputs it is required to obtain an array corresponding to the contours of the original image extracted with the help of the Sobel operator.

Table 6. Structure of convolutional autocoder for outlines compression.

Parameter	Value
Type of layers used	Convolutional, subsampling layers, layers of increasing dimension
The size of the convolution kernel	2x2
The size of the subsampling kernel	2x2
Dimension of the original image	512x512x1
Dimension of the compressed image	128x128x1
Number of learning epochs	8
Number of images in the training and validation samples	1500/500
The type of the error function (the nature of the change), Optimizer	RMS (reduction over all epochs), Adam
Activation function	ReLU - for all layers except the last one Sigmoidal - output layer
The number of weighting coefficients (total, in the autocoder)	1060356 / 528129

The results of this neural network are shown in Figures 9 and 10.

Figure 9. The contours extracted by means of the Sobel transformation.

Figure 10. The contours restored after compression.

4.3. Features selection by the NasNet network

As a result of the analysis of the NASNet neural network architecture it was concluded that the number of features needed to build a network for coloring can be extracted from the 257-th layer, if to count from the last layer of the network. This layer has the form 32x32x16, which allows it to be transformed into a layer of dimension 128x128x1, which will be convenient for forming the final input figure for the neural network for colorization.

5. Experiments on image colorization

The implementation of all structures and architectures of neural networks described in the previous chapter was performed in the Google Colaboratory environment using the Keras library. The experiments are performed according to the Table 7:

Table 7. Experiments on image colorization.

Experiment	Input data	Features
Experiment 1	Uncompressed image	A small number of images on which a color change occurred even in the case of colorization of the image of a particular class. Long learning and getting results
Experiment 2	Image compressed with the autocoder 16 times	Realistic colorization of a large number of images of the same class. Low definition of output images in some cases
Experiment 3	An image compressed with an autocoder, as well as a compressed representation of contours	Colorization is unrealistic, but reliably colored objects (sky, water) are observed. In general, the clarity of output images is higher than without using contours
Experiment 4	The image compressed by the autocoder, as well as the compressed image hint	Most of the photos colored with low accuracy. Sharpness of images is broken, not always objects are discernible by a person. In some cases, images are obtained, painted completely reliably (there are differences from the original)
Experiment 5 (6)	Original image, outlines, hint, (NASNet features) in compressed form	Colonization is absolutely unreliable. The network is uneducable.

5.1. Colorization using a fully-connected neural network

As a result of the colorization with the help of a fully connected neural network, trained on the set of “Fruits”, unrealistic images were obtained. Colorization is reduced to replacing monochrome black and white images with monochrome brown images. However, when coloring the test sample, positive results were also obtained. In particular, black-and-white photographs obtained natural dark blue shades, as well as natural shades of green when staining stems. Training neural network took a long time, this neural network of all implemented has the greatest number of coefficients, as well as addition/ multiplication operations for obtaining results – Figure 11.

Figure 11. Image from training sample “Fruits” and two output images.

5.2. Colorization with the help of convolutional autocoder

This network structure was tested on the aircraft photos of the CIFAR set. The training of this neural network was carried out in eight epochs. Training took less time than in the case of a fully connected network. The results of coloration can be characterized as good. The shades of the sky are transmitted quite accurately, realistically, the sky's coloring does not overlap planes. The color of the aircraft itself is incomplete, but distortions are not perceived by a person without viewing the original images. However, there is a part of the images, the output versions of which are very fuzzy, blurry, the detailing is much lower than the original images.

Examples of coloration using a neural network of this structure are shown in Figure 12.

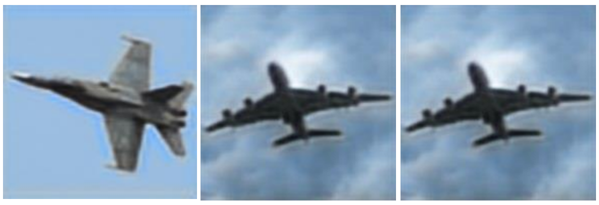

Figure 12. Examples of coloration.

5.3. Colorization using a compressed representation of images and a compressed representation of contours

Colorization with the addition of a compressed representation of the contours to the original image led to improved results. This type of colorization successfully showed itself in the photographs of aircraft, as it led to an improvement in the quality of output images and was tested on a set of arbitrary images. The resulting images have become clearer than using coloring without contours, as can be seen in Figure 13, but the color component has become less significant.

Only some areas of the sky were correctly colored. Color quality is comparable to the first works by Emil Wolner.

Figure 13. Colorization without using contours and with using contours.

5.4. Coloring an arbitrary image using a color image hint

When using a color image-hint, the color component of the output image has in many cases significantly improved. Some arbitrary photographs are painted realistically and do not cause problems in human perception. However, for photos for which the hint-image is too far away, the coloration is unnatural. Objects are blurred, sometimes unrecognizable. Also typical is the situation where a neural network “does not recognize” objects and covers the entire image in blue. Also sometimes there is a situation when the network “learns” only part of the image, spends the coloration of this part, and the rest of the image turns muddy, indistinct, and also remains black and white or acquires an unnatural color. In general, this kind of colorization gives an ambiguous result. On the one hand, this method produced the best, most natural images in some cases, but in others – the images at all ceased to be recognizable, which was not observed in other types of colorization.

Examples of coloration using hinting images are presented in Figure 14.

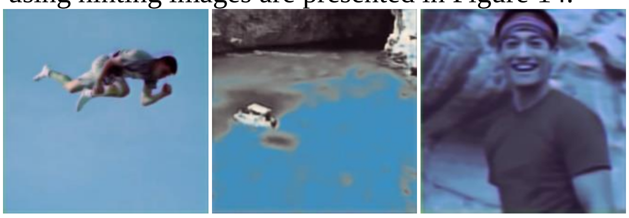

Figure 14. A successful example of coloring using hints, an example of partially correct coloring, an example of incorrect coloring.

5.5. Colorization with the help of a complete set of selected features

When using a compressed original image, a compressed representation of the contours, and a compressed image of the hint, the learning network could not be obtained. As a result of using a neural network after one learning epoch, it was discovered that the output image for any input looks like a monotonically colored square. When analyzing activities at the outputs of a neural network, one can see that there are differences in brightness, but they are insignificant and when they are rounded up to integers they are the same.

Any noticeable changes, except for increasing the learning time and obtaining results were not observed when adding to the set of input data features extracted with the NASNet network. The results of colorization are also single-color images [16-18].

6. Conclusions

The proposed algorithms for processing images based on the extraction and compression of features using neural networks for colorization of black and white images are based on the use of deep convolutional networks of a heterogeneous architecture with pre-trained modules for solving individual subtasks.

The architecture of the neural network for image colorization is developed, based on existing solutions, characterized by the organization of the input space of high dimensionality features and the reduced number of layers and neurons in the hidden layers, which allows to increase the speed of image processing and maintain the required quality of processing.

The proposed solution uses allocation of image contours with the help of the neural network InceptionV3 to improve the colorization of images through the use of metfeatures. The hint is a color image. If the original image is used in its entirety, its outlines, extracted features, and the uncompressed image-hint, the neural network will have too many adjustable coefficients, which will lead to a significant increase in the requirements for computing resources for learning and further work of the NN in the colorization mode. It is proposed to compress images (original monochrome and image-hints), as well as submit selected outlines in a compressed form, which allowed to significantly reduce the number of customized NN coefficients and reduce the requirements for computational resources.

In the future, it is possible to develop the architecture of the colorization system, which is possible by a small increase in the depth of the network, as well as the number of filters on each layer. Perhaps, other architectures should be tested, except for convolutional ones, for example, recurrent neural networks.

7. References

[1] VGG–19 in Keras URL: https://keras.io/applications/#vgg19 (10.06.2018)
[2] ResNet50 in Keras URL: https://keras.io/applications/#resnet50 (10.06.2018)
[3] NASNet in Keras URL: https://keras.io/applications/#nasnet (10.06.2018)
[4] Convolutional Layers in Keras URL: https://keras.io/layers/convolutional/ (10.06.2018)
[5] Profile of Andrey Karpaty *Official site of Stanford University* URL: https://www.cs.stanford.edu/~karpathy/ (10.06.2018)
[6] Hand Colored Films URL: http://www.widescreenmuseum.com/old–color/handtint.htm (10.06.2018)
[7] Soldatova O P, Garshin A A 2010 The use of convolutional neural network for handwriting digit recognition *Computer Optics* **34(2)** 252-259
[8] Izotov P Yu, Kazanskiy N L, Golovashkin D L and Sukhanov S V 2011 CUDA-Enable Implementation of a Neural Network Algorithm for Handwritten Digit Recognition *Optical Memory and Neural Networks (Information Optics)* **20(2)** 98-106 DOI: 10.3103/S1060992X11020032
[9] Zoev I V, Beresnev A P, Markov N G and Malchukov A N 2017 FPGA-based device for recognizing handwritten digits in images *Computer Optics* **41(6)** 938-949 DOI: 10.18287/2412-6179-2017-41-6-938-949
[10] Vizil'ter Yu V, Gorbatsevich V S, Vorotnikov A V and Kostromov N A 2017 Real-time face identification with the use of convolutional neural network and a hashing forest *Computer Optics* **41(2)** 254-265 DOI: 10.18287/2412-6179-2017-41-2-254-265
[11] AI-Powered Software for Colorizing Black and White Photos URL: https://gizmodo.com/ai–powered–software–makes–it–incredibly–easy–to–coloriz–1795298582 (10.06.2018)
[12] Colorizing B & W photos with Neural Networks URL: https://blog.floydhub.com/colorizing–b–w–photos–with–neural–networks/ (10.06.2018)
[13] Image Recognition *InceptionV3* URL: https://www.tensorflow.org/tutorials/image_recognition/ (10.06.2018)
[14] Nikolenko S, Kadurin A and Arkhangelskaya E 2017 *Deep Learning* (Immersion in the World of Neural Networks) p 480

[15] Image Colorization using CNNs and Inception–ResNet–V2 URL: https://arxiv.org /abs/1712.03400 (10.06.2018)
[16] Gonzalez R and Woods R 2005 *Digital Image Processing* (Moscow: Tehnosfera) p 1007
[17] Rangayyan R M 2015 *Biomedical signal analysis* (John Wiley & Sons) p 720
[18] Rutkovskaya D, Pilihjskij M and Rutkovskij L 2008 *Neural Networks, Genetic Algorithms and Fuzzy Systems* (Moscow: Goryachaya Liniya – Telekom)

Acknowledgments
This work was supported by the Russian Foundation for Basic Research, research № 17-08-01569.