Image Colorization

D. M. Mikhalina[1], A. A. Kuzmenko[1], K.V. Dergachev[1], V. A. Shkaberin[1]
mikhalinadasha97@gmail.com | alex-rf-32@yandex.ru | kv.dergachev@gmail.com | vash@tu-bryansk.ru
[1]Bryansk State Technical University, Bryansk, Russian Federation

The article discusses one of the latest ways to colorize a black and white image using deep learning methods. For colorization, a convolutional neural network with a large number of layers (Deep convolutional) is used, the architecture of which includes a ResNet model. This model was pre-trained on images of the ImageNet dataset. A neural network receives a black and white image and returns a colorized color. Since, due to the characteristics of ResNet, an input multiple of 255 is received, a program was written that, using frames, enlarges the image for the required size. During the operation of the neural network, the CIE Lab color model is used, which allows to separate the black and white component of the image from the color. For training the neural network, the Place 365 dataset was used, containing 365 different classes, such as animals, landscape elements, people, and so on. The training was carried out on the Nvidia GTX 1080 video card. The result was a trained neural network capable of colorizing images of any size and format. As example we had a speed of 0.08 seconds and an image of 256 by 256 pixels in size. In connection with the concept of the dataset used for training, the resulting model is focused on the recognition of natural landscapes and urban areas.

Keywords: *ResNet, convolutional neural network, CIE Lab, Place 365, image colorizing*.

1. Introduction

Nowadays, data processing automatisation is a globally urgent task. One of the directions is to automate colorizing monochrome (black and white) images. Most of coloring is now done manually, which makes this process extremely time-consuming and expensive.

Image colorization is a fundamental problem of computer graphics and machine learning. In recent years, there have been many successful works in this area. For example, in 2011 ILSVRC reached a good error-rate classification, which was 25%. In 2012 AlexNet was developed [1]. This is the first model based on 8 convolution neural networks (CNN). AlexNet got 16% of errors in ImageNet call. In the next couple of years, VG 19 [2] with 19 layers and GoogleNet [3] with 22 layers reduced the error rate to a few percent.

Although CNN made some breakthrough in accuracy, they are difficult to be trained for a number of reasons.

1. The problem of a vanishing gradient is the effect of multiplying n small numbers from the activation function to compute gradients in n-layer network, meaning that the gradient (error signal) decreases exponentially with n, thus, the front layers are trained very slowly.
2. CNN usually have a great number of parameters in their models which increase complexity, so training takes much more time.

For developing the software system of image colorization we studied a number of libraries:

- OpenCV ia a library of computer vision algorithms, image processing and numerical algorithms.
- NumPy is a library for Python, a programming language, with optimized computational algorithms for working with multidimensional data arrays.
- PyTorch is a machine learning library for Python that is used for natural language processing.

To conduct the study, a convolutional neural network was chosen, the result of which is an output image with segmented objects written in Python.

Fig. 1. Segmentation of image objects

The architecture of the neural network is based on ResNet-18 structure.

The main difference of ResNets is that it has connections parallel to conventional convolutional layers. These connections are always active, and the gradients can easily propagate through them, resulting in faster learning. ResNet with 152 layers achieves the best results with an error rate of 3% [4]. This type of deep convolutional network exceeds the human level of image classification. It allows low-, medium -, and high-level features to be extracted in an end-to-end multilayer manner, and an increase in the number of stacked layers enriches the feature "levels". Stacked layer is crucial.

The main problem when collapsing a deep network is a rapid deterioration of learning accuracy with increasing the network depth. To overcome this problem, Microsoft introduced a deep "residual" learning structure. Instead of believing that every few stacked layers directly correspond to the desired main view, they explicitly allow these layers to correspond to the "residual" ones. Formula F(x) + x can be implemented using neural networks with connections for quick access.

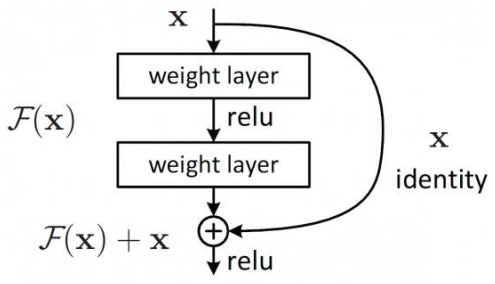

Fig. 2. Layers

2. Approach to image restoration

Let us consider images of H × W size in the colorspace CIE L * a * b *. [5]. Starting with the brightness component XL = RH = W = 1, the goal of our model is to estimate the remaining components to generate the full color version X=∈ R H=× W=×.

$$F: X_L \rightarrow \left(\tilde{X}_a, \tilde{X_b}\right).$$

In this paper we assume that there is an image F described by the equationwhere X˜a, X˜b are components a *, b * of the restored image, which together with the input give assessed colour image X˜ = (XL, X˜a, X˜b).

In order not to depend on the size of the input data, the architecture is entirely based on CNN model [6]. In short, a convolutional layer is a set of small trainable filters that correspond to specific local patterns in the input image. Layers close to the input look for simple patterns, such as contours, and layers close to the output extract more complex elements [6].

As it was mentioned above, the system selects the colorspace CIE L * a * b * to represent the input imageswhere L is the brightness channel, which is a value from black to white (from 0 to 100), and the spectrum is in the range from green to red (in values from +128 to -128), b is the spectrum in the range from blue to yellow (in values from +128 to -128) [7].

The CIE Lab color space was chosen to represent the input images, because in it the color characteristics (a, b) are separated from the brightness (L). Brightness can be considered as a black and white image, similar to that which is fed to the input of the neural network. Thanks to this color scheme, the operation of a neural network is reduced to the selection of 2 numerical values for each pixel that reflect its color.

The combination of brightness with predicted color components provides a high level of details for the final restored image.

3. Neural Network Architecture

Convolutional neural networks have partial resistance to distortion of two-dimensional images: change of angle, rotation and shift, zooming.

Currently, convolutional neural networks are considered the best in speed and accuracy of finding objects in images. Since 2012, the SNA has been number one in ImageNet.

The neural network gets an image with 3 color channels, as well as parameters such as height (H) and width (W). Then the preliminary preparation of the neural network was carried out using ResNet-18 model (ResNet is a deep neural network, 18 is a number of layers) to enter images in gray shades.

At the output of ResNet layers, a matrix of reduced size is obtained. It depends on the original image size, then the deconvolutional operation is applied to this matrix. This procedure is performed by alternating convolutional and upsampling layers. Upsampling of the matrix occurs in this way: the input matrix increases its size by duplicating its elements according to the size of the scan core due to the fact that each matrix element is projected into a larger matrix.

Then we get ResNet-18-Gray model, capable of working with images in gray shades.

The convolutional neural network has a multilayer structure and consists of convolutional, sampling and umsampling layers.

The input is a black and white image. The ResNet architecture is designed in such a way that it can only process images that are multiples of 255. To solve this problem, a program was written in Python that increases the size of the image to the required (multiple of 255 pixels) using frame extensions.

The input layer consists of cards, the number of cards depends on the type of image. If the image is color, then there will be 3 cards according to the number of color channels (red, blue, green). In our case, the image is in shades of gray, so the card will be one. Further, the input pixel values of the image are normalized in the range from 0 to 1

Convolutional layer - the convolution layer of the image. It is a set of feature cards. Each card has a synaptic core that “glides” over the entire image area, performing a multiplication operation with the input data, and then, based on the values obtained, finding certain features of the objects.

First, the values of the characteristics map of the convolutional layer are 0. The values of the weights of the kernels are set randomly within [-0.5; 0.5]. The kernel glides over the map and multiplies

The window of the kernel size passes with the given step the whole image, at each step element-by-element multiplies the contents of the window by the kernel, the result is summed and written into the result matrix.

Then the results are transferred to convolution and upsample layers, where it gradually increases with each layer to its original size.

The output is a color image from the input grayscale image. After that, the result of the work (Output) is compared with the original color image (Ground Truth).

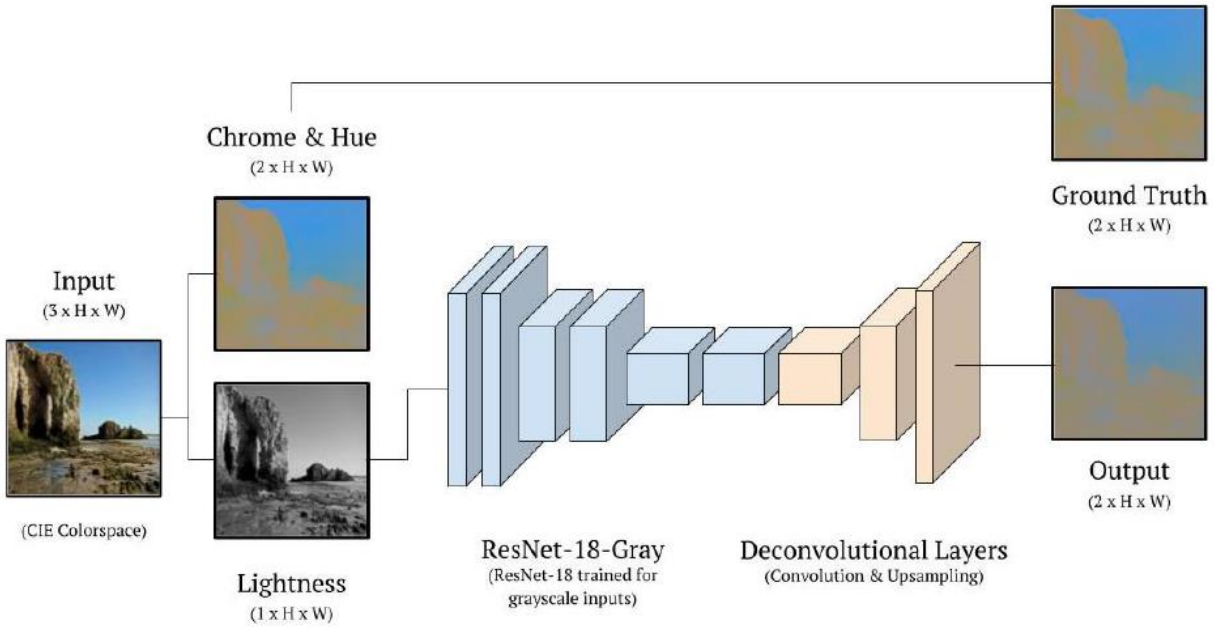

Fig. 3. Neural network architecture

The neural network under study was trained on Places365 data array, which mainly consists of images of landscapes and cities. 365 Places is built as image pairs. One is black and white and the other is colored. During learning the neural network gets this pair of images and finding certain patterns, it learns how to paint other black and white images.

4. Neural Network Training

The optimal parameters of the model are determined by minimizing the objective function defined on the basis of the expected result. To quantify the loss of the model, we use the standard error between the assumed pixel colors in space * b * and their real value. For X image MSE is defined as:

$$C(X,\theta) = \frac{1}{2HW}\sum_{k\in\{a,b\}}\sum_{i=1}^{H}\sum_{j=1}^{W}(X_{k_{i,j}} - \widetilde{X_{k_{i,j}}})^2,$$

where θ defines all parameters of models, $X_{ki,j}$ and $\tilde{X}_{ki,j}$ denote *ij* values: *th*-pixel of *k*-component: *th*-target and restored images, respectively. This can be easily extended by averaging the weight among all the images in the package.

During training, this loss propagates inversely to updating the model parameters θ using Adam Optimizer [7] with an initial learning rate η = 0.001. During training, the input image is set to a fixed size for batch processing.

Adam Optimazer is an optimization algorithm for iteratively updating the weights of a neural network based on training data. It is an improved analogue of the classical procedure of stochastic gradient descent. Its advantage is that Adam is an adaptive algorithm, that is, it calculates individual learning speeds of various parameters of the neural network, which allows you to adjust the learning speed.

5. Results Achieved

After training the neural network provided monochrome images for colorization. The results were quite good for most images. Fig. 4 illustrates the results for some examples. So the images of nature were processed with high accuracy, the colors were not distorted as close as possible to the originals, the processing speed of the photo was __ seconds. Image processing of containing people had parameters similar to those of the environment. The speed of work and the accuracy of color rendition are primarily related to the selection of photographs in which the neural network was trained, their quantity and subject matter. Tested on a subset of the Place365 dataset, ResNet-Gray achieves 75.7% accuracy. Per-pixel mean squared error (MSE) on the Places365 validation set is 0.0025 for 10 epochs and 0.0019 for 40 epochs.

For training and testing the described architecture, scripts were written using the Python language and the Pytorch library. For training, a data loader was used to load a color image, translate it into the CIE color scheme. A black and white image channel was sent to the network input. The result was compared with the original for the redistribution of the weights of the neural network. To test the work, color images were also used to visually compare the results of the neural network.

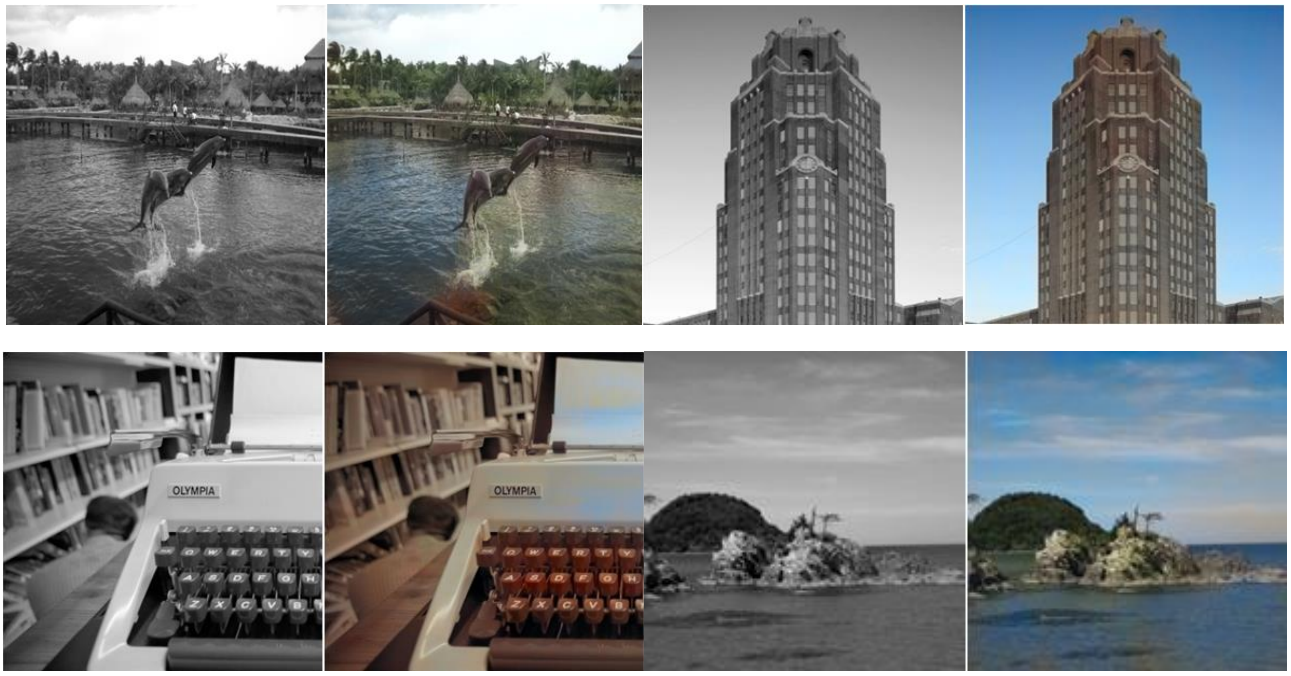

Fig. 4. The results of operation of a taught neural network

6. References

1. Zhang, Richard, Phillip Isola, and Alexei A. Efros. «Colorful image colorization» European Conference on Computer Vision. Springer International Publishing, 2016.
2. Liang, Xiangguo, et al. «Deep patch-wise colorization model for grayscale images» SIGGRAPH ASIA 2016 Technical Briefs. ACM, 2016.
3. Cheng, Zezhou, Qingxiong Yang, and Bin Sheng. «Deep colorization» Proceedings of the IEEE International Conference on Computer Vision. 2015.
4. Dahl, Ryan. «Automatic colorization» (2016).
5. Goodfellow, Ian, et al. «Generative adversarial nets» Advances in neural information processing systems. 2014.
6. Medsker, L. R., and L. C. Jain. «Recurrent neural networks» Design and Applications 5 (2001).
7. Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. Journal of machine learning research, 15(1): 1929–1958, 2014.