Grayscale Portrait Colorization with Neural Networks

Szürkeárnyalatos Portré Színezés Neurális Hálókkal

Varjas István Péter, Vitanov George

Abstract—English and Hungarian

English: We chose grayscale image colorization as our task. We decided to focus on grayscale portrait images, because most of the old pictures taken before the popularization of color photography were portraits. We used an IMDB dataset of portraits, about quarter million images as our training and testing data. We converted all images into LAB colorspace, and rescaled them to 128x128 pixels for training. We used Unet architecture as our neural network. The first five layers make up the encoder, consisting of 2D convolutional layers going from (128x128x1) to (8x8x512). Our decoder consisted of up sampling, 2D convolution, and concatenate layers, resulting in our (128x128x2) 2 color channel output. We used the concatenate layers, to connect up the encoder layers directly with the corresponding decoder layers. Both in the encoder and decoder we used ReLu activation, and batch normalization between each layer of neurons, and MAE then later on SSIM as loss functions. With this structure, we achieved decent results after 20 epochs on our database, after which we did not find any improvement. Our network currently is capable of accurately recognising human faces, and coloring them realistically, including facial characteristics, such as lips, eyebrows, and beards. The network can also distinguish between different hair colors, although sometimes mistaking gray hair for bright blond. We also achieved the realistic colorizations of different ethnicities, for example we can colorize afro Americans and Europeans accurately in the same image too. The shortcoming of our network comes in colorizing the background of the portraits. We attributed this to the very diverse artificial backgrounds contained in our dataset, thus not representing any pattern which the network could learn.

Hungarian: Feladatunknak szürkeárnyalatos képek kiszínezését választottuk. Úgy döntöttünk, hogy szürkeárnyalatos portrék kíszínezésével fogunk foglalkozni, mivel a színes fényképészet elterjedése előtt főleg portrékat készítettek. Adatbázisként egy IMDB portrékból összeválogatott adathalmazt használtunk, melyből 240000 használható képet sikerült kiszűrnünk. A jobb kezelhetőség érdekében a tanító és teszt adatainkat LAB színtérbe konvertáltuk, és átméreteztük 128x128 pixel méretűre. A feladat megvalósításához a Unet nevű neurális háló struktúrát alkalmaztuk. Encoderként 2D konvolúciós rétegeket használtunk (128x128x1) pixeles bemenettől egészen (8x8x512) konvolúcóig elmenve 5 lépésben. Dekóderként upsampling, 2D konvolúciós, és összekapcsoló rétegeket használtunk, melyek (128x128x2) kétféle szín réteg kimenetet adnak. Mind az enkóderben, mind a dekóderben ReLu aktivációs függvényt használtunk, és batch normalizációt rétegenként. A költségfüggvény kezdetben átlagos abszolút hiba volt, majd 24 epoch után a képek minőségének értékelésére kifejlesztett Structural Similarity (SSIM) metódust alkalmaztuk. Ezzel a struktúrával kielégítő eredményeket értünk el 20 tanítási epoch után, melyet követően már nem javult a háló teljesítménye. A hálónk pontosan felismeri az emberi arcokat, és élethűen kiszínezi a bőrfelületeket, az ajkakat, a szemöldököket, és szakállat. Továbbá a háló meg tudja különböztetni az eltérő hajszíneket, habár az ősz hajat néha világos szőkének véli. Ezen kívül különböző etnikumból származó embereket is pontosan tud kiszínezni, például afro-amerikaiakat és európaiakkal egy képen belül is valósághűen színezünk. A neurális hálózatunk gyengesége a portrék hátterének kiszínezésében rejlik. A hiba forrását az adatbázisunkban található rengeteg különböző mesterséges háttérnek tulajdonítjuk, mivel így nincs semmilyen minta a hátterekben melyre a hálónk rá tudna tanulni.

Index Terms—Neural Network, GAN, General Adversary Networks, Image Colorization, Portrait

1 Introduction

THE abstracts already summarized what we have done, so first lets review what this paper contains:

- Previous solutions
- The structure of our Neural Network
- Execution:
    - Data collection
    - Training and optimization
    - Results
- Possible Improvements
- Summary

2 Previous Solutions

We found numerous approaches and solutions to the grayscale image colorization problem, but none were specific to portraits, that is why we decided to build one. Bellow are some solutions which we examined:

- Colorful Image Colorization [13]
- Deep Colorization [4]
- Image-to-Image Translation with Conditional Adversarial Networks [7]
- Image Colorization with Generative Adversarial Networks [9]
- Image Colorization Using a Deep Convolutional Neural Network [10]
- A learning-based approach for automatic image and video colorization [5]

Some approaches use color labels, preset by humans on the image, to apply the selected color automatically to the segmented area [8]. We wanted a more autonomous solution without human interaction so we decided against this.

An other similar approach is, to interpret the colorization problem first as a segmentation problem, then colorize the segments [6]. We wanted a simpler solution, with the segmentation and colorization as a unified task. There are also solutions [7] which use general pretrained networks on image classification, coloring, and segmentation problems, but we chose a more specific problem to leverage its special characteristics, and using this approach would have eliminated this advantage.

We adopted two approaches from the listed solutions: we used a Convolutional Neural Network for colorization [13] [10], and experimented with training it further, as a generator network in a GAN architecture [9].

3 The structure of our neural network

3.1 U-Net

We based our neural network on the U-Net architecture [3]. The U-Net is built up from an encoder and a decoder part. We built the encoder using 2D convolution layers, increasing the resulting convolution depth in each step. The decoder consists of 2D up sampling, concatenate, and convolution layers in this order. The concatenate layers join each encoder layer to the corresponding decoder layer, to better represent our input data in the output. After each concatenate, we use 2D convolution to mix the information coming from the encoder, with that of the decoder. Experimenting during training, we found ReLu to be optimal for our needs, thus used it as activation in each layer, with batch normalization. We settled on Adam as an optimizer, and sigmoid as the output layer activation function. After the last convolution layer, we used one more concatenate layer, to join the grayscale input image data, to our two predicted color layers.

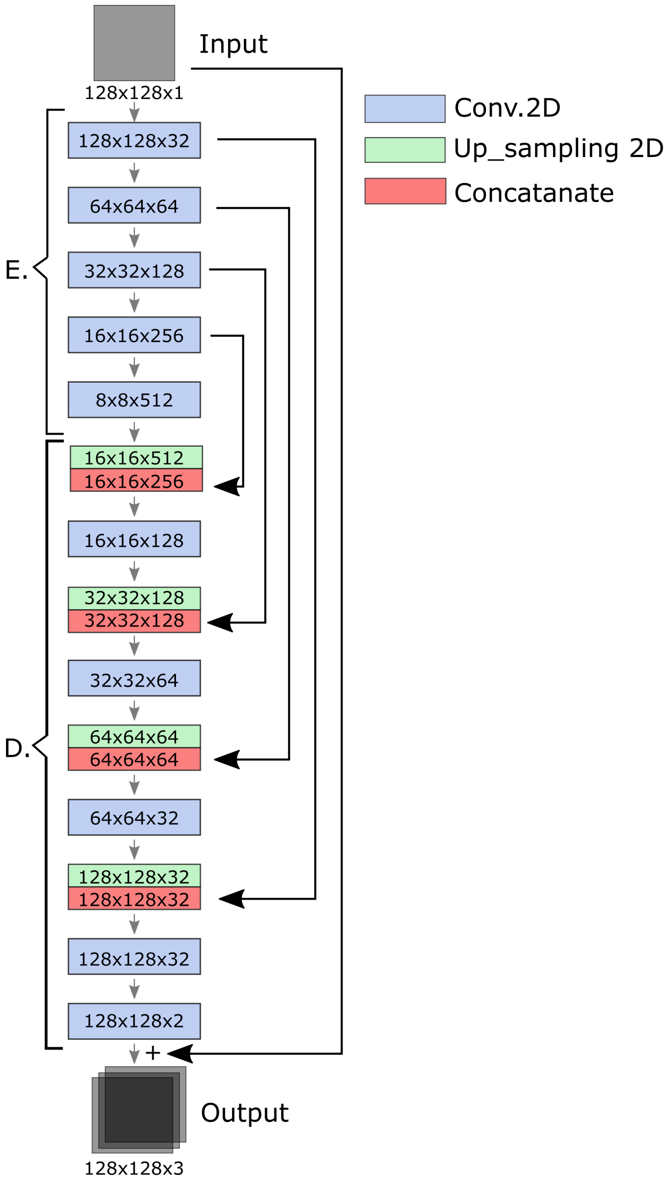

Figure 1: U-Net Neural Network Structure

3.2 GAN

After we trained our U-Net to the fullest potential we could achieve, we wanted further improvement. We decided to try, and incorporate our trained neural network into a GAN architecture. First, we trained our discriminator network, to be able to differentiate between U-Net generated and real images. Then we trained the discriminator and generator combined, in hopes, that our generator would get more creative, and try to guess more vibrant colors for the background often. The structure of our discriminator:

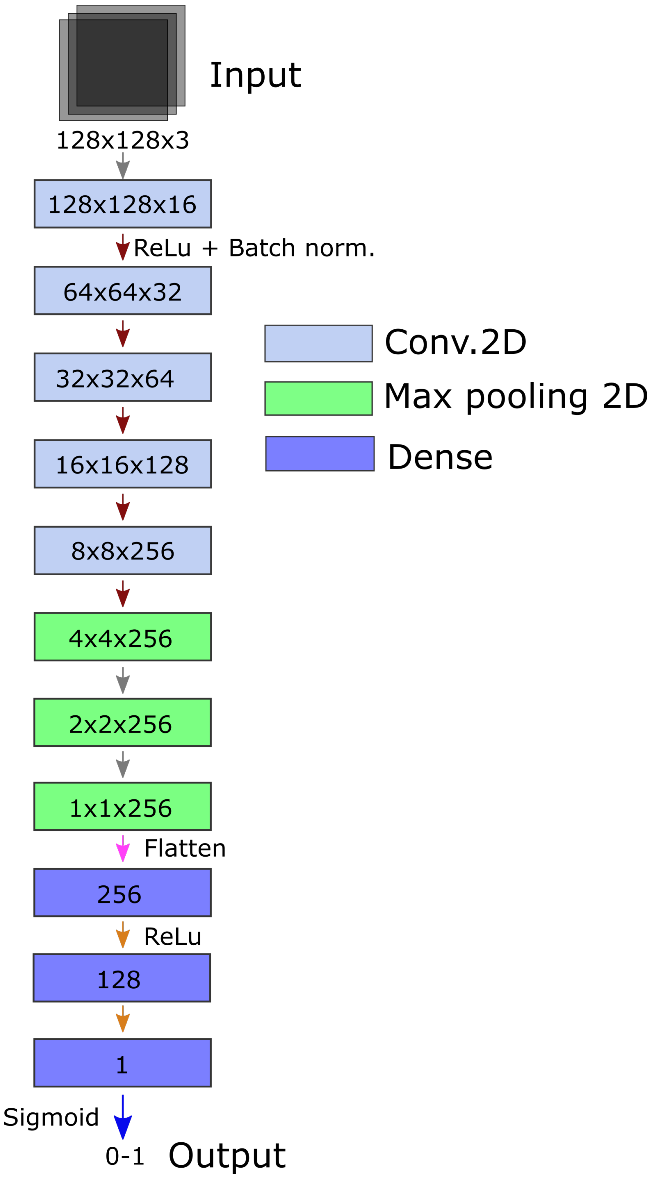

Figure 2: Discriminator network

4 Execution

4.1 Data Collection

We used an IMDB dataset [11] of portraits as our training and testing data. Properties of the database:

- IMDB database [2], with cropped images of faces [1]
- JPEG format files
- Variable sizes and aspect ratios
- Mostly small images ~10-100 kb
- Containing grayscale and faulty images ~40%!

We converted all images into LAB color space because of the advantages this presented. The Lab color space consist of 3 elements:

1) **L** – Lightness (Intensity).
2) **a** – color component ranging from Green to Magenta.
3) **b** – color component ranging from Blue to Yellow.

Figure 3: Faulty images reduced our dataset

In RGB color space the color information is separated into three channels but the same three channels also encode brightness information. On the other hand, in Lab color space, the L channel is independent of color information and encodes brightness only. The other two channels encode color. This results in the following properties:

- Perceptually uniform color space which approximates how we perceive color.
- Independent of device (capturing or displaying).
- Related to the RGB color space by a complex transformation equation.

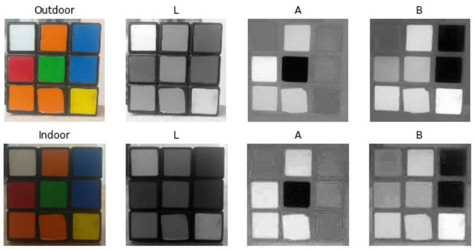

Figure 4: **LAB color space images**

We cropped and rescaled all images to 128x128 pixels for the training and testing database. Considering fast data processing and small file size important parameters, we stored our data in hdf5 format using the h5py python library. We decided to store the LAB pixel values in int format instead of floats to save space, and normalized the training data, when loaded by the generator function. This meant rescaling our input data to 0-1 scale. The necessity of this lies in the fact that we used ReLu as activation function in our Neural Network, and it would have altogether ignored negative input values.

4.2 Training and optimization

We started training our network by running test manually with different parameters, and applying the settings that produced the best results. First we started using Mean Absolute Error as our loss function. This produced good results, but after a while our network did not improve further. To counter this, we decided to switch loss functions between training epochs, and continue training with an updated loss function. We used SSIM for this purpose [12] [14].

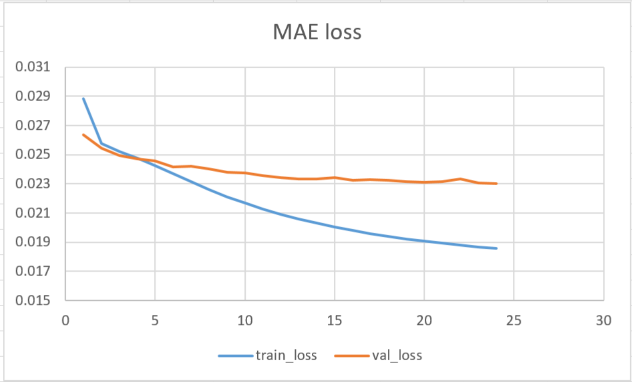

Figure 5: MAE loss function learning rates

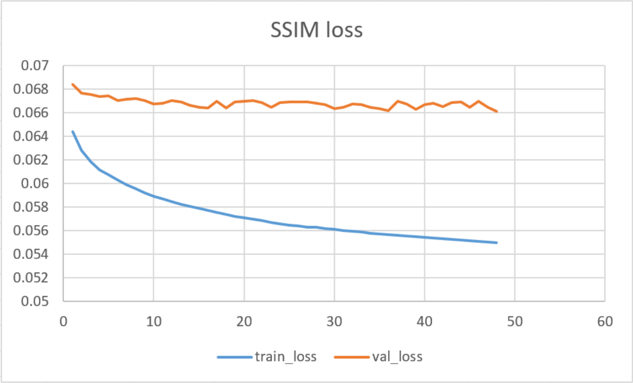

Figure 6: SSIM loss function learning rates

When training with the MAE loss function, we noticed that we were overfitting on our training database. It can be clearly seen on figure 5, that our training loss continues to decline, although the validation loss stays the same. This rather surprised us, considering our large database. To solve this issue, we implemented batch normalization after each layer as a tool against over fitting. This proved very efficient, resulting in similar training and validation loss values.

4.3 Results

For evaluating the performance of our networks, we relied on our own judgment of realistic colorizations. We managed to meet all our preset goals, namely: to achieve realistic colorization of human faces, facial features, and accurately color different skin colors. We can realistically color most formal attires, and different hair colors.

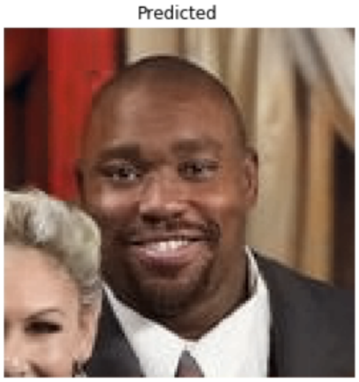

Figure 7: Coloring people with different skin colors

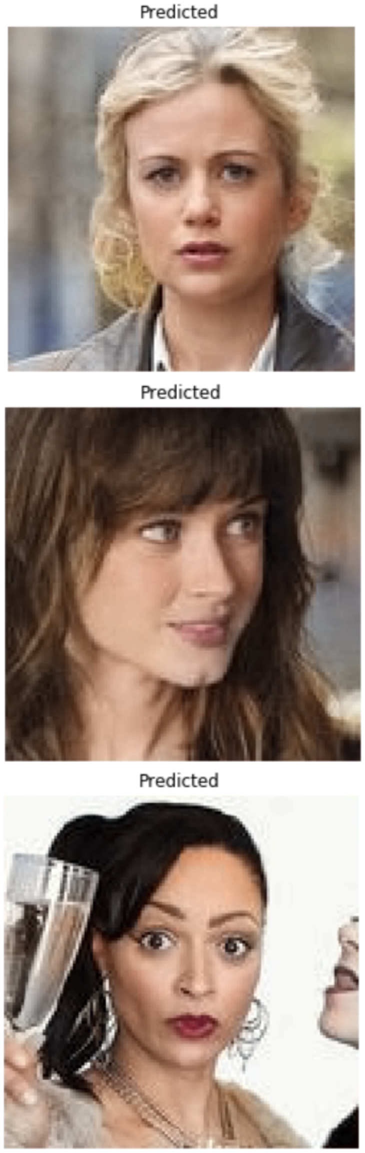

Figure 8: Different hair colors

5 Possible Improvements

Of course our solution is not perfect, there are occasions when it does not achieve perfect results. We can not color accurately rare and flashy colors, for example bright purple, because our dataset was really sparse of images containing objects colored bright purple. This is not a serious problem though, because we can still achieve realistic colorization, only in a different color. The biggest crux was trying to color the background accurately, because we were trying to achieve this without image segmentation. In some cases we succeed in this more or less, but at occasions, the network does not want to make a guess, and just leaves the background gray. The difficulty of the situation also comes from our dataset. Images of actors, often contain very various artificial backgrounds, like filming sets, photo canvas in all colors, which are very hard to guess accurately from a grayscale image.

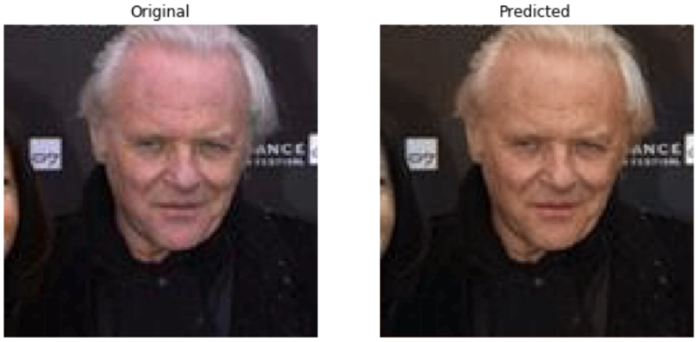

Figure 9: Gray hair mistakes

5.1 Testing

To demonstrate our results we would like to ask the reader, to guess which of the following pictures is the original, and which is the colorized one. The correct answers can be found on the next page.

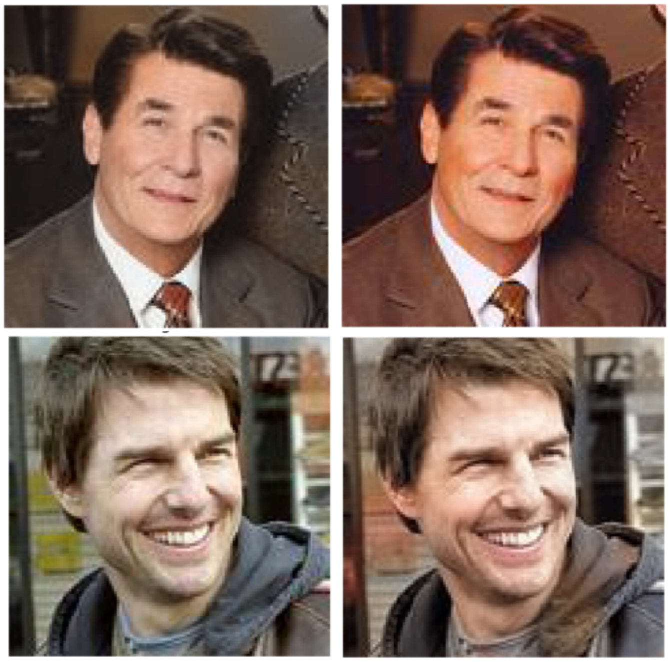

Figure 10: Predicted or Original?

The correct answers are: from the first figure, the left one is the predicted, from the second the left one is the original. We also tried coloring old photographs of scientists and famous people:

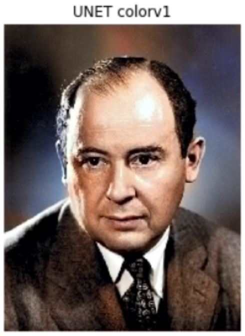

Figure 11: John von Neumann, famous Hungarian mathematician, physicist, computer scientist

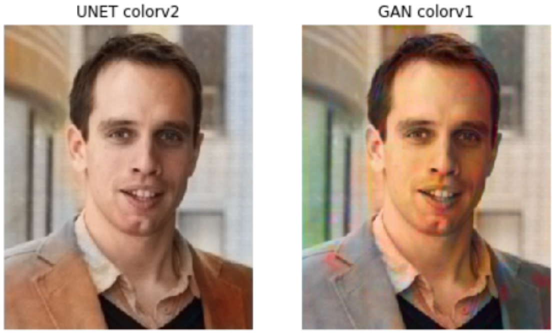

Figure 12: Dr. Gyires-Tóth Bálint, Nvidia fellow, Deep Learning professor

On this last image, we can see that our GAN architecture indeed produced more vibrant colors as we hoped, but we did not managed to achieve such realistic images from it, as produced by U-Net alone.

6 Summary

To summarize, we achieved all our preset goals of colorizing portraits. Achieving realistic colorization more than unrealistic. On occasions, our colorized images more vibrant, and more realistic, seemingly retaining better lightning conditions than the original images. We achieved these results using the U-Net structure, with batch normalization, and we are working on improving our image generation with a GAN architecture with the same U-Net network as generator. We have come a long way, but we want to improve our network in the future, to produce consistent, and better quality images than the original ones.

References

[1] Imdb-wiki – 500k face images: only faces.

[2] Imdb-wiki – 500k face images with age and gender labels.

[3] Vincent Billaut, Matthieu de Rochemonteix, and Marc Thibault. Colorunet: A convolutional classification approach to colorization, 2018.

[4] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. *CoRR*, abs/1605.00075, 2016.

[5] Raj Kumar Gupta, Alex Yong Sang Chia, Deepu Rajan, and Zhiyong Huang. A learning-based approach for automatic image and video colorization. *CoRR*, abs/1704.04610, 2017.

[6] Revital Ironi, Daniel Cohen-Or, and Dani Lischinski. Colorization by example. pages 201–210, 01 2005.

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016.

[8] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. *ACM Transactions on Graphics*, 23, 06 2004.

[9] Kamyar Nazeri and Eric Ng. Image colorization with generative adversarial networks. *CoRR*, abs/1803.05400, 2018.

[10] Tung Duc Nguyen, Kazuki Mori, and Ruck Thawonmas. Image colorization using a deep convolutional neural network. *CoRR*, abs/1604.07904, 2016.

[11] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision (IJCV)*, July 2016.

[12] Zhou Wang, Alan Bovik, Hamid Rahim Sheikh, and Eero Simoncelli. Image quality assessment: From error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13:600 – 612, 05 2004.

[13] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. *CoRR*, abs/1603.08511, 2016.

[14] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for neural networks for image processing. *CoRR*, abs/1511.08861, 2015.