Image Colorization

DELIAR MOHAMMADI, University of Calgary, Canada
XINZHOU LI, University of Calgary, Canada
JOHN ZHENG, University of Calgary, Canada

Key Words and Phrases: Computer Vison, Colorization, Convolutional Networks, U-net, Encoders, Decoders

ACM Reference Format:
Deliar Mohammadi, Xinzhou Li, and John Zheng. 2023. Image Colorization. 1, 1 (April 2023), 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 ABSTRACT

This project attempts to create a believable coloured rendition of a human picture that is in gray-scale. We created a totally automated method that generates the coloured version of an image using a modified U-Net architecture. We approached the problem as a classification task whereby each pixel can take on many different values (classes) and our model must correctly predict the class of each pixel. The model is trained on over 7000 images, each of which have unique backgrounds, humans, facial gestures and lighting. To evaluate the performance of our model we ran gray-scale images through our model and asked ourselves weather or not the image would pass as plausible or realistic in our own eyes. Roughly half the images generated by the model passed our "turing test".

2 INTRODUCTION

Image colorization has been a hot-spot in the area of computer vision for the past decade. There are many applications of image colorization such as bringing life back into old photos, colorizing historic images and videos, and even colorizing security footage shot in black and white. In the past, colorization of images and video were done entirely by human artists through the use of paint. This reliance on artists had several issues, first of all it was extremely slow to color a black & white image, each artist could only do a fixed amount of work per day and there were limited artists that were even capable of performing the task to an acceptable standard driving the cost of colorization up. In addition to cost and time, there was also the issue of perception. One artist may interpret an apple in an image as red, whereas another artist may think it's green, this led to varying results from person to person.

With the rise of deep learning and an increase in computational power, scientists were able to now use modern techniques and technology to automate this task. There have been many classical approaches using classic convolutional neural networks, however many of these fell short due to the large problem space and lack of structure being kept from the original image. The reason that the problem space is so large is due to the number of classes each pixel can fall into. Take a single pixel for examle that holds an R, G, and B value (red,blue and , green), these values fall in the range $[0, 255]$, and since we have three of them we can easily calculate the total number of possibilities to be $255^3 = 16581375$, which is an incredibly large number of classes to predict from. In order to reduce the problem space and maintain the structural integrity of an image, researchers began using the CIE-LAB color space to predict image color[Wang et al. 2022]. The L dimension represents the luminance of an image (gray-scale), the A and B dimensions represent the red-green axis and the yellow-blue axis respectively. The A and B axes values range from $[-128, 127]$ so a quick calculation tells us that there are $256^2 = 65536$ possible classes for each pixel. This drastically reduces the color space from the RGB approach. Given this property of the LAB space, we can now pass the L dimension into our network and have it output the A and B dimensions, at which point we can reconstruct the entire image by stacking the L dimension back onto AB, providing us with our final LAB image. Using the LAB colorspace also addresses our problem of not being able to keep image structure. Since the L dimension stays unchanged, the grayscale image will always be present, therefore the outputted image won't just be random noise. We found that almost all modern papers that are tackling the problem of colorization utilize the LAB color space for the reasons mentioned above. Figure one shows an example output of our model (right) compared to the ground truth image(left).

Fig. 1. Example output of our model

In the remainder of this paper we will explore colorization through a slightly more complex network architecture called CU-Net [Wang et al. 2022], which is a variation of the original U-Net architecture proposed in Ronnenberger's *Convolutional Networks for Biomedical Image Segmentation* [Ronneberger et al. 2015]. This model takes the single L channel of an image as mentioned above, and outputs the

Authors' addresses: Deliar Mohammadi, deliar.mohammadi@ucalgary.ca, University of Calgary, 2500 University Dr NW, Calgary, Alberta, Canada, T2N-1N4; Xinzhou Li, xinzhou.li@ucalgary.ca, University of Calgary, 2500 University Dr NW, Calgary, Alberta, Canada, T2N-1N4; John Zheng, john.zheng1@ucalgary.ca, University of Calgary, 2500 University Dr NW, Calgary, Alberta, Canada, T2N-1N4.

XXXX-XXXX/2023/4-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

predicted A,B channels. We will then evaluate the performance of the model by looking at the loss and sampling output images to see weather or not they can pass as real images.

3 MODEL ARCHITECTURE

The CU-net is a model based on the U-Net architecture [Wang et al. 2022]. Similar to the U-Net, downsampling layers to encode information and allow extraction of high-level features, upsampling layers to decode and increase feature resolution, and skip connections to allow the model to retrieve low-level information.

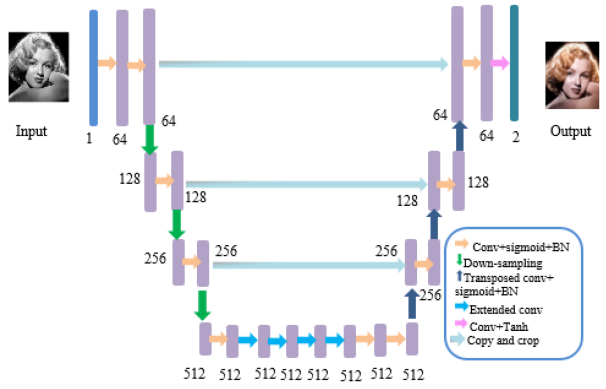

Fig. 2. CU-net overview [Wang et al. 2022].

The major difference between U-net and CU-net is the addition of extended convolution, or dilated convolution, in CU-net. An extended convolution is a convolution layer with a dilation rate, which introduces gaps in the area that the convolution covers. The intuition behind a dilated convolution is that it allows a larger receptive field than a standard convolution of the same parameter count without increasing computation time.

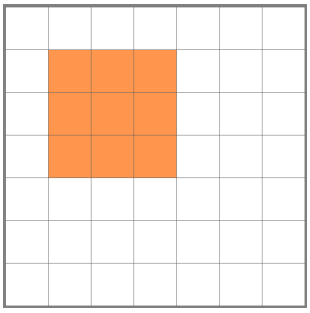

Fig. 3. Kernel of a normal 3x3 convolution without dilation.

The advantage of using extended convolution instead of successive downsampling is that extended convolution preserves feature resolution while still allowing high-level features to be learned through a large receptive field. It also maintains a similar training time to successive downsampling and upsampling.

Our model was an attempt at recreating the the CU-net as described by Wang et al. For this project, we used Pytorch to create

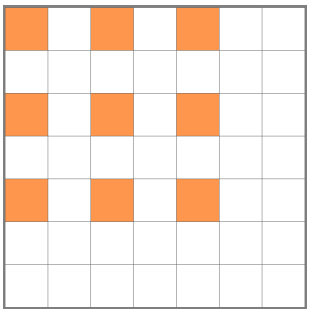

Fig. 4. Kernel of a 3x3 convolution with dilation of 1.

our model. As indicated by Figure two, we used 5 pure convolution layers in the downsampling stages, each with a 3x3 convolution followed by a sigmoid activation function and a batch normalization layer. The paper did not mention any further details of the batch normalization layer, so the default-parameter 2D batch normalization from Pytorch with learnable parameters was used.

The downsampling step in the figure actually contains a convolution, as the increase in channel number would be illogical otherwise. For every green arrow, we used a convolution layer to increase channel count (along with the sigmoid activation and the batch normalization as described), followed by a pooling operation to reduce the feature dimension. The paper suggests an average pool of 2x2 kernel size for downsampling, so that is what was used in our model.

After three downsampling steps, we reach the extended convolution part of the model which replaces the further downsampling in the standard U-net. There are four extended convolution layers used. Each extended convolution is also followed by a sigmoid and a batch normalization. Each extended convolution has 3x3 kernel count and a dilation factor of one.

After that, there are two more normal convolutions, followed by upsampling. Upsampling is done through a transpose convolution with a sigmoid and a batch normalization. The skip layers are performed by concatenating the original with the transposed layer, and then using the combined features as input to the next convolution layer of the model. Finally, there is a convolution with Tanh activation function at the end to force the model into a output range between -1 and 1.

Furthermore, due to the way that we constructed our training loop and database, the expected output was in the range between -128 and 127. This meant that one additional layer was required in our model to scale the output range to the same expected range.

4 DATASET

The data-set used for this project is entirely open-source and available for download on Kaggle [Gupta 2020]. Named "Human Faces" this dataset contains 7000 images of humans that were scraped from the web. It's also worth noting that a very small portion of these images were generated by a GAN. The data contains a very diverse range of human faces, including different hair colors, eye color, skin color, age, race, image angle, and lighting. Given the diversity and size of the data-set, it makes it much easier to build a model that can

generalize to new images. Figures 5 through 8 show several samples from our data-set

Fig. 5. A sample image from the data-set.

Fig. 6. A sample image from the data-set.

Fig. 7. A sample image from the data-set.

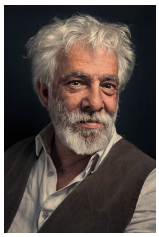

Fig. 8. A sample image from the data-set.

The images in our dataset are not ready to be fed into our network out of the box. There are a series of transformations that we apply to the images, beginning with resizing. Our network takes 256x256 images only, so they must be resized using a built-in torch function. In addition to the resizing, the images must be converted to the LAB color space, for this we used the color module from sci-kit learn which has a built in function for converting from RGB to LAB. Once the images have been resized and converted to LAB, we then extract the L channel and feed that into our network. We must keep the

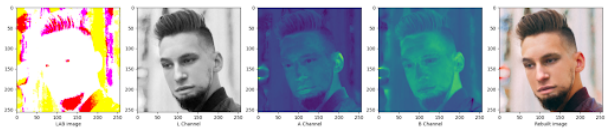

Fig. 9. The different channels of a sample image.

original AB channels since these are the ground truth's that will be used in computing our loss function later on.

Figure 9 demonstrates the entire data augmentation pipeline for a sample image. On the very far left is the LAB image, to the right of that is the L channel of the image, which is what we fill be feeding into our network. The two images to the right of that are the A and B channels respectively. And the right-most image is the ground truth image which was rebuilt by stacking the individual L,A and B layers on top of each other again.

Although this data-set is enough for the purposes of this project, it's far from the ideal dataset. In order to improve it, we would need many more images. Having more images would allow us to build an even more generalizable model that can perform well for any image with humans in it. In addition to the size of the dataset, a large portion of the images contain background, and sometimes the background can represent a large portion of the image itself. Having these backgrounds may throw off the model and cause inaccuracies in predicting colors for pixels.

5 TRAINING

During the training phase, a preliminary test was conducted with a dataset size of approximately 20 to verify the model's capability of learning colors. It was found that reasonable colors could only be produced when the loss value was reduced to below 1000, which was used as a critical criterion for subsequent training. Once it was confirmed that the model was capable of producing certain colors, training was initiated.

Initially, all categories of images were included in the dataset for training. However, it was observed that the loss value continued to increase with the number of epochs trained, and this trend did not decrease after several epochs of training. It was concluded that this was due to the dataset containing too many images, while the size of the model was too small to capture the relationship between the input and output variables accurately.

To address this issue, the dataset was replaced with a portrait-only dataset. The training data size was gradually increased from 20 until the model was underfitting, meaning the loss value could not be reduced to below 500. The training data sizes of 20, 40, 100, and 200 were tested. For the first four sizes of dataset, the loss value could be decreased in the range between 400-600 after about 400 epochs of training with the learning rate (lr) set to 0.001.

When attempting to train with a dataset size of 200, the lr was initially set to 0.001. The loss value started at 17632 and gradually decreased. After about 200 epochs loss value fluctuated at around 6000. The training was paused and the lr was reduced to 0.0001, which caused the loss value to decrease further. After 356 epochs, the loss value fluctuated steadily between 2400-2700 again. This process was repeated by setting the lr to a smaller value (0.00001). After 250

epochs of training, the loss value gradually decreased to about 1000. Despite repeated attempts to lower the lr, no significant further decrease in the loss value was observed. Therefore, the model of training dataset size of 200 was finalized.

When testing the training data size of 250, the model was underfitting , and the loss value did not decrease below 3000, regardless of how the lr and other hyper parameters were adjusted. Therefore, the upper limit of CUnet is dataset size of 200

Overall, the results suggest that a portrait-only dataset with a training data size between 100-200 and a lr of 0.0001-0.00001 is optimal for the model to achieve reasonable color generation with a loss value below 1000.

6 EVALUATION

To evaluate a colorization model, several metrics were used. One common approach is to compare the model's output with the ground truth images, i.e., the original colored images. This can be done using quantitative metrics such as mean square error (MSE).

In addition to quantitative metrics, non-quantitative evaluation methods can also be used to evaluate a colorization model. These methods often involve human perception and subjective evaluation of the model's output. One approach is to conduct user studies where human participants are asked to rate the quality of the model's output images based on their visual appeal, color accuracy, and overall quality.

The primary scoring criteria depend on the presence of any incongruous color block or color choice, with particular emphasis on skin color, hair color, and eye color. However, errors in selecting colors for clothing or background are acceptable, as these colors are not the sole focus, and clothing color options are diverse. As such, these aspects of coloring errors should not significantly reduce the overall score, as they are not the most critical points of concern. Thus, it can be argued that Mean Squared Error (MSE) is not the most reasonable metric to use, given the two quantitative scoring criteria outlined above. In cases where an image portion of background and clothes is enormous, even if the model accurately selects colors for the human face parts, the resulting MSE value may still be very high, leading to the overall failure of the coloring task. Hence, it is advisable to adopt a differential approach towards the MSE values of various parts of the color selection error. This approach will help to better reflect the accuracy and reasonableness of the model's color selection for different areas of the image, thereby enhancing the overall effectiveness of the coloring task.

7 DISCUSSION

There are many further areas of investigation, including increasing model depth for more general tasks, hyperparameter tuning, and alternative loss functions.

While we were able to achieve decent results with the 7000 images of human faces, it would be interesting to see if the CU-net can be scaled up to higher depth for general image recolorization. Additionally, it could be tested as a direct replacement for the U-Net in all of U-Net's other applications such as image segmentation.

We used the Adam optimizer with three successively lower learning rates of 0.001, 0.0001, and 0.00001. However, there may be room for improvement with a different optimizer/learning rate combination for either better results or faster training.

The loss function we used was MSE. However, that loss function may be prone to conservative colouring with less saturation than desired. One alternative is the categorization loss function [Richard Zhang 2016], though it would be interesting to see its effect on a model which is not designed for categorization tasks. Another alternative would be to use a differentiable conversion from LAB color space to to HSL and have the difference in saturation parameter directly contribute to loss.

Fig. 10. Background and face are both less saturated than they should be.

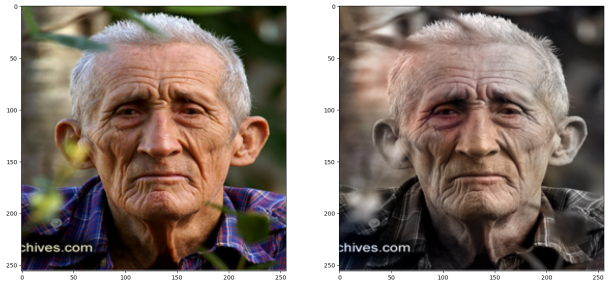

Fig. 11. The face is noticeably undersaturated. Also note that the model predicted the clothing to be gray, which is a trend across all images with clothing.

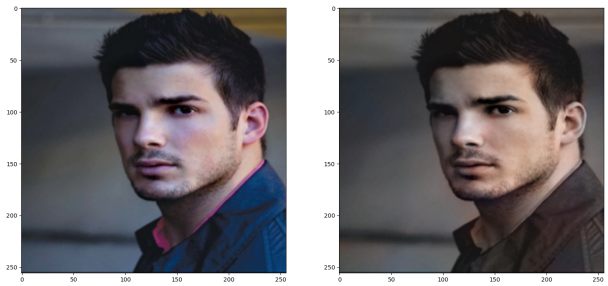

Fig. 12. Another example of an undersaturated image.

Our dataset also had some images which were black-and-white to begin with, and thus the black-and-white image was being used

as the ground truth for the model. This may have contributed to the model predicting undersaturated images. Removing these images from the dataset entirely would likely improve model performance.

8 CONCLUSION

In this paper we explored the task of image colorization with a CU-Net architecture and a dataset of 7000 human images. The images were converted to LAB space, and the L channel was fed into our network, while the AB channels were used as the ground truth for our loss function. After training for several hundred epochs, the model was able to color images well enough to pass as plausible or realistic in roughly half of all cases.

In the future, improvements can be made to this model by using much larger datasets and an increase in the number of trainable parameters by adding more down-sampling and up-sampling layers to the model. With the ability to colorize images, this now opens the door to the colorization of other forms of data such as video, astronomic photography, and human-assisted coloring of images.

Overall, it's exciting to see how advances in deep learning and computer vision are making tasks like image colorization more accessible and efficient, opening up new possibilities for historical preservation, artistic expression, and beyond.

REFERENCES

Ashwin Gupta. 2020. *Human Faces Dataset.* https://www.kaggle.com/datasets/ashwingupta3012/human-faces

Alexei A. Efros Richard Zhang, Phillip Isola. 2016. Colorful Image Colorization. 1, 1 (March 2016). https://arxiv.org/abs/1603.08511

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. Convolutional Networks for Biomedical Image Segmentation. 1, 1 (May 2015). https://arxiv.org/pdf/1505.04597.pdf

Na Wang, Guo-Dong Chen, and Ying Tian. 2022. Image Colorization Algorithm Based on Deep Learning. 1, 1 (Aug. 2022). https://www.mdpi.com/2073-8994/14/11/2295

Received 12 April 2023