Colorizing grayscale images using a Convolutional Neural Network

Emilie Klefbom, Sara Mjöberg

Chalmers University of Technology

CHALMERS UNIVERSITY OF TECHNOLOGY

Project description

The naive solution to the recolorization problem is to think of it as a regression task where the model predicts the color value for each pixel. We can implement this using CNNs and a regression loss such as mean squared error (MSE). A common issue with this type of implementation is that image predictions tend to get desaturated since MSE punish vibrant colors more harshly if they are incorrect.

Because of this we wanted to further explore the following aspects of recolorization models:

- How susceptible are recolorization modules to bias?
- Can intentionally select our dataset to create model predictions that are more vibrant but still realistic?
- Can we ensemble biased models to generate model predictions that are more vibrant but still realistic?

Model

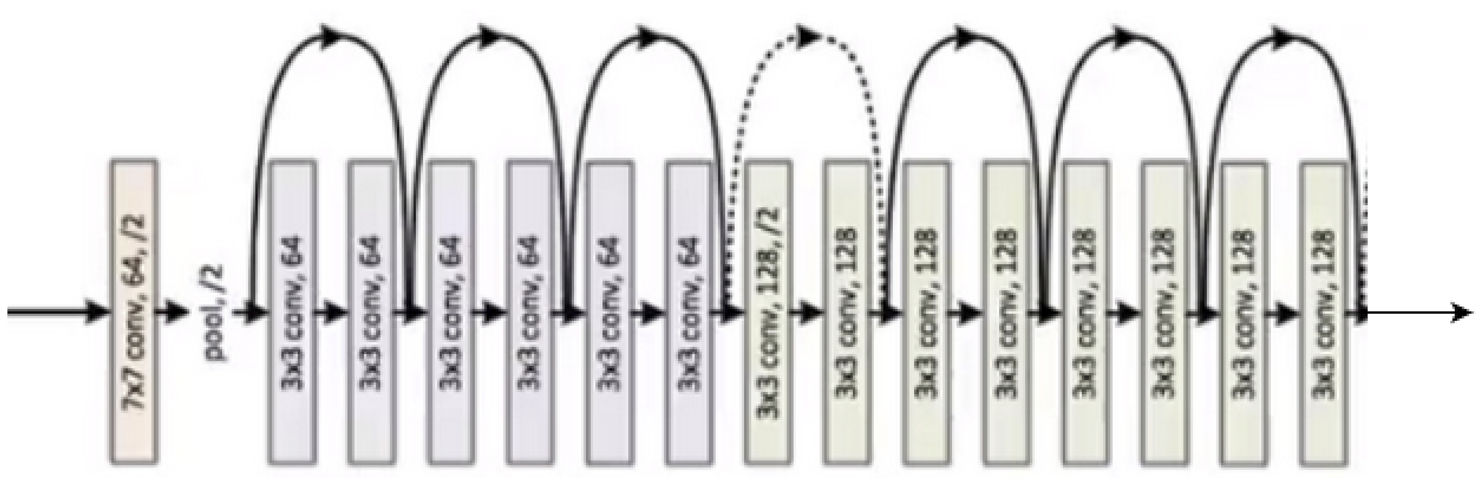

Figure 1: Resnet layers used for encoding. Note that the network takes in images of size 256x256x1 where values range from 0 to 1.

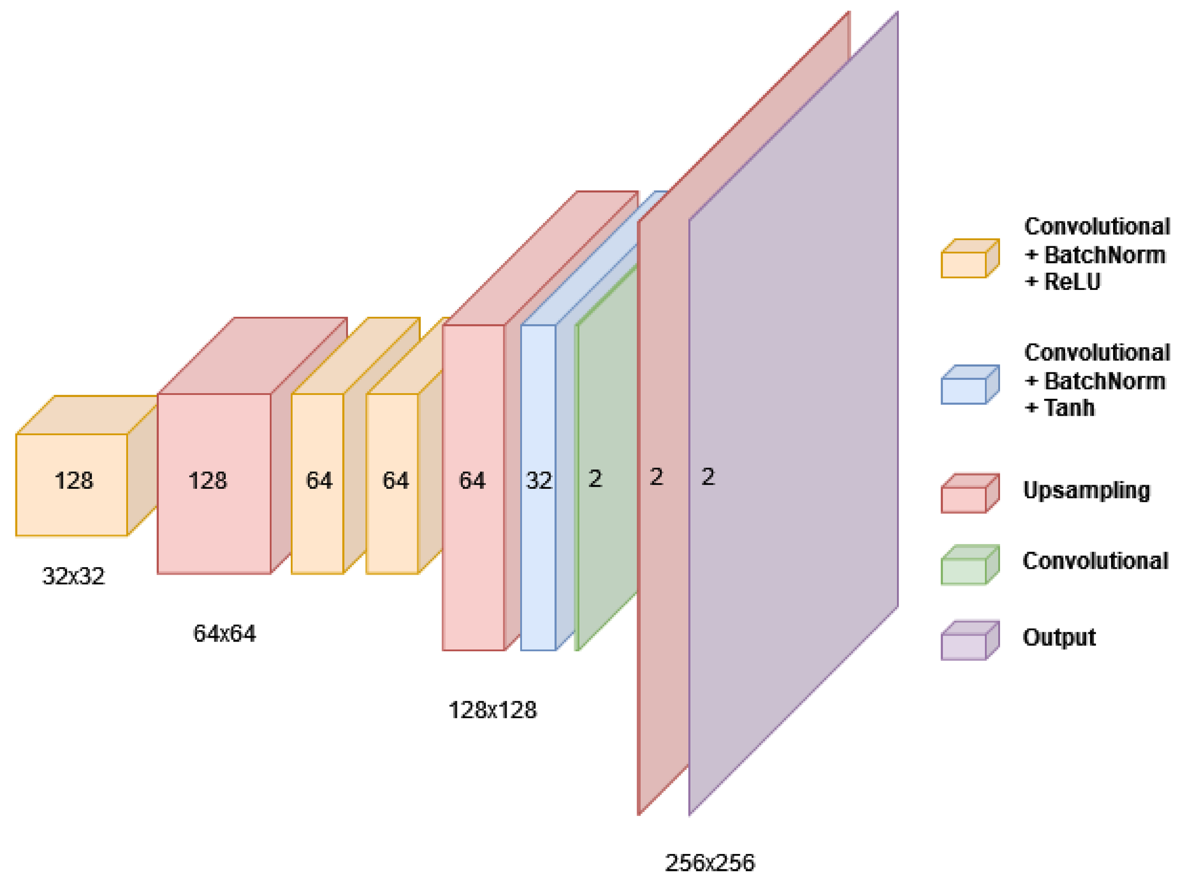

Figure 2: Cnn model used for decoding. Before training the tensors were normalized. Output tensor value, therefore, ranges between -1 to 1 since A and B values typically range between -127 to 127. This is why our final activation is tanh instead or ReLU. For upsampling the 'nearest' upsampling algorithm is used. Based on [1]

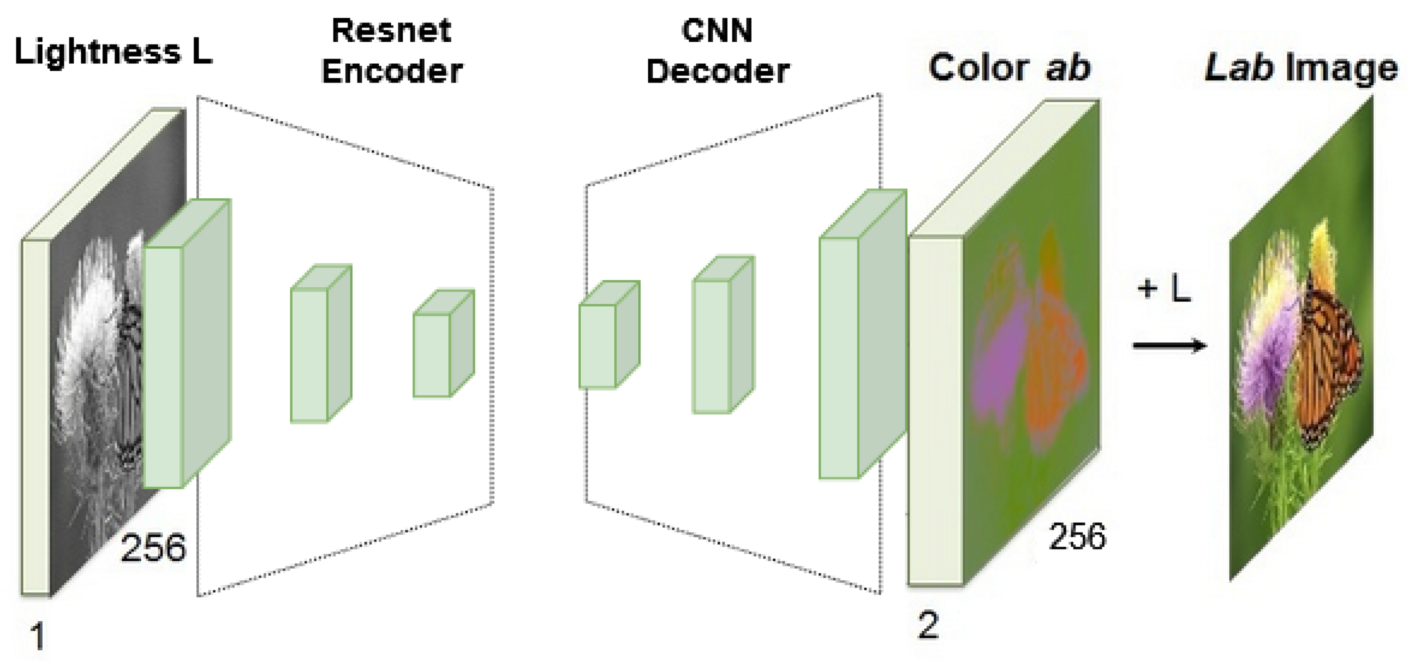

Figure 3: Overview of the full model. By combining our predicted AB output with the input L tensor the predicted image loses no information, unlike an RGB prediction where the lightness can change when we change the color. Partially based on [2]

Crash course in LAB color space

Lightness: Grayscale values
A channel: Green to red scale
B channel: Yellow to blue scale
[3]

Trained model

- Blue - trained on a blue dataset (24 blue images)
- Yellow - trained on a yellow dataset (24 yellow images)
- Green - trained on a green dataset(24 green images)
- Baseline - trained on a dataset of 864 mixed images
- BY - trained on both blue and yellow datasets
- BGY - trained on both blue, yellow, and green datasets
- BY-Combined - Average of Blue and Yellow
- BGY-Combined - Average of Blue, Yellow, and Green

Results

Figure 4: RGB and grayscale version of the test image. Dataset used [4]

This is an example where the combined model generates both a more believable but also more vivid color prediction than our baseline. Notably, the combined models are trained for the same amount of time but in total have access to a much smaller dataset

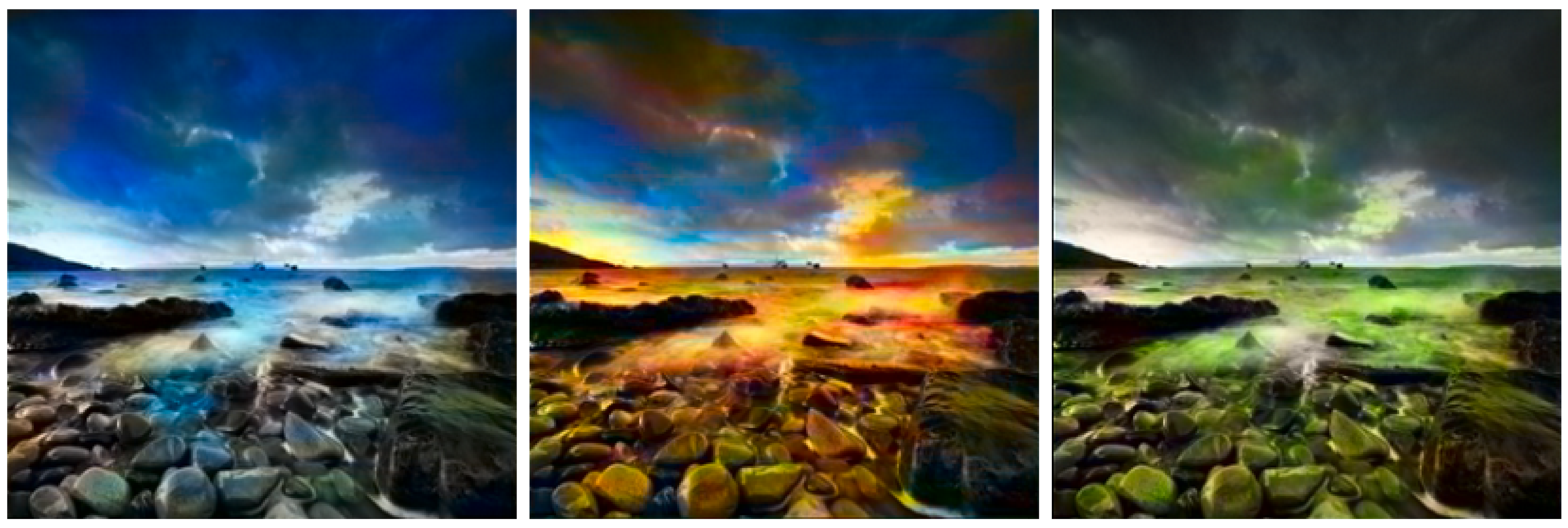

Figure 5: The blue, yellow and green model predictions.

Figure 6: The baseline prediction(left), BY prediction (middle) and BY-Combined prediction(right).

Figure 7: Top left: Original RGB image. Top right: Baseline prediction. Bottom left: BY-Combined prediction. Bottom left: BGY-Combined prediction.

When the original image was more desaturated we saw that the combined models struggled more and generated patchier images. We can also see that most models learn that the sky should be blue. This illustrates the multiplicity of recolorization problems where one object can in reality have more than one color.

References

[1] L. Melas-Kyriazi, "Image colorization with convolutional neural networks," May 2018. [Online]. Available: https://lukemelas.github.io/image-colorization.html

[2] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," *ECCV*, 2016. [Online]. Available: http://richzhang.github.io/colorization/

[3] Wikipedia, "Cielab color space," Oct 2020. [Online]. Available: https://en.wikipedia.org/wiki/CIELAB_color_space

[4] Kaggle, "Landscape pictures." [Online]. Available: https://www.kaggle.com/datasets/arnaud58/landscape-pictures?resource=download

klefbom@student.chalmers.se , saramj@student.chalmers.se