Image Colorization using Residual Networks

Stefano Fenu, Carden Bagwell

Abstract

Given a grayscale image of a natural scene, this paper presents an approach to the estimation of a recolorization of that image. While the approach presented here is fully automatic, it has an easy insertion point for human interaction to finetune results and train improved colorizers. The system as presented is a feed-forward operation in a CNN augmented with residual connections, explicitly estimating a downsampled color histogram or color palette then using this color palette to estimate a plausible coloring of an image.

1 Introduction

Color estimation for grayscale images is a very underconstrained problem, as it requires the hallucination of two additional dimensions for an image. In many cases, it is thought that the semantics of a scene and the textures of objects in the image can provide additional cues about how different parts of an image should be colored. The ocean is generally blue and grass is generally green. Priors about the semantics of natural images can greatly aid in the generation of plausible colorizations as seen in [4].

Working in the HSV color space, given V the system presented in this paper attempts to estimate H and S. While it is possible to use any color image for training such a system, by using the H and S channels as ground truth for the input of the corresponding V values, most current approaches using pixelwise prediction accuracy [1-3] tend to look dull or desaturated. This is because for the loss functions used in these systems, simply returning the grayscale image can achieve a very low loss (over 80% accuracy, as shown in Table 1). This paper explores several different loss functions that do not suffer from this problem, including total-color-variation loss and palettewise loss as

described below. The result of, while not as perceptually accurate than those in [4], do offer possible options for additional supervisory inputs from humans so as to better define what constitutes a visually plausible image.

Figure 1: Example image colorizations

2 Related Work

This section provides a brief overview of previous approaches for similar problems.

User Driven Colorization

There have been several systems proposed which would accept a reference images or metadata by which to find a reference image from a user [7], and match the coloration of the reference image by computing local pixel and superpixel similarity between the desired and reference image. These methods benefit from the easy ability to fine-tune colorization results but are not entirely automatic, being heavily reliant on user interaction for their performance.

Convolutional Neural Networks

This being the current trend in successful computer vision, showing great accuracy in large-scale image classification [8,9]. Variations to neural network architectures have also presented methods of performing dense estimates for simultaneous detection and segmentation [5], optical flow estimation [6], and image colorization [1-4].

Figure 2: Example results from Zhang et al

3 Method

Loss Function

Given a lightness channel as input X, we try to learn some mapping F to the two associated color channels Y. We denote $\bar{Y}$ to be the predictions, and Y to be the ground truth. We use HSV color space as it simplifies the intensity representation of the image. The loss function used to measure the prediction accuracy is a linear combination of L1 loss for pixel color and the L1 loss for colorspace variance.

$$L(\bar{Y}, Y) = -\frac{1}{HW}(\sum w|Y_{ij} - \bar{Y_{ij}}| + \alpha \sum w|(Y_{ij} - X_{ij})^2 - (\bar{Y_{ij}} - X_{ij})^2|)$$

In order to avoid giving excessive weight to background color regions, we also reweight the pixelwise losses inversely to the frequency with which their colors appear, w. An ideal weight for the variance loss relative to the pixelwise loss is then determined experimentally.

Explicit Palette Estimates

A simpler version of the colorization problem is to attempt to estimate the color histogram of an image, regardless of the location of the colors. By performing this estimation explicitly we can then simplify the colorization problem to that of coloring an image from a palette. To perform this estimation we use a network architecture described in the section below, with binary cross entropy as the supervisory input.

Network Architecture

Figure 2 shows the proposed architecture for the neural networks used. The weights for the convolutional layers fc1-fc5 have been drawn from a pretrained version of VGG-16 net [9] in order to establish a

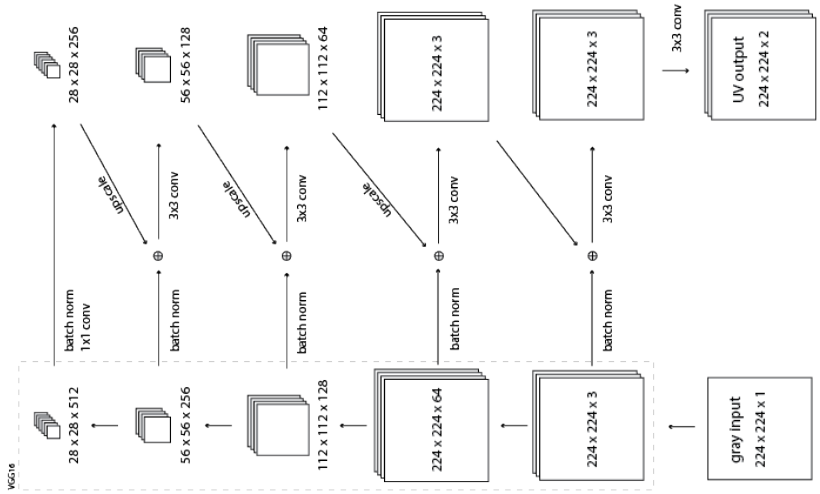

Figure 3: Residual Encoder network first used by Dahl [1]. For this paper the same overall architecture was used, with one additional layer from VGG and with additional residual connections for the estimated histograms before the batch normalization step.

1-1 comparison with the system presented in [1] and [4]. Instead of averaging the fist layer weights to the first convolutional layer as in [4], we simply pass an RGB version of the grayscale image as input to the network. In order to simplify the learning task, weights for the pretrained convolutional layers fc1-fc5 were pinned, and the supervisory input was only applied to the later stages of the network. The initial supervisory input was just the pixelwise cross entropy loss with respect to the ground truth. Residual connections in the network were introduced to allow for the explicit addition of an estimated color palette for the image, and a comparison to a network using residual connections to carry low-resolution image estimates can be seen in Figure 3. A slight increase in accuracy was found when introducing both residual connections for an estimated or human-driven palette and low resolution color estimates from previous layers. The histogram estimation is performed by a network that is very similar, with the exception that in place of the upsampling convolutional layers, it has a fully connected layer, passing output to a linear classification layer.

4 Dataset

In order to establish a 1-1 comparison with the work of Dahl [1] and Zhang et al [4], we use a subset of 1 million images from Imagenet [10] for network training.

5 Results

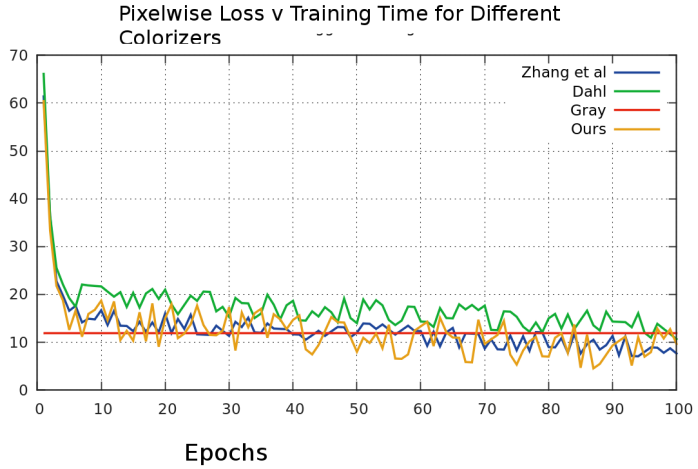

Figure 4: Pixelwise loss from different colorization methods at different stages in training

A comparison of pixelwise prediction for each of the methods is presented in Table 1. As discussed above and in [4], pixelwise prediction accuracy is not a particularly good metric for this problem, but a visual Turing test has not yet been conducted for the colorization methods presented in this paper so a more meaningful 1-1 comparison between this project and the work of Zhang et al has yet to be performed.

Figures 4-5 contain several visual comparisons of example results from the different systems used.

The method presented here also allows for manual finetuning of colorization results by editing the histograms passed along residual connections, as seen in figures 4-6.

Method	Pixelwise Accuracy
Grayscale	88.1%
Dahl	90%
Ours	90.6%
Zhang	91.1%

Figure 5: Pixelwise accuracy results are comparable but not clearly superior to the work of Zhang.

Figure 6: (Left) Grayscale image. (second left) Initial colorization (center) User-finetuned towards red. (second right) User fine-tuned towards blue. (rightmost) Ground truth

6 Conclusion

This paper presents a novel fully automatic approach to image colorization that achieves comparable accuracy to other modern approaches for the problem. The method provided also presents an easy insertion point for human interaction with the colorization system, allowing for the finetuning of image colorizations and may allow for better training by providing an easy way of generating alternative plausible colorizations for the same image. In the future it may be of interest to explore other network architectures for this problem but until a simpler way of evaluating colorization plausibility that mirrors human understanding of color, it is difficult to say what approaches to this problem present the likeliest solution.

7 References

1. Dahl, R.: Automatic colorization. In: http://tinyclouds.org/colorize/. (2016)

2. Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: Proceedings of the IEEE International Conference on Computer Vision.

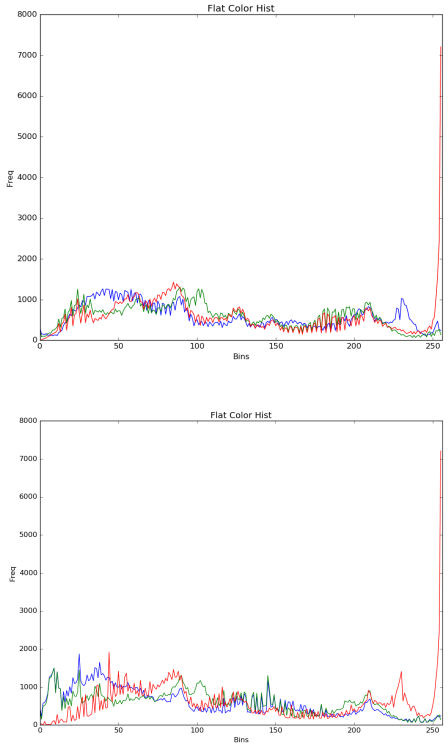

Figure 7: (Top) Estimated color histogram (Bottom) Histogram shifted towards red.

(2015) 415–423

3. Deshpande, A., Rock, J., Forsyth, D.: Learning large-scale automatic image colorization. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 567–575

4. Zhang, R., Isola, P., Efros, A.: Colorful Image Colorization. In arXiv March 2016.

5. Hariharan, B., Arbelaez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: ECCV 2014. Springer (2014)

6. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C. "DeepFlow: Large displacement optical flow with deep matching. ICCV 2013.

7. Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to greyscale images. ACM Transactions on Graphics (TOG) 21(3) (2002) 277–280

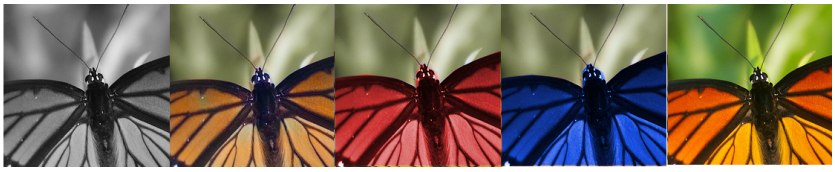

Figure 8: (Left) Grayscale image. (second left) Initial colorization (center) User-finetuned towards red. (second right) User fine-tuned towards blue. (rightmost) Ground truth

8. Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

9. Simonyan, Karen, and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).

10. Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) ImageNet Large Scale Visual Recognition Challenge. arXiv:1409.0575, 2014.