Reproducing *Colorful Image Colorization* [Zhang et al. (2016)]

Timo Nicolai Álvaro Orgaz Expósito Carolina Bianchi

May 21, 2019

Introduction (I)

Problem statement:

- Infer colours given a grayscale image
- Ill-posed problem due to inherent multimodality
  $\rightarrow$ network should predict per-pixel colour distributions
  $\rightarrow$ produce *plausible* colourization

Input

Ground truth

Colourized

Introduction (II)

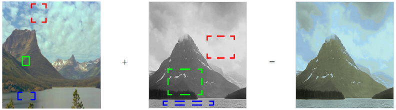
Applications:

- Colourization of historical images
- Preprocessing step in grayscale image classification
- Representation learning for transfer learning tasks

Related work (I)

Early approaches to the problem:

- Synthesise colours from reference pictures [Welsh et al. (2002)]



- Colourization as an optimization problem [Levin et al. (2004)]

Related work (II)

Modern approaches:

- Leverage large-scale data: deep learning approaches with different architectures and cost functions [Larsson et al. (2016), Iizuka et al. (2016), Zhang et al. (2016)]
- Use Generative Adversarial Network to automatically learn the cost function [Nazeri et al. (2018)]
- Exemplar-based colourization with automatic reference retrieval [He et al. (2018)]

Data (I)

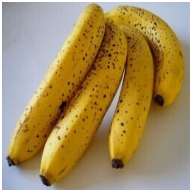
- Analyze images in the L*a*b* colour space
  $\rightarrow$ Use L channel as input
  $\rightarrow$ Use a and b channel as supervisory signals
- Resize images to 256x256 px
- Randomly crop images to 176x176 px during training

Original　L channel　a channel　b channel

Data (II)

Dataset:

- Subset of ImageNet (42.566 images)
- Semantically related categories (mostly fruits and vegetables)
  $\rightarrow$ Make training feasible given computational resources
  $\rightarrow$ Vibrant colours: easy to inspect quality of the results

Examples of images from our training set

Methods (I): Network Output

Input:

- Luminance channel L

Target:

- Discrete, in-gamut ab output space with $Q = 313$ bins

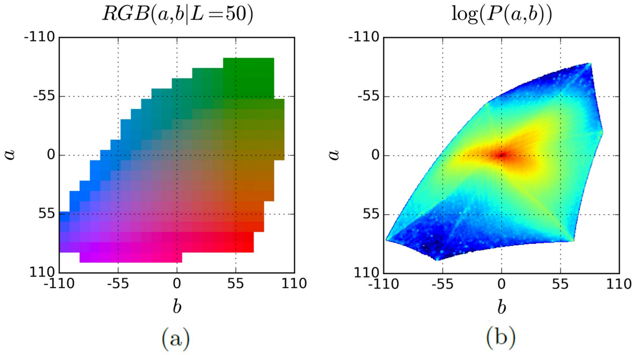


Figure from Zhang et al. (2016)

Methods (II): Loss Function

- Raw output: probability distribution $\hat{Z} \in [0,1]^{H \times W \times Q}$
- Obtain Z from ground truth via **soft encoding**

$$L(Z, \hat{Z}) = -\sum_{w,h} v(Z_{w,h}) \sum_{q} Z_{w,h,q} \cdot log(\hat{Z}_{w,h,q}) \tag{1}$$

- Use **class rebalancing** to achieve plausible colourizations

$$v(Z_{w,h}) \propto \left((1-\lambda)\tilde{\boldsymbol{p}} + \frac{\lambda}{Q}\right)^{-1} \tag{2}$$

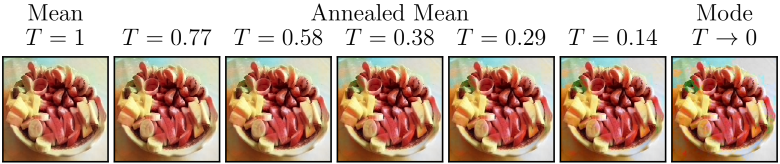
- $\tilde{\boldsymbol{p}}$ is prior distribution (from ImageNet training set)

Methods (III): From Colour Probabilities to Point Estimates

- Decode $\hat{Z}$ to $ab \in [-110, 110]^{H \times W \times 2}$ via **Annealed Mean**

$$H(Z_{h,w}) = E[f(Z_{h,w})] \qquad f(z) = \frac{exp(log(z)/T)}{\sum_q exp(log(z)/T)} \tag{3}$$

- Can adjust **Temperature** parameter $T \in [1, 0)$
  $\rightarrow$ lower T: higher vibrancy
  $\rightarrow$ higher T: higher spatial consistency



Methods (IV): Network Architecture

Input data:

- Luminance channel L

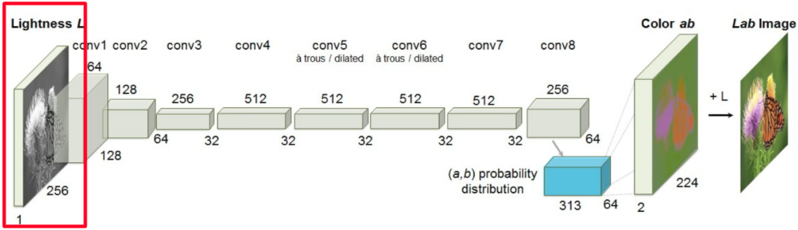


Figure from Zhang et al. (2016)

Methods (IV): Network Architecture

Feature extraction:

- VGG like network structure
- (atrous) convolution, deconvolution and batchnorm layers
- ReLU activations
- Kernel size 3×3

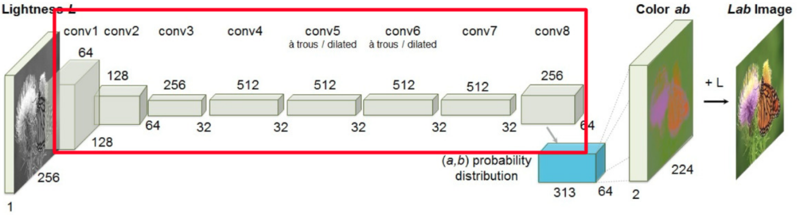


Figure from Zhang et al. (2016)

Methods (IV): Network Architecture

Raw output:

- Probability distribution $Z \in [0,1]^{H \times W \times Q}$

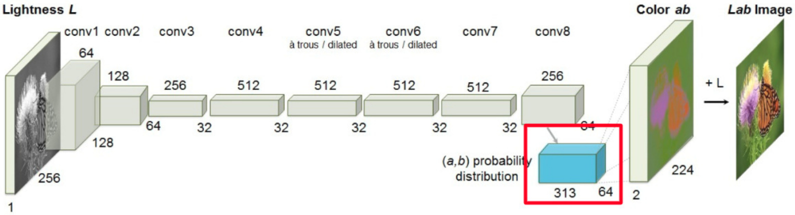


Figure from Zhang et al. (2016)

Methods (IV): Network Architecture

Annealed mean:

- $Z \in [0,1]^{H \times W \times Q} \rightarrow ab \in [-110, 110]^{H \times W \times 2}$

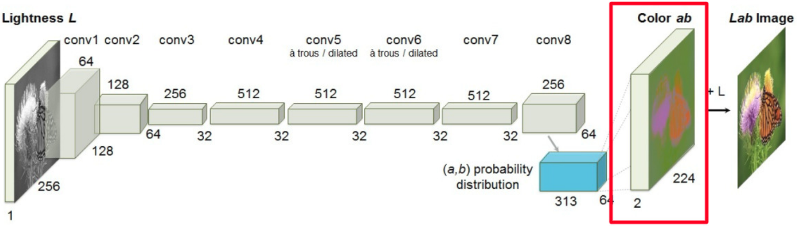
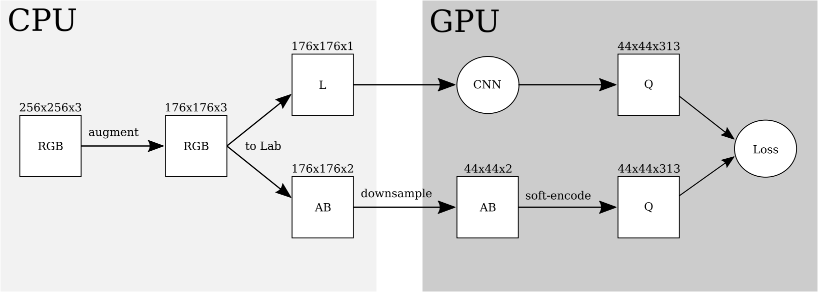


Figure from Zhang et al. (2016)

Training (I): Overview



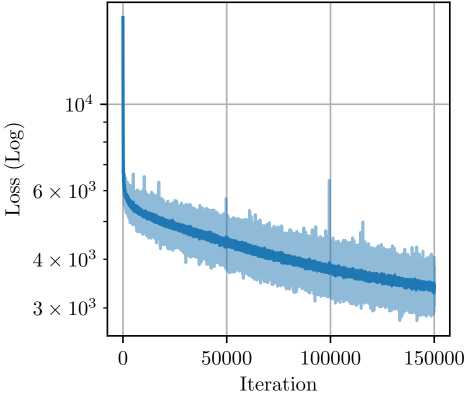
Training (II): Optimization

Adam optimizer:

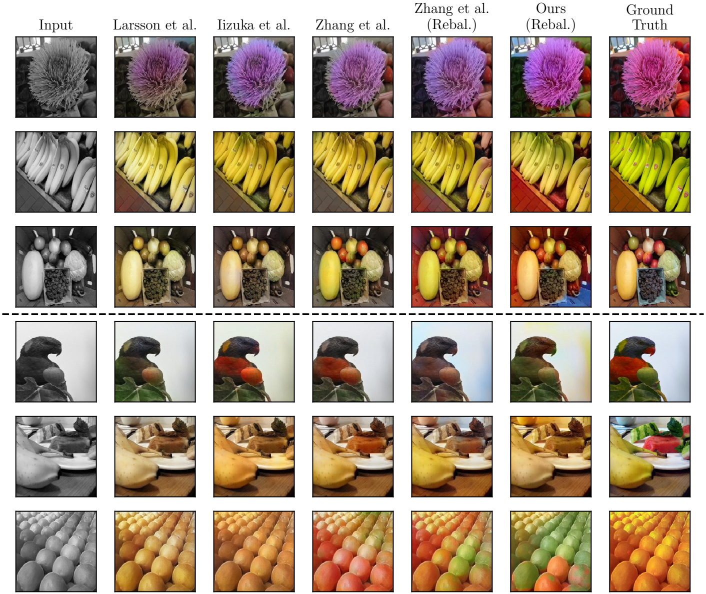
- $\beta_1 = .9$, $\beta_2 = .99$
- Weight decay $= 10^-3$
- $\eta = 3.16^{-5}$ (constant)
- Batch size $= 40$

Training (III): Learning Curve

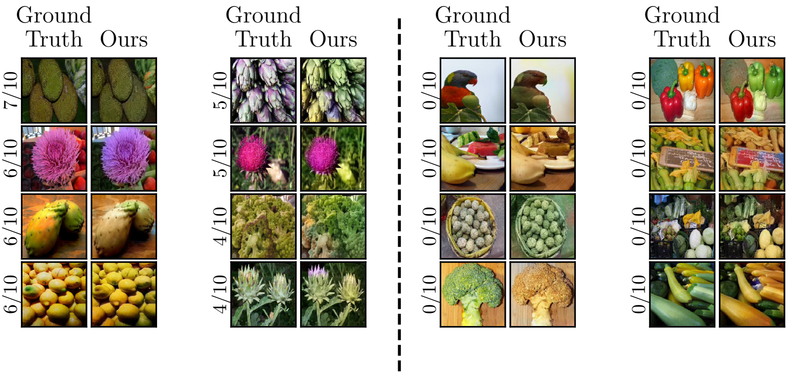
- $\approx$ 20 hours of training on NVIDIA Tesla V100



Examples


Input
Larsson et al.
Iizuka et al.
Zhang et al.
Zhang et al. (Rebal.)
Ours (Rebal.)
Ground Truth

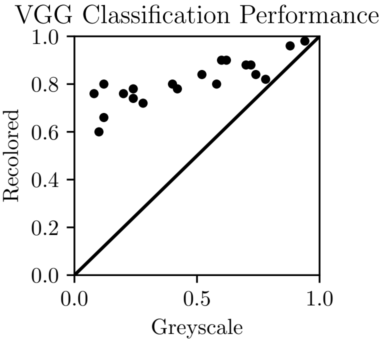
Perceptual Realism Study



- 50 randomly chosen validation set images
- 10 participants
- Fooled on average 18.78% of the time
- Photorealistic results only for “easy” images

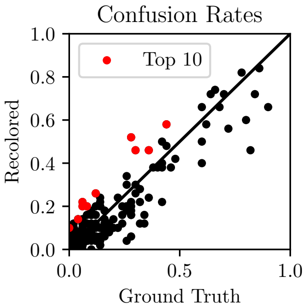
Colourization as Preprocessing (I)

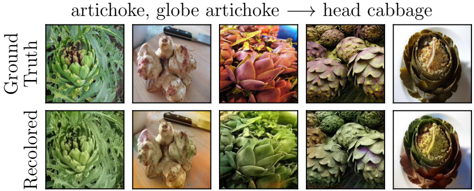
Method	VGG-16 Top-5 Acc.
Ground truth	92.1
Grayscale	46.4
Random colour	17.4
Zhang et al.	67.7
Ours	81.0



- Top-5 classification accuracy for 1000 validation set images
- $\rightarrow$ dramatically improved by colourization!

Colourization as Preprocessing (II)





- Colourization amplifies certain confusion cases

Conclusions

- We successfully reproduced the results of Zhang et al. on a reduced dataset
- We showed how our network can improve grayscale image classification accuracy
- The network tends to miscolour background objects, future work might include:
  - Exploring post-processing approaches that enforce spatial consistency
  - Segmenting images into fore- and background and colouring them with separate networks

Thank you for your attention!

References I

He, M., Chen, D., Liao, J., Sander, P. V., and Yuan, L. (2018). Deep exemplar-based colorization. 37(4):47:1–47:16.

Iizuka, S., Simo-Serra, E., and Ishikawa, H. (2016). Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. 35(4):110:1–110:11.

Larsson, G., Maire, M., and Shakhnarovich, G. (2016). Learning representations for automatic colorization. In *European Conference on Computer Vision (ECCV)*.

Levin, A., Lischinski, D., and Weiss, Y. (2004). Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 689–694. ACM.

References II

Nazeri, K., Ng, E., and Ebrahimi, M. (2018). Image colorization using generative adversarial networks. In *Articulated Motion and Deformable Objects*, pages 85–94. Springer International Publishing.

Welsh, T., Ashikhmin, M., and Mueller, K. (2002). Transferring color to greyscale images. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, pages 277–280. ACM.

Zhang, R., Isola, P., and Efros, A. A. (2016). Colorful image colorization. In *ECCV*.

Learning outcomes (I)

Timo:

- In-depth PyTorch skills, including implementing predefined architectures from common building blocks and implementing custom layers
- Increased familiarity with common CNN layers types, including (separable/transposed/atrous) convolutions, batch normalization, dropout etc.
- Deeper insight into the colourization problem: Lab colour space, input encoding and output decoding schemes, implementation and advantages of different loss functions and rebalancing schemes and how to assess colourization quality

Learning outcomes (II)

Álvaro:

- Deeper insight into the implementation of deep learning projects, concretely CNN concepts and image processing
- Useful programming skills, including PyTorch, remote Google computational resources, and Linux commands
- Customizing existing algorithms: preprocessing the target, customizing the loss function and obtaining the final prediction via an operation over the raw network output

Learning outcomes (III)

Carolina:

- Better knowledge of deep learning theory (CNN), techniques (PyTorch) and methodology
- Increased confidence with Google remote computing platforms
- Better understanding of the challenges involved in having to formulate the colourization problem in a computationally feasible way