Image Colorization and Classification

CS230: Peng Li[1], Zhefan Wang[2]
CS231A: Zhefan Wang
[1]Department of Civil and Environmental Engineering, Stanford University
[2]Department of Electrical Engineering, Stanford University
{zwang141,lipeng93}@stanford.edu

Abstract

In this work we build an automatic colorization system that takes in grayscale images and outputs visually plausible colorized images. We train the colorization network with two different CNN models. We also fine-tuned the VGG classification network [1] to test if the synthesized images can potentially improve classification accuracy and confidence.

1 Introduction

Image colorization has very useful applications such as historic photo reconstruction. We hope that realistic colorization can potentially improve the classification accuracy comparing to grayscale images due to extra information provided by colors.

For the colorization task, the input to the CNN is a grayscale image and the output is a colorized Lab space image. We try two different approaches. First, we train a regression model from scratch and optimize it with L2 loss. Second, we fit a classification model and optimize it using softmax loss. We train the classification model both from scratch and by transfer learning from a pretrained VGG classification network. Then, we compare their performance.

Next, to evaluate the result of colorization and test out if colorization can potentially improve classification accuracy, we do fine tuning on a pretrained VGG classification network that takes in an image and outputs a class label and a corresponding confidence value. We compare the accuracy of prediction among grayscale, synthesized and the ground truth color images.

The colorization network is for CS230 and the VGG classification network is for CS231A.

2 Related work

Traditional image colorization approaches [2] need human intervention to specify colors in different regions of the image (ie. scribbling). Scribble based methods can be very slow and the performance largely depends on whether the person performing the task is skillful or not.

With the rise of large-scale machine learning, recent efforts have been shifted towards automatic colorization methods. These parametric methods learn prediction functions from large datasets of color images, treating the task as either regression onto continuous color space [3, 4] or classification of quantized color values [5]. Work [6] trained conditional generative adversarial networks to model the distribution. The GAN can produce multiple realistic colorized images for a single grayscale image.

In this project, we explore CNN based regression and classification models proposed in work [7] and test the plausibility of the results using a VGG classification network.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

3 Dataset and Pre-processing

3.1 Colorization

We use CIFAR-10 dataset [8] composed of 50,000 $32 \times 32 \times 3$ RGB training images and 10,000 $32 \times 32 \times 3$ RGB test images. Figure 1 shows the classes in the dataset as well as 10 random images from each class. All colorization models are trained on the full dataset. We split the dataset such that the training set has 50,000 images, the test set has 5,000 images and the dev set has 5,000 images.

The CIFAR images are converted from RGB color space to CIE Lab color space, where L channel encodes lightness and channels a and b encode color components. Lab space is preferred because it well models perceptual distance. The L channel is normalized by 100 before fed into the network.

Both the regression model and the classification model are trained to predict the ab channels based on the L channel fed in. For the regression model, the "ab" channels are mapped to be in between 0 and 1 such that the final result can be predicted by a sigmoid function. For the classification model, the "ab" space is evenly split into 313 bins and each (a, b) pair is represented by a bin label. This is shown in Figure 2

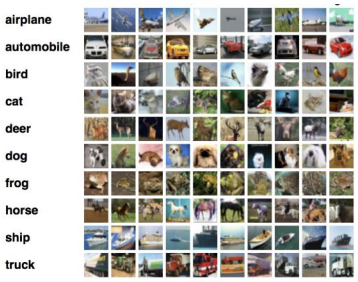


Figure 1: Classes in CIFAR-10 with 10 random images from each

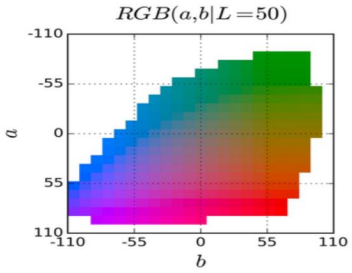


Figure 2: Quantized ab color space with a grid size of 10

3.2 VGG

For consistency, the VGG network also uses the CIFAR-10 dataset. But because the VGG network is trained by transfer learning, the data needed is much less. A training set of 10,000 images, a test set of 1,000 images and a dev set of 1,000 images is already sufficient for the VGG model to achieve descent performance. Note that input images to VGG network should be of size 224×224, thus CIFAR-10 images need to be scaled up. Then the mean RGB value is subtracted from the training images before they are fed into the network.

4 Methods

4.1 Colorization

4.1.1 Regression Model

In this approach, a CNN is trained to directly map from a grayscale image to a colored image using the architecture shown in Figure 3. Architectural details are described in Figure 4.

The loss function used is the Euclidean distance between predicted and ground truth pixel values, plus a regularization term:

$$L_2(Y, \hat{Y}) = \frac{1}{2}\sum_{h,w} \left\| Y_{h,w} - \hat{Y}_{h,w} \right\|_2^2 + \lambda \sum_{W} \|W\|_2^2$$

4.1.2 Classification Model

In this approach, we treat the problem as a multiclass classification task. A CNN is trained to map from a grayscale image to a distribution of possible colors over quantized (a, b) pairs for each pixel using the architecture shown in Figure 5. Architectural details are described in Figure 4 of section 4.1.1.

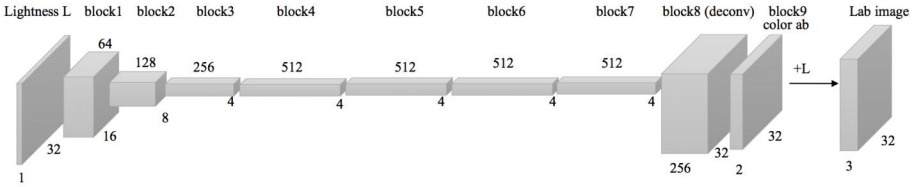


Figure 3: Regression Model

Block1	Block2	Block3	Block4	Block5	Block6	Block7	Block8	Block9
Conv3-s1-64 Conv3-s2-64 BN	Conv3-128 Conv3-s2-128 BN	Conv3-256 Conv3-256 Conv3-s2-256 BN	Conv3-512 Conv3-512 Conv3-512 BN	Conv3-d2-512 Conv3-d2-512 Conv3-d2-512 BN	Conv3-d2-512 Conv3-d2-512 Conv3-d2-512 BN	Conv3-512 Conv3-512 Conv3-512 BN	Deconv4-s2-256 Deconv4-s2-256 Deconv4-s2-256	Regression: Conv1-2 sigmoid Classification: Conv1-313 softmax

Figure 4: The convolutional/deconvolutional layer parameters are denoted as "conv/deconv <filter size> - s<stride> - d <dilation> - number of channels". If not specified, the stride or dilation is 1. The paddings of all layers are "same". Each conv layer is followed by a ReLU layer. The network has no pool layers.

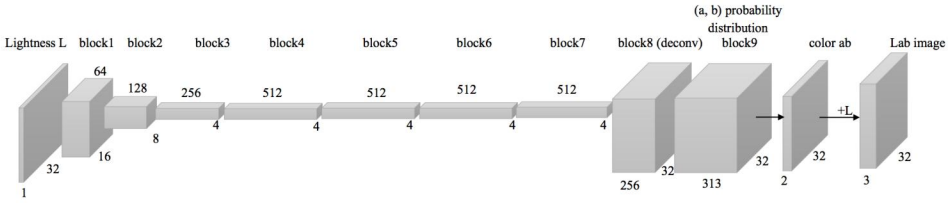


Figure 5: Classification Model

The ab output space is quantized into bins with grid size 10 and all $Q = 313$ pairs of (a, b) values that are in-gamut are kept. Given input $\mathbf{X}$, the model learns a mapping $\hat{\mathbf{Z}} = G(\mathbf{X})$ to a probability distribution over possible colors $\hat{\mathbf{Z}} \in [0, 1]^{H \times W \times Q}$, where Q is the number of quantized ab pairs [7].

To compare the predicted result against the ground truth, a function $\mathbf{Z} = {H_{gt}}^{-1}(\mathbf{Y})$ is defined. The function converts the ground truth color $\mathbf{Y}$ to vector $\mathbf{Z}$ using a soft-encoding scheme [4, 8]. The loss function used is the cross-entropy loss defined as:

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

Finally, the model maps the probability distribution $\hat{\mathbf{Z}}$ to colored output $\hat{\mathbf{Y}}$ by taking the mode of the predicted distribution for each pixel.

For this model, we train it both from scratch and by transfer learning. For transfer learning, block1 to block7 in Figure 5 are replaced with the first 5 blocks of the VGG network in Figure 6. To be consistent with the VGG network, we resize images to be $224 \times 224 \times 3$ and use the RGB representation of grayscale images instead of L channel as input.

4.1.3 Anneal-mean Technique

There are multiple ways to map the predicted distribution $\hat{\mathbf{Z}}$ to point estimate $\hat{\mathbf{Y}}$. One approach is to take the mean of the predicted distribution. However, this method produces spatially consistent but desaturated results. The reason is similar to what is explained in section 5. A better approach is to take the mode of the predicted distribution for each pixel, which is what is done in section 4.1.2. This provides vibrant but sometimes spatially inconsistent results. To get results that are both vibrant and spatially consistent, annealed-mean interpolation is implemented by re-adjusting the temperature T of the softmax distribution $\hat{\mathbf{Z}}$ and taking the mean of the result[7]. The work is inspired by paper [9]. To be more specific, the model maps the probability distribution $\hat{\mathbf{Z}}$ to colored output $\hat{\mathbf{Y}}$ with mapping $\hat{\mathbf{Y}} = H(\hat{\mathbf{Z}})$ defined as follows:

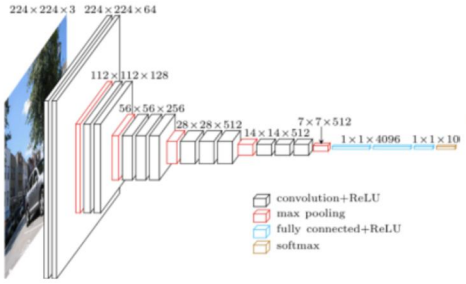
$$\mathcal{H}(\mathbf{Z}_{h,w}) = \mathbb{E}\big[f_T(\mathbf{Z}_{h,w})\big], \quad f_T(\mathbf{z}) = \frac{\exp(\log(\mathbf{z})/T)}{\sum_q \exp(\log(\mathbf{z}_q)/T)}$$

Setting $T = 1$ doesn't affect the original distribution. Smaller T produces more strongly peaked distribution. Setting $T \to 0$ is equivalent to taking the mode of the distribution. In our case, we find that $T = 0.89$ works the best at concurrently capturing both vibrancy and spatial coherence.

4.2 VGG

Given an input image, a VGG network is trained to determine which class does the input belong to. Note that CIFAR-10 images need to be scaled from $32 \times 32 \times 3$ to $224 \times 224 \times 3$ before feeding into the VGG network. Besides, the original VGG in work [1] was trained on ImageNet, which contains 1,000 classes. But there are only 10 classes in CIFAR-10 dataset. Therefore, the last block of the architecture is slightly different from the original VGG network. When performing transfer learning, block1 to block5 use the pre-trained weights provided by [1] and block6 is trained from scratch.

The architecture and corresponding architectural details are described in Figure 6.



Block1	Block2	Block3	Block4	Block5	Block6
Conv3-64	Conv3-128	Conv3-256	Conv3-512	Conv3-512	FC-4096
Conv3-64	Conv3-128	Conv3-256	Conv3-512	Conv3-512	FC-4096
Maxpool	Maxpool	Conv3-256	Conv3-512	Conv3-512	FC-10
		Maxpool	Maxpool	Maxpool	Softmax

Note: The convolutional layer parameters are denoted as "conv(kernel size)-(number of channels)". The ReLU activation function is not shown for brevity.

Figure 6: VGG Model

The loss function is the softmax loss between the predicted class and the ground truth class:

$$L(Y, \hat{Y}) = -\sum_{q=1}^{10} Y_q log\ (\hat{Y}_q)$$

5 Experiments and Results

5.1 Colorization

We use VGG classification accuracy as the evaluation metric for the colorization network. All test set grayscale images, results generated by the regression model, results generated by the classification model, colored images with anneal-mean applied, and the ground truth colored images are fed into the VGG network. As one can see from the summary in Figure 7, all colorization models successfully increase the classification accuracy. Among all the algorithms we implemented, the classification model with anneal-mean achieves highest accuracy.

	Grayscale	Regression	Classification	Annealed	Transfer Learning	Ground Truth
Classification Accuracy	22%	46%	57%	65%	63%	80%

Figure 7: Colorization and VGG Results

We also visually evaluated the various results. The upper-left part of Figure 8 shows a sample of the results of various colorization models. The first row is a selection of the grayscale input images, and last row is the corresponding ground truth color images. Row 2 are outputs from the regression model. Row 3 are outputs from the classification model trained from scratch. Row 4 are outputs from the classification model trained from scratch with annealed mean technique applied. Row 5 are outputs from the classification model trained using transfer learning from VGG model.

For the regression model, learning rate is 10^{-4}, training set batch size is 128, dev and test set batch size is 64, and there are a total of 100 training epochs. As one might notice, comparing to the classification model, the regression model cannot generalize very well and outputs many grayish images (For example, row 2 column 2 and row 2 column 11). This is because the goal of the

Truth	Grayscale	Regression	Classification	Annealed	Transfer Learning	Ground Truth
horse	airplane (25%)	dog (88%)	dog (23%)	dog (70%)	cat (41%)	horse (37%)
deer	cat (24%)	frog (41%)	cat (35%)	frog (57%)	cat (48%)	deer (67%)
horse	dog (23%)	dog (51 %)	horse (95%)	horse(90%)	horse(97%)	horse (99%)
frog	airplane (25%)	frog (77%)	frog (78%)	frog (87%)	frog (81%)	frog (91%)
horse	cat (20%)	deer (66%)	horse (84%)	horse(90%)	horse(88 %)	horse (99%)

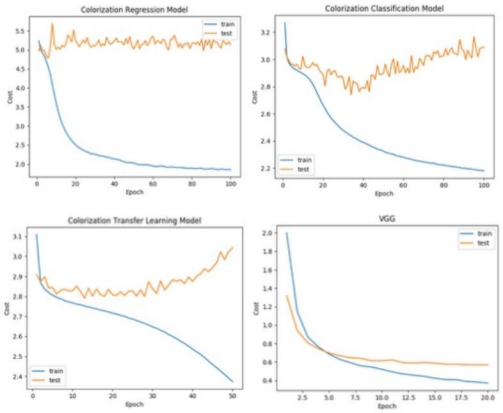


Figure 8: Colorization and VGG Results

regression model is to minimize Euclidean distance between the prediction and the ground truth. Therefore, L2 loss encourages conservative predictions and is not robust to the multi-modal nature of the colorization task. A grayscale apple can be colored as either red, green or yellow. In such a case, the optimal solution to the L2 loss is to take the mean of the three colors. In color prediction, taking average leads to grayish pixels.

For the classification model, learning rate is 10^{-4}, training set batch size is 128, and dev and test set batch size is 64. There are a total of 100 training epochs if to train from scratch and a total of 50 epochs if to train by transfer learning. $\lambda = 0.01$ for regularization to prevent overfitting. The classification model appropriately models the multi-modal nature of the colorization problem. In general, this model generates very satisfying results and training from scratch and training by transfer learning achieves equally realistic images.

Finally, taking the annealed-mean of the prediction distribution gives the best plausible results (row 4). Re-adjusting the temperature and taking the average results in more vibrant and spatially coherent results. In our case, $T = 0.89$ works the best. One can notice huge improvement in row 4 column 2 (the frog) compared to row 3 column 2 and row 5 column 2. Also, the blob around the airplane in row 3 column 10 disappeared after applying annealed-mean.

5.2 VGG

For the VGG network, learning rate is 10^{-4}, training set batch size is 128, dev and test set batch size is 64, and there are a total of 20 training epochs. The lower left part of Figure 8 shows the classification results of the VGG model. The percentage in parenthesis represents confidence of the corresponding prediction. As one can see, adding in color information in general boosts prediction accuracy and confidence. Take row 4 of the classification images as an example. The initial prediction on the grayscale image is wrong. After colorized the image, the prediction is correct. The confidence of the prediction gets even higher after annealed-mean is applied.

6 Conclusion and Future Work

For the colorization task, two different models are built and the results indicate that classification model outperforms the regression model. The classification model treats the problem as multinomial classification and resolves the multi-modal nature of colorization task, whereas the regression model emphasizes more on minimizing the Euclidean distance and thus favors grayish desaturated images.

We also find that Annealed-mean technique can effectively produce both vibrant and spatially more consistent colorization results.

The results of VGG network shows that colorization in general boosts object classification accuracy and confidence.

For the future work, we would like to train our models on larger datasets such as ImageNet. We would also like to implement a conditional GAN to perform the colorization task.

Code for the project can be found at Github: $https://github.com/1ipeng/CS230\_Project$

7 Contributions

VGG: Zhefan Wang
Data Prepocess: Peng Li
Anneal-mean: Zhefan Wang, Peng Li
Colorization Model: Zhefan Wang, Peng Li
Colorization Training and Test: Peng Li

References

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[2] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu, "An adaptive edge detection based colorization algorithm and its applications," *Proceedings of the 13th annual ACM international conference on Multimedia - MULTIMEDIA 05*, 2005.

[3] A. Deshpande, J. Rock, and D. Forsyth, "Learning large-scale automatic image colorization," *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[4] R. Dahl, "automatic colorization."

[5] G. Charpiat, M. Hofmann, and B. Schölkopf, "Automatic image colorization via multimodal predictions," *Lecture Notes in Computer Science Computer Vision – ECCV 2008*, p. 126–139, 2008.

[6] Y. Cao, Z. Zhou, W. Zhang, and Y. Yu, "Unsupervised diverse colorization via generative adversarial networks," *Machine Learning and Knowledge Discovery in Databases Lecture Notes in Computer Science*, p. 151–166, 2017.

[7] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," *Computer Vision – ECCV 2016 Lecture Notes in Computer Science*, p. 649–666, 2016.

[8] A. Krizhevsky, "cifar-10 and cifar-100 dataset."

[9] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Readings in Computer Vision*, p. 606–615, 1987.

[10] "using photoshop and color management for printing."

[11] D. Frossard, "Vgg in tensorflow," Jun 2016.