Colorful Image Colorization

Richard Zhang, Phillip Isola, Alexei A. Efros
{rich.zhang, isola, efros}@eecs.berkeley.edu

University of California, Berkeley

Abstract. Given a grayscale photograph as input, this paper attacks the problem of hallucinating a *plausible* color version of the photograph. This problem is clearly underconstrained, so previous approaches have either relied on significant user interaction or resulted in desaturated colorizations. We propose a fully automatic approach that produces vibrant and realistic colorizations. We embrace the underlying uncertainty of the problem by posing it as a classification task and explore using class-rebalancing at training time to increase the diversity of colors in the result. The system is implemented as a feed-forward operation in a CNN at test time and is trained on over a million color images. We evaluate our algorithm using a "colorization Turing test", asking human subjects to choose between a generated and ground truth color image. Our method successfully fools humans 20% of the time, significantly higher than previous methods.

Keywords: Colorization, Vision for Graphics, CNNs

1 Introduction

Consider the grayscale photographs in the top row of Figure 1. At first glance, hallucinating their color seems daunting, since so much of the information (two out of the three dimensions!) has been lost. Looking more closely, however, one notices that in many cases, the semantics of the scene and its surface texture provide ample cues for many regions in each image: the grass is typically green, the sky is typically blue, and the dog's tongue is most definitely red. Of course, these kinds of semantic priors do not work for everything, e.g., the sports car on the grass might not, in reality, be red (though it's a pretty good guess). However, for this paper, our goal is not necessarily to recover the actual ground truth color, but rather to produce *a plausible colorization that could potentially fool a human observer*. Therefore, our task becomes a much more achievable one: to model enough of the statistical dependencies between the semantics and the textures of grayscale images and their color versions in order to produce visually compelling results.

Given the lightness map L of an image, our system predicts its a and b color channels in the CIE *Lab* colorspace. To solve this problem, we leverage large-scale data. Predicting color has the nice property that training data is practically free: any color photo can be used as a training example simply by taking the

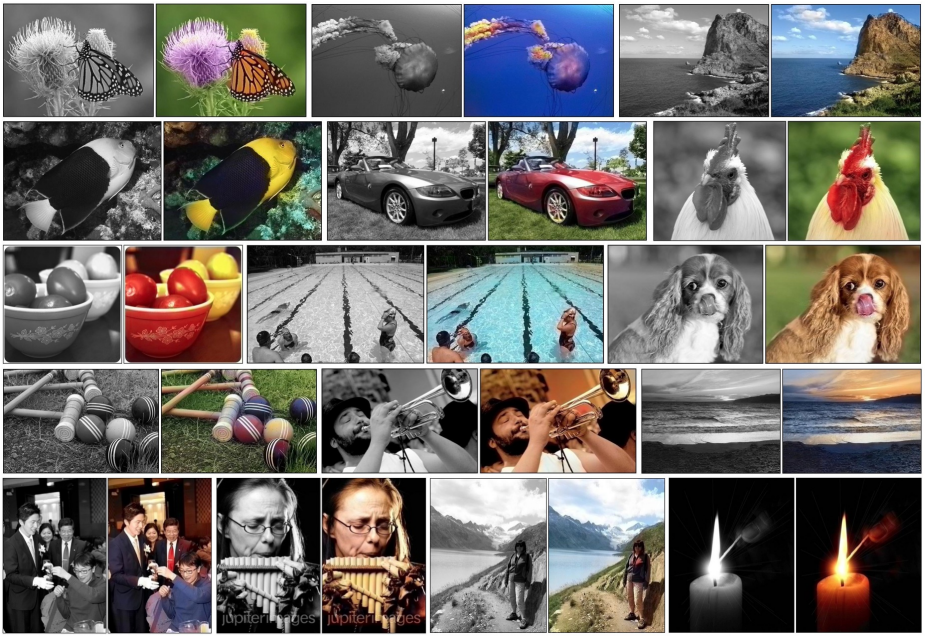

Fig. 1. Example input grayscale photos and output colorizations from our algorithm. These examples are cases where our model works especially well. Please visit `http://richzhang.github.io/colorization/` to see the full range of results. Best viewed in color (obviously).

image's L channel as input and its ab channels as the supervisory signal. Others have noted the easy availability of training data and trained convolutional neural networks (CNNs) to predict color on large datasets [1,2]. However, the results from these and other past attempts tend to look desaturated (see Figures 6 and 12). One explanation is that [1,2] use loss functions that encourage conservative predictions. These losses are inherited from standard regression problems, where the goal is to minimize Euclidean error between an estimate and the ground truth.

We instead utilize a loss tailored to the colorization problem. As pointed out by [3], color prediction is inherently multimodal – many objects, such as a shirt, can plausibly be colored one of several distinct values. To appropriately model the multimodal nature of the problem, we predict a distribution of possible colors for each pixel. Further, we explore reweighting the loss at training time to emphasize rare colors. This encourages our model to exploit the full diversity of the large-scale data on which it is trained. Finally, we produce a final colorization by taking the *annealed-mean* of the distribution. The end result is colorizations that are more vibrant and perceptually realistic than those of previous approaches.

Evaluating the quality of synthesized images is notoriously difficult [4]. Since our ultimate goal is to make results that are compelling to a human observer, we introduce a novel way of evaluating colorization results, directly testing their perceptual realism. We set up a "colorization Turing test", in which we show

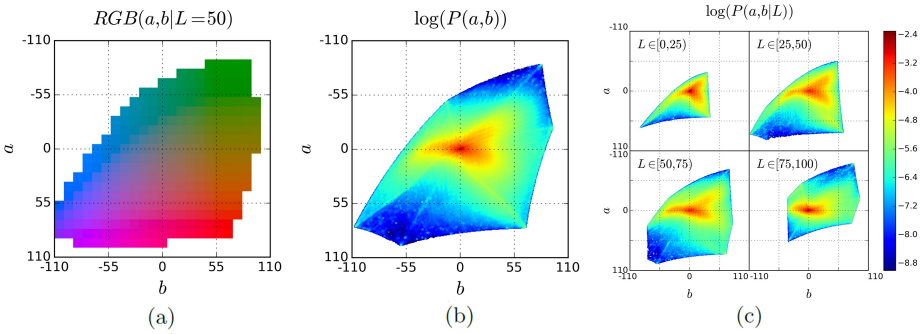

Fig. 2. (a) Quantized *ab* color space with a grid size of 10. A total of 313 *ab* pairs are in gamut. (b) Empirical probability distribution of *ab* values, shown in log scale. (c) Empirical probability distribution of *ab* values, conditioned on L, shown in log scale. Note that the *Lab* gamut has different support for different values of L. The gamuts are shown in the same relative positions as bounding boxes.

participants real and synthesized colors for a photo, and ask them which is which. In this quite difficult paradigm, we are able to fool participants 20% of the time, significantly higher than alternative algorithms. The test demonstrates that in many cases, our algorithm is producing nearly photorealistic results (see Figure 1 for selected successful examples from our algorithm). We also show that our system output can be useful for downstream tasks, in particular grayscale object classification, using an off-the-shelf VGG network [5].

Our contributions in this paper are to: 1) set a new high-water mark on the task of automatic image colorization by training on a million color photos, 2) design an objective function appropriate for the colorization problem that handles the multimodal uncertainty in the problem and captures a wide diversity of colors, and 3) introduce a novel framework for testing colorization algorithms, potentially applicable to other image synthesis tasks.

2 Previous Work

Nonparametric methods Nonparametric methods for image colorization have been well-studied. First, color reference images are provided by the user [3,6,7] or automatically retrieved with user-supplied metadata [8,9]. A colorization is then produced by evaluating local pixel or superpixel similarity between the reference and target images while enforcing spatial coherency. Welsh et al. [6] perform global lightness remapping and match lightness and texture statistics of target grayscale image to the reference color image within a local neighborhood. The method also provides the ability for the user to interact by manually selecting rectangular swatches to match. Gupta et al. [7] use superpixels to help reduce runtime and enforce spatial coherence. Liu et al. [8] and Chia et al. [9] propose methods which utilize reference images gathered from the Internet. Charpiat et al. [3] account for multimodality directly by predicting a distribution over possible colors. The method discretizes the color space based on a provided

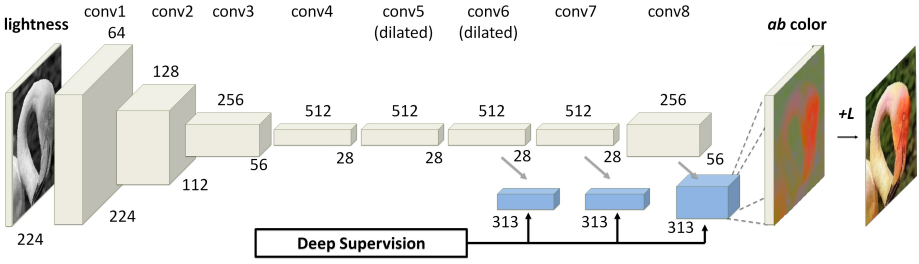

Fig. 3. Our network architecture. Each `conv` layer refers to a block of 2 or 3 repeated `conv` and `ReLU` layers, followed by a `BatchNorm` [23] layer. The net has no `pool` layers. Changes in resolution are achieved through spatial downsampling or upsampling between `conv` blocks.

reference image. We extend the idea of predicting a distribution of colors, but use a large-scale database in a parametric framework.

Parametric methods The algorithms proposed by Deshpande et al. [10], Cheng et al. [1], and Dahl [2] aim to automatically color an image, and are trained on a small, medium, and large-scale data, respectively. Deshpande et al. [10] train a Gaussian random field on SIFT features [11] on a small set of 240 images from 6 scene categories. Cheng et al. [1] learn a 3-layer neural network from the concatenation of patches, DAISY features [12], and the output of an FCN [13] on 47 pre-defined classes. The only attempt at a large-scale fully automatic method is proposed by Dahl [2], which uses VGG features in a Laplacian pyramid framework to produce colorizations. The method is trained on 1.3M images from ImageNet [14].

Convolutional neural networks CNNs have recently been used to great success in computer vision, starting with large-scale image classification by [15]. Improvements to architectures have been made in subsequent years in networks such as VGG [5], GoogLeNet [16], and most recently ResNet [17]. Girshick et al. [18] and others found that initializing with a network trained on a large-scale database, such as ImageNet, provides a strong performance boost on other tasks, such as the PASCAL VOC detection task [19]. Methods for performing pixel-prediction have been explored for the tasks of semantic segmentation [13], segmentation [20], and simultaneous detection and segmentation [21]. Yu et al. [22] found that removing downsample striding and accumulating the stride factors as dilations in subsequent layers improved performance.

3 Approach

In Section 3.1, we outline the formulation of the problem into multinomial classification. In Section 3.2, we explore using a class rebalancing methodology at training time to emphasize rare colors. Then, in Section 3.3, we describe the process of converting a predicted color distribution into a single colorization estimate by defining and using the annealed-mean operation. Finally, in Section 3.4, we characterize our CNN network architecture.

3.1 Objective Function

Given a single lightness channel as input $\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$, our objective is to learn a mapping $\mathcal{F}$ to the two associated color channels $\mathbf{Y} \in \mathbb{R}^{H \times W \times 2}$, where H, W are image dimensions:

$$\widehat{\mathbf{Y}} = \mathcal{F}(\mathbf{X}) \tag{1}$$

We will denote predictions with a $\widehat{\cdot}$ symbol and ground truth without. We perform this task in the CIE *Lab* color space. Because distances in this space aim to represent perceptual distance, a natural objective function that has been used [1,2] is the Euclidean loss between the predicted and ground truth colors. However, the loss is not robust to the inherent ambiguity and multimodal nature of the coloring problem. If an object can take on a set of multiple *ab* values, the optimal solution to the Euclidean loss will be the mean of the set. In color prediction, this averaging effect favors grayish, desaturated results. Additionally, if the set of plausible colorings is not convex, the solution will in fact be out of the set, giving implausible results.

Instead, we treat the problem as a multinomial classification. For a given input $\mathbf{X}$, we learn a mapping $\mathcal{G}$ to predict a probability distribution over possible colorings $\widehat{\mathbf{Z}} \in [0,1]^{HxWxQ}$, where Q is the number of quantized *ab* values:

$$\widehat{\mathbf{Z}} = \mathcal{G}(\mathbf{X}) \tag{2}$$

The mapping $\mathcal{G}$ is learned with a CNN, and the architecture is discussed in Section 3.4. We must define encoder $\mathcal{H}_{gt}^{-1}$, which converts ground truth color $\mathbf{Y}$ to encoded value $\mathbf{Z}$. We quantize the *ab* output space into bins with grid size 10 and keep the $Q = 313$ values which are in-gamut, as shown in Figure 2. Each ground truth value $\mathbf{Y}_{h,w}$ can be encoded as a 1-hot vector $\mathbf{Z}_{h,w}$ by searching for the nearest quantized *ab* bin. However, we found that *soft*-encoding worked well for training, and allowed the network to quickly learn the relationship between elements in the output space [24]. We find the k-nearest neighbors to $\mathbf{Y}_{h,w}$ in the output space and weight them proportionally to their distance from the ground truth using a Gaussian kernel with σ. We find that $k = 10$ and $\sigma = 5$ work well in practice. We then use the multinomial cross entropy loss $\mathrm{L}(\cdot,\cdot)$, with an additional weighting term, defined as:

$$\mathrm{L}(\widehat{\mathbf{Z}}, \mathbf{Z}) = -\frac{1}{HW} \sum_{h,w} v(\mathbf{Z}_{h,w}) \sum_{q} \mathbf{Z}_{h,w,q} \log(\widehat{\mathbf{Z}}_{h,w,q}) \tag{3}$$

Function v can be used to rebalance the loss based on color-class rarity, defined in Section 3.2 below.

Finally, we map probability distribution $\widehat{\mathbf{Z}}$ to color values $\widehat{\mathbf{Y}}$ with decoder $\mathcal{H}$, which will be further discussed in Section 3.3:

$$\widehat{\mathbf{Y}} = \mathcal{H}(\widehat{\mathbf{Z}}) \tag{4}$$

3.2 Class-rebalancing

The distribution of ab values in natural images is strongly biased towards values with low ab values, due to the appearance of backgrounds such as clouds, pavement, dirt, and walls. Figure 2(b) shows the empirical distribution of pixels in ab space, gathered from 1.3M training images in ImageNet [14]. Observe that the number of pixels in natural images at desaturated values are orders of magnitude higher than for saturated values. Without accounting for this strong prior at training time, the loss function is dominated by desaturated ab values. Also, if there is any uncertainty during test time, the predictor may tend to produce desaturated outputs due to the strong prior of natural images. We attempt to account for the class-imbalance problem by reweighting the loss of each pixel at train time based on the pixel color rarity. This is asymptotically equivalent to the typical approach of resampling the training space [25]. Each pixel is weighed by factor $\mathbf{w} \in \mathbb{R}^Q$, based on its closest ab bin.

$$v(\mathbf{Z}_{h,w}) = \mathbf{w}_{q^*}, \text{ where } q^* = \arg\max_q \mathbf{Z}_{h,w,q} \tag{5}$$

$$\mathbf{w} \propto \big((1-\lambda)(\mathbf{G}_\sigma \circ \mathbf{p}) + \lambda\big)^{-1}, \quad \sum_q [\mathbf{G}_\sigma \circ \mathbf{p}]_q \mathbf{w}_q = 1 \tag{6}$$

We first estimate the empirical probability of colors in the quantized ab space $\mathbf{p} \in \Delta^Q$ from the full ImageNet training set and smooth the distribution with a Gaussian kernel $\mathbf{G}_\sigma$. We then mix the distribution with a uniform distribution with weight $\lambda \in [0, 1]$, take the reciprocal, and normalize so the weighting factor is 1 on expectation. We found that values of $\lambda = \frac{1}{2}$ and $\sigma = 5$, as used in soft-encoding, worked well. We also explore the effect of optimizing to the *non*-class-rebalanced objective, where $\lambda = 1$.

$$v(\mathbf{Z}_{h,w}) = 1 \tag{7}$$

We further explore the quantitative and qualitative effect of class-rebalancing in Section 4.2.

3.3 Class Probabilities to Point Estimates

Finally, we define $\mathcal{H}$, which maps the predicted distribution $\widehat{\mathbf{Z}}$ to point estimate $\widehat{\mathbf{Y}}$ in ab space. One choice is to take the mode of the predicted distribution for each pixel, as shown for in the right column of Figure 4 for some qualitative examples. This provides a vibrant but sometimes spatially inconsistent result. For example, pieces of the tent in the third image are red, cyan, and orange, and the fourth image presents a salient blue artifact in the top left corner. On the other hand, taking the mean or expectation of the predicted distribution, shown in the left column, results in a spatially consistent but desaturated prediction. For example, the table and the purple flowers in the first two images take on an unnatural sepia tone. This is unsurprising, as taking the mean after performing classification suffers from some of the problems as optimizing for a Euclidean

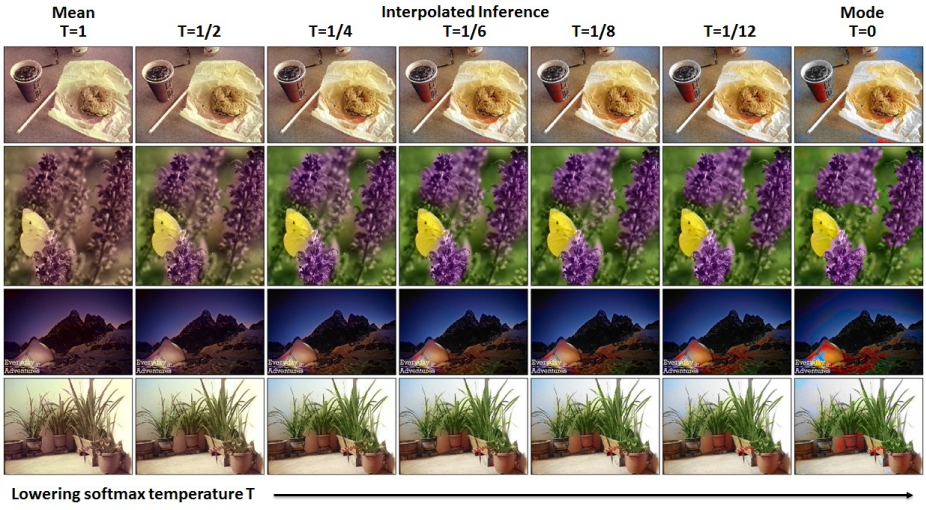

Fig. 4. Sweeping across the temperature setting while inferring a point estimate $\widehat{\mathbf{Y}}$ from predicted distribution $\widehat{\mathbf{Z}}$ using the *annealed-mean* operation. Left to right: $T = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{6}, \frac{1}{8}, \frac{1}{12}, 0$. The left-most images show the distribution mean and the right-most show the mode. We use $T = \frac{1}{6}$ in our system.

loss in a regression framework. In order to explore the space in between the mean and the mode, we *interpolate* by re-adjusting the temperature T of the distribution and taking the mean of the result. We draw inspiration from the simulated annealing technique [**?**], and thus refer to the operation as taking the *annealed-mean* of the distribution:

$$\mathcal{H}(\mathbf{Z}_{h,w}) = \mathbb{E}\big(f_T(\log \mathbf{Z}_{h,w})\big), \quad f_T(\mathbf{z}) = \frac{\exp(\mathbf{z}/T)}{\sum_q \exp(\mathbf{z}_q/T)} \tag{8}$$

Setting $T = 1$ leaves the distribution unchanged, lowering the temperature T produces a more strongly peaked distribution, and setting $T \to 0$ results in a 1-hot encoding at the distribution mode. We found that temperature $T = \frac{1}{6}$, shown in the middle column of Figure 4, captures the vibrancy of the mode while maintaining the spatial coherence of the mean. In the examples shown, the butterfly and flowers remain vibrant and bright, the tent appears to be glowing, and the unwanted artifact above the plant is smoothed into a much less noticeable gradient.

Our final system $\mathcal{F}$ is the composition of CNN $\mathcal{G}$, which produces a predicted distribution over all pixels, and the annealed-mean operation $\mathcal{H}$, which produces a final prediction. The system is not quite end-to-end trainable, but note that the mapping $\mathcal{H}$ operates on each pixel independently, with a single parameter, and can be implemented as part of a feed-forward pass of the CNN.

3.4 Network Design

Our CNN $\mathcal{G}$ architecture is shown in Figure 3 and is trained to predict a probability distribution $\widehat{\mathbf{Z}}$ given a grayscale image $\mathbf{X}$.

Our network has 8 blocks of convolutional layers. The first 5 convolutional blocks are initialized from the VGG convolutional layers [5], with some architectural modifications. We remove the pooling layers, place the stride in the preceding convolutional layer [22], and add batch normalization after every block of convolutions [23]. Since the input is single-channel lightness rather than a three-channel RGB image, the weights of the first convolutional layer are averaged. The VGG network downsamples the spatial feature maps by a factor of 2 after each block of convolutional layers, and doubles the number of output channels. We remove the stride in the final `conv4` layer and dilate the `conv5` layer kernels by a factor of 2 to compensate for the increased input spatial resolution [22]. This allows us to produce 4 times the amount of spatial information in the network bottleneck, with a small 10.3% increase in memory per image.

In addition, we add three additional blocks of convolutions, `conv6`, `conv7`, and `conv8` to mirror layers `conv3`, `conv4`, and `conv5`. Layers `conv6` and `conv7` have dilation 2 and 1, respectively. Layer `conv8` is an upsampling convolution. One could further learn upsampling layers, at the expense of increasing amounts of memory. We found that performing a simple bilinear upsampling from a reduced feature map produced results that were sufficiently detailed. We use deep supervision to help guide learning [26]. To supervise an intermediate layer in the net, we add a readout layer on top, consisting of a 1x1 convolution to convert from feature to output space, followed by a softmax. The output is supervised using cross entropy loss with respect to the down-sampled ground truth. We add these readout layers on `conv6`, `conv7`, and `conv8`. The losses are normalized to the number of spatial locations evaluated, with losses deeper in the network weighted twice the amount as the previous layer.

4 Experiments

Evaluating the quality of synthesized images is well-known to be a difficult task, as simple quantitative metrics, like RMS error on pixel values, often fail to capture visual realism. In Section 4.1, we characterize three evaluation metrics used to evaluate our algorithm in comparison to baselines.

In Section 4.2, we compare the following methods on the ImageNet test set:

1. **Ours** Our method trained with class rebalancing ($\lambda = \frac{1}{2}$ in Equation 6).
2. **Ours fine-tuned without class-rebalancing** To specifically test the contribution of class-rebalancing, we finetune our trained model on a non-rebalanced objective ($\lambda = 1$ in Equation 6) for 16k iterations.
3. **Dahl** [2] Regression model using a Laplacian pyramid on VGG features.
4. **Random** As a naive baseline, we copy the colors from a randomly chosen image from the training set. This baseline has natural image color statistics.
5. **Gray** We use a naive baseline which colors every pixel gray.

Results on ImageNet

Algorithm	AuC CMF (%)	AuC CMF (Rebalanced) (%)	VGG Classification Accuracy (%)	AMT Labeled Real (%)
Ours	88.1	**63.5**	**76.6**	**20.4±1.6**
- f.t. w/o class-rebal	**91.1**	60.1	**76.4**	**18.9±1.5**
Dahl [2]	90.0	54.5	72.1	16.4±1.4
Random	84.9	54.0	62.4	4.2±0.8
Gray	88.8	53.8	73.5	–
Ground Truth	100	100	87.4	50

Table 1. Results on the last 10k images in validation set of ImageNet [14]. Results column 1 contains the area under the curve of thresholded CMF over *ab* space. Column 2 shows the color class-rebalanced variant. Column 3 is the classification accuracy after colorization using the VGG-16 [5] network. Column 4 shows results from our AMT *real vs. fake* test. Higher is better for all metrics. Rows refer to different algorithms. Ours is the output of our full system. The second row is our network, fine-tuned without class rebalancing. Dahl [2] is a previous method. Random baseline is generated by using the *ab* color channel from an incorrect image. Gray baseline predicts $a = b = 0$ for each pixel. Note that on the AMT experiment, an algorithm which produces exact ground truth images would achieve 50% performance in expectation.

Additionally, in Section 4.3, we present a comparison to the method proposed by Deshpande et al. [10] on the LEARCH test set, which was held out during training in our method. We provide qualitative comparisons to Cheng et al. [1] in the supplementary material; quantitative comparisons are not possible, as the authors have not released their code, model, or test set results. Finally, we examine the effect of low-level cues in Section 4.4 and show the multimodality in predict color distributions in Section 4.5.

4.1 Metrics

To address the shortcomings of any individual evaluation metric, we evaluate three that measure different senses of quality:

1. **Raw accuracy (AuC CMF)** As a low-level test, we first assess color prediction quality by computing the AuC (area under curve) under the CMF (conditional mass function), as introduced in [10] for colorization. We compute the percentage of pixels within a threshold of Euclidean distance in *ab* color space, sweep across *ab* distance from 0 to 150, integrate the area under the curve, and normalize. We compute a rebalanced variant of the AuC CMF metric by re-weighting the pixels inversely by color class probability, as described in Equation 6 with $\lambda = 0$. Class-balanced metrics are commonly employed in other tasks such object detection [19] and semantic segmentation [25].
2. **Semantic interpretability (VGG classification):** The AuC CMF metric is a per-pixel measure of colorization accuracy, and is unable to capture

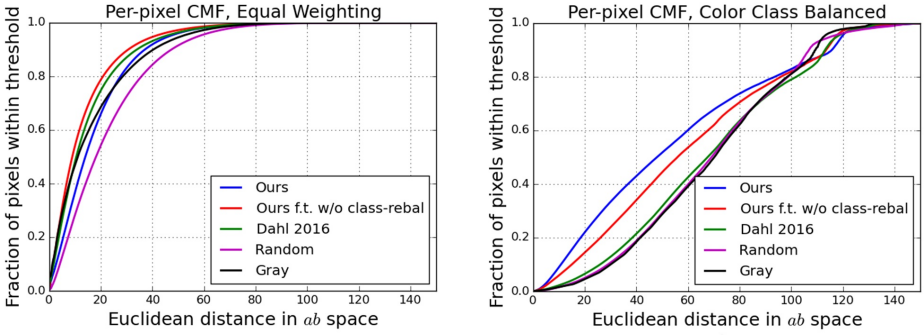

Fig. 5. Comparison of CMF metric across different methods. (a) per-pixel CMF curve. (b) per-pixel CMF curve with color rebalancing.

the visual realism of the synthesized output. We take a first step towards evaluating visual realism with respect to object semantics by using a pre-trained VGG [5] network to evaluate our synthesized color images.

3. **Perceptual realism (AMT):** For many applications, such as those in graphics, the ultimate test of colorization is how compelling the colors look to a human observer. To test the quality of our colorizations in this sense, we ran a *real vs. fake* two-alternative forced choice (2AFC) experiment on Amazon Mechanical Turk (AMT). Participants in the experiment were shown a series of pairs of images. Each pair consisted of a color photo next to a re-colorized version, produced by either our algorithm or a baseline. Participants were asked to click on the photo they believed contained *fake* colors generated by a computer program. Individual images were shown for one second each and after each pair, participants were given unlimited time to respond. All images were resized to 224×224 pixels. Before starting the actual experiment, participants completed 10 practice trials on which they were given feedback. After making their selection, participants were again shown the image pair for 2.4 seconds with the correct answer highlighted. No feedback was given during the actual experiment. Participants were permitted to complete multiple trials of the test.

4.2 Comparisons on ImageNet

We train our network on the 1.3M training images from ImageNet [14], validate on the first 2k images in the validation set, and quantitatively test on the last 10k images in the validation set. Additional qualitative analysis is performed on the last 48k images in the validation set. We also filter out grayscale images, which is further explained in the supplementary material. We present a quantitative analysis on three quantitative metrics in Table 1. A qualitative comparison for selected success and failure cases is shown in Figures 6 and 7. For a comparison on a full selection of random images, please see the supplementary material.

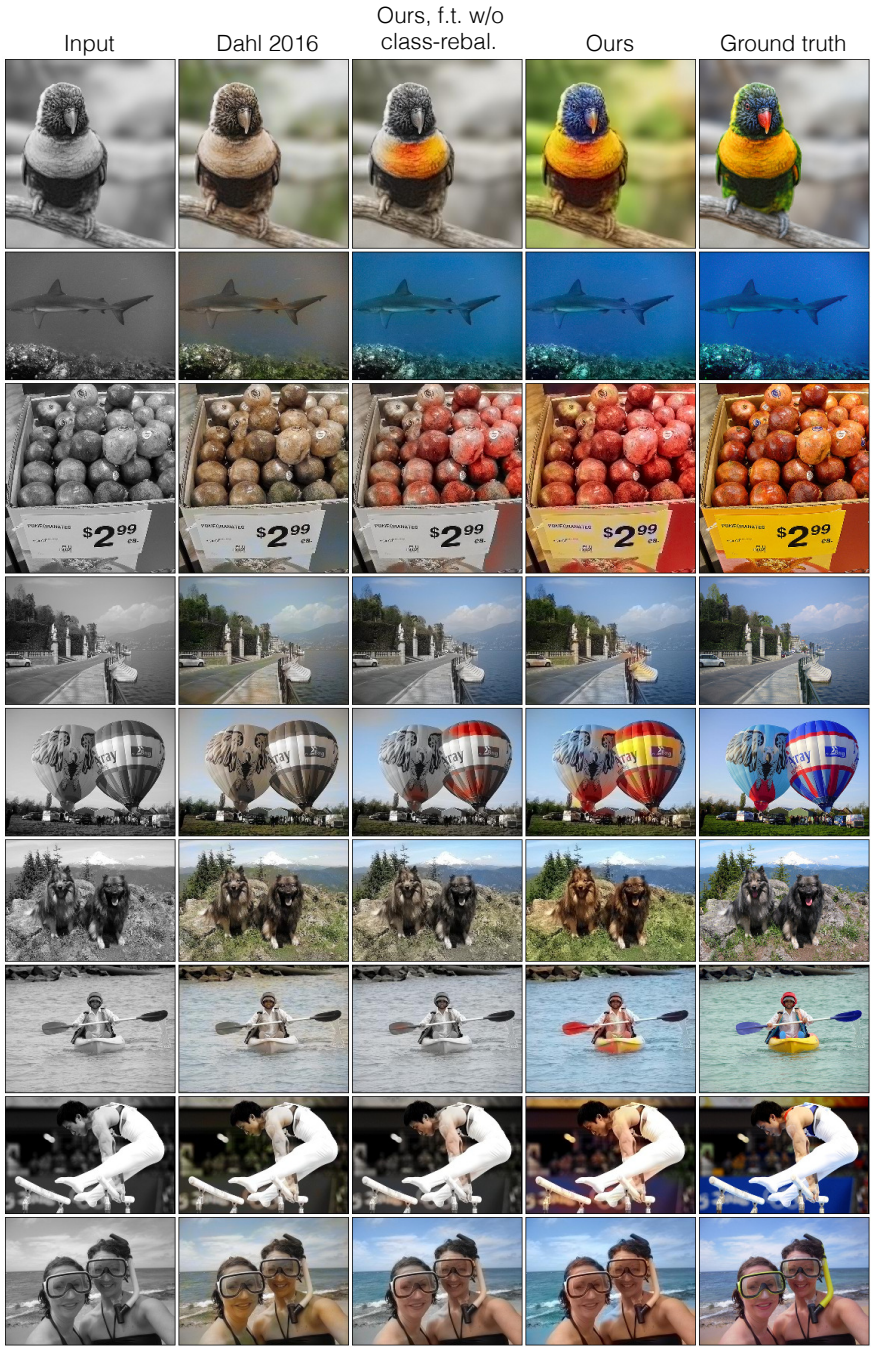

Fig. 6. Example results from our ImageNet test set. Our method produces more accurate and vibrant results than alternative algorithms. Please visit `http://richzhang.github.io/colorization/` to see the full range of results.

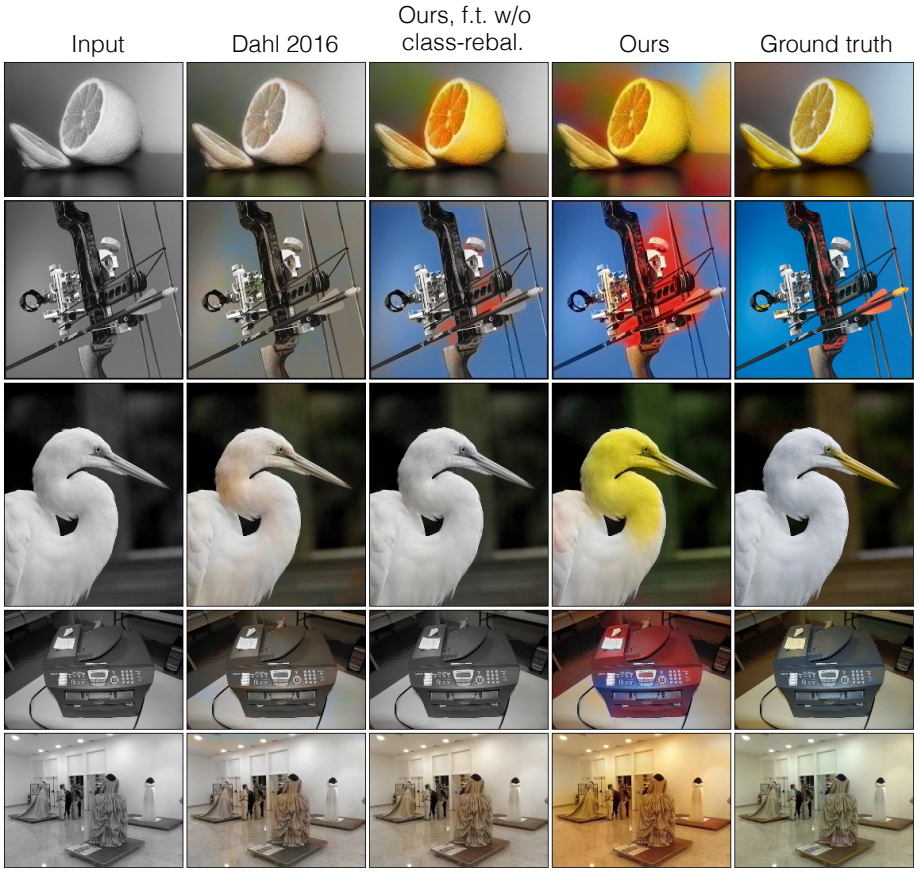

Fig. 7. Common failure modes of our model. Often, the net hallucinates rare and bright colors where there should be none. This is in part because the class-rebalanced objective overestimates the prior probability of rare colors. Finetuning without class-rebalancing reduces these errors, but leaves many images desaturated (see Figure 5). Other common failures include mottled hues in ambiguous regions, such as on walls and artificial objects (rows 1 2, and 4), synthesizing color changes where there is no lightness change (rows 1-4), and producing poorly white balanced results (last row).

Fig. 8. Images colorized by our algorithm from selected categories. Categories are sorted by VGG object classification accuracy of our colorized images relative to accuracy on gracyscale images. Top: example categories where our colorization *helps* the most. Bottom: example categories where our colorization *hurts* the most. Number in parentheses indicates category rank amongst all 1000. Notice that the categories most affected by colorization are those for which color information is highly diagnostic, such as birds and fruits. The bottom examples show several kinds of failures: 1) artificial objects such as modems and clothes have ambiguous colors; color is not very informative for classification, and moreover, our algorithm tends to predict an incoherent distribution of red and blue, 2) for certain categories, like the gray fox, our algorithm systematically predicts the wrong color, confusing the species.

Raw accuracy (AuC CMF): This metric measures the raw per-pixel prediction accuracy of a method. The CMF sweep for different methods is shown in Figure 5(a) and the first results column of Table 1. Because of the strong prior distribution of pixels in the *ab* space towards desaturated colors, predicting gray for every pixel actually results in a very high score of 88.8% for this metric. As expected, predicting random colors drastically reduces performance. For low threshold values, our method, in fact, has a lower score than predicting gray at every pixel. This can occur in cases such as the hot-air balloon in the fifth row of Figure 6, which has ground truth color blue. While our method colors the balloon red, a gray color would actually incur smaller penalty in this metric, even though red is much more realistic than gray in this context. For high threshold values, our method does better than the baseline, but the overall score is slightly lower at 88.1%. Fine-tuning without class rebalancing results in more desaturated predictions and in this regime, our score of 91.1% is higher than the

Dahl [2] baseline of 90.0%. For example, in the third column of Figure 7, the fine-tuned result removes the undesired color from the swan and copy machine on the third and fourth images, respectively.

We also compute a class-balanced variant, as shown in Figure 5(b) and the second column of Table 1. Because colors of interesting objects tend to be more saturated than backgrounds, they are underrepresented in the raw AuC CMF metric used above. It is clear that our method, with a score of 63.5%, is able to better capture diverse colors relative to Dahl [2], Random, and Grayscale baselines, which have scores of 54.5%, 54.0%, and 53.8%, respectively. Without class-rebalancing, the score decreases to 60.1%, quantitatively confirming that training with the color class-rebalancing will more accurately predict rarer colors in the space. For example, consider our results Figure 6. The lorikeet and pomegranates are sepia-toned in the Dahl baseline, shown in the second column. Our method fine-tuned without class rebalancing, shown in the third column, starts to color the belly of the lorikeet and some of the pomegranates, but leaves much of the image desaturated. Finally, our full method, shown in the fourth column, produces bright and realistic colorizations of these images.

Semantic interpretability (VGG classification): As a higher-level test, we evaluate how "interpretable" the synthesized color images are to an off-the-self VGG object classifier [5]. The VGG network was trained to predict from the 1000 ImageNet [14] object classes on natural color images. Because multiple objects may appear in a single image, a top-5 accuracy metric is typically reported for image classification tasks. The ground truth images are classified with an accuracy of 87.4% (this number is slightly lower than the reported VGG validation accuracies, as we test with no ensembling and report on a subset of 10k validation images). After ablating the input images by removing their ab channels, the VGG classification accuracy drops to 73.5%. Naively coloring with the ab channels of random image drastically drops performance to 62.4%. After coloring using our method, the classification performance of the VGG network is improved to 76.6%, a 22.3% relative performance recovery. Our network fine-tuned without class-rebalancing performs at the same level, and both variants outperform the method proposed by Dahl [2] which, at 72.1%, is below the color-ablated performance of the Gray baseline. The results are shown in the third column of Table 1.

In Figure 8, we show a selection of classes that have the most improvement in VGG classification with respect to grayscale, along with the classes for which our colorizations hurt the most. The images shown are automatically selected to be the images that recolorized classification performs best on. This analysis was performed using 48k images from the ImageNet validation set. Interestingly, many of the top classes actually have a color in their name, such as the green snake, orange, and goldfinch. The bottom classes show some common confusions of our system, such as coloring clothing incorrectly and inconsistently and coloring an animal with a plausible but incorrect color. Images in the top and bottom 10 classes are provided in the supplementary material.

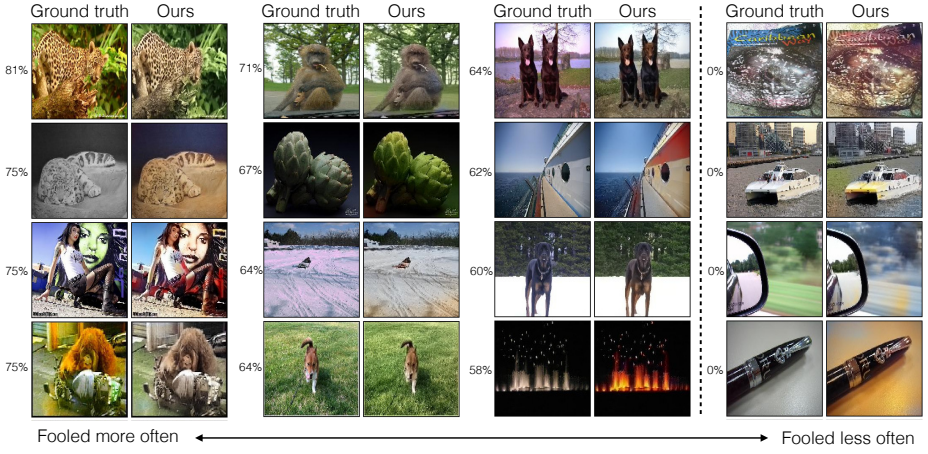

Fig. 9. Images sorted by the percent of times participants in our AMT experiment chose the colorization produced by our algorithm over the ground truth colors. Participants saw pairs like this, with the ground truth and our colorization appearing side by side, one after another. In all pairs to the left of the dotted line, participants selected our colorizations as being more real than ground truth on the majority of trials. In some cases, this may be because the ground truth image was poorly white balanced and our algorithm corrects this by predicting a more prototypical appearance. Right of the dotted line are examples where participants were never fooled and always identified the ground truth as real.

In addition to serving as a perceptual metric, this analysis demonstrates a practical usage for our algorithm: without any additional training or fine-tuning, we can improve performance on grayscale image classification simply by colorizing the images with our algorithm and then passing them to an off-the-shelf classifier trained on color images.

Perceptual realism (AMT): Finally, we run a "colorization Turing test" to test how realistic our synthesized color images are to naive human observers. To check that participants were competent at this task, we compared against the Random baseline, where we colorized each image with the *ab* map from another randomly chosen image. Participants successfully identified these random colorizations as fake 96% of the time, indicating that they understood the task and were paying attention. Figure 9 gives a better sense of the participants' competency at detecting subtle errors made by our algorithm. The right column shows example pairs where participants identified the fake image successfully in 100% of the instances. Each of these pairs was scored at least 10 times. Close inspection reveals that on these images, our colorizations tend to have giveaway artifacts, such as the yellow blotches on the boat and next to the mirror, which ruin otherwise strong results.

Nonetheless, our algorithm fooled participants on 20.4% of trials, as shown in Table 1. After fine-tuning on the non-class-rebalanced objective, the results become less colorful and fool the participants less often at 18.9%. However, this

Results on LEARCH [10] dataset

Algorithm	AuC CMF (%)	AMT Labeled Real (%)
Ours	**90.1**	**17.2±1.9**
Deshpande et al. [10]	88.8	9.8±1.5
Grayscale	89.3	–
Ground Truth	100	50

Fig. 10. Results on LEARCH [10] test set, containing 240 images from 6 categories *beach, outdoor, castle, bedroom, kitchen,* and *living room.* Results column 1 shows the AuC of thresholded CMF over *ab* space. Results column 2 are from our AMT real vs. fake test.

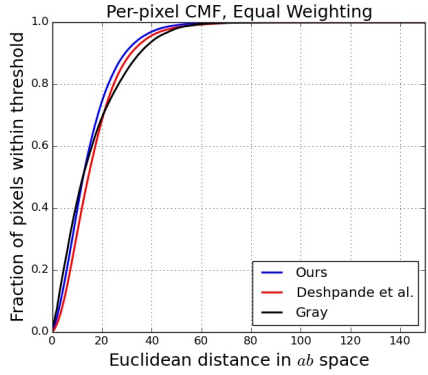

Fig. 11. CMF on the LEARCH [10] test set

difference is not statistically significant ($p = 0.093$), possibly because the less colorful results have fewer giveaway artifacts. Compared to the Dahl [2] baseline performance of 16.4%, our method wins by a statistically significant margin.

Note that if our algorithm exactly reproduced the ground truth colors, the forced choice would be between two identical images an participants would be fooled 50% on expectation. Interestingly, we can identify cases where participants were fooled *more* often than 50% of the time, indicating our results were deemed more realistic than the original. Some examples are shown in the first three columns of Figure 9. In some of these cases, the ground truth image is poorly white balanced or has unusual colors, whereas our system produces a more prototypical appearance.

4.3 Does our model generalize to heldout datasets?

Though our model was trained on object-centric ImageNet dataset, we demonstrate that it nonetheless remains effective for photos from the scene-centric SUN dataset [28] selected by Deshpande et al. [10]. Deshpande et al. recently established a benchmark for colorization using a subset of the SUN dataset and reported top results using an algorithm based on LEARCH [29]. Table 10 provides a quantitative comparison of our method to Deshpande et al.. For fair comparison, we use the same grayscale input as [10], which is $\frac{R+G+B}{3}$. Note that this input space is non-linearly related to the L channel on which we trained. Despite differences in grayscale space and training dataset, our method outperforms Deshpande et al. in both the raw accuracy AuC CMF and perceptual realism AMT metrics. Figure 11 shows qualitative comparisons between our method and Deshpande et al., one from each of the six scene categories. A complete comparison on all 240 images are included in the supplementary material. Our results are able to fool participants in the *real vs. fake* task 17.2% of the time, significantly higher than Deshpande et al. at 9.8%.

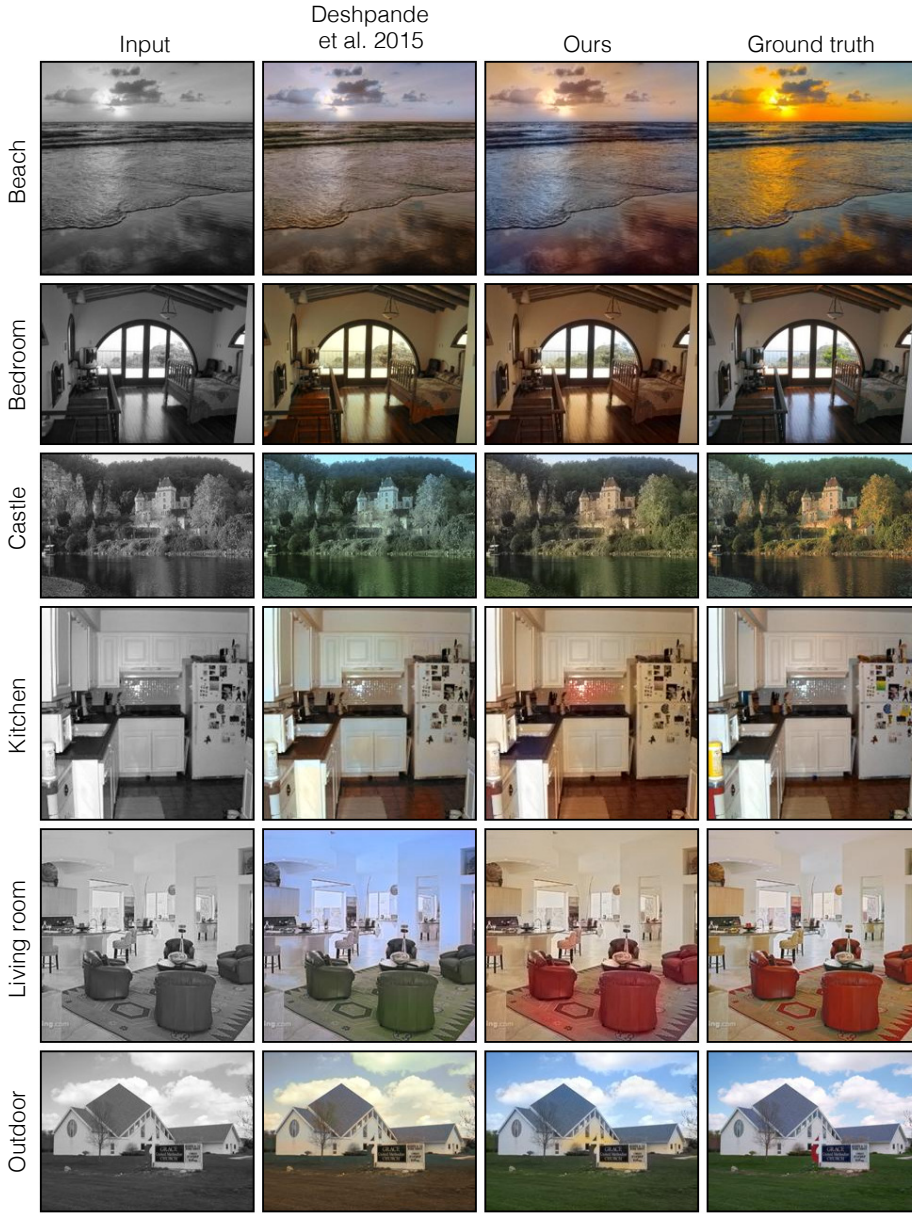

Fig. 12. Our model generalizes well to datasets on which it was not trained. Here we show results on the dataset from [10], which consists of six scene categories from SUN [28]. Compared to the state of the art algorithm on this dataset [10], our method produces more perceptually plausible colorization (see also Figures 10 and 11). Please visit `http://richzhang.github.io/colorization/` to see the results on all 240 images.

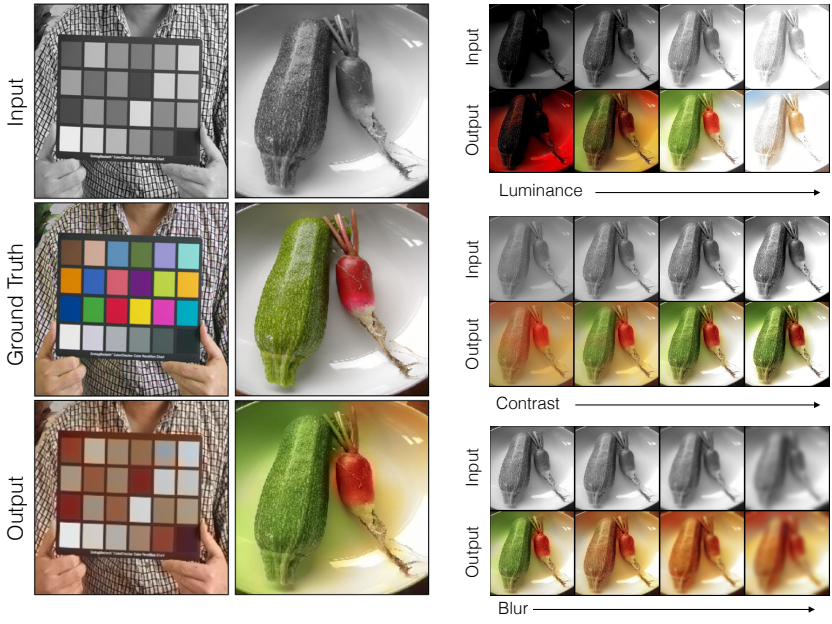

Fig. 13. Left: pixel lightness on its own does not reveal color, as shown by the color chart. In contrast, two vegetables that are nearly isoluminant are recognized as having different colors. Right: stability of the network predictions with respect to low-level image transformations.

In addition, we show qualitative examples of applying our model to legacy black and white photographs. Figures 15, 16, and 17 show examples including work of renowned photographers, such as Ansel Adams and Henri Cartier-Bresson, photographs of politicians and celebrities, and old family photos. One can see that our model is often able to produce good colorizations, even though the low-level image statistics of old legacy photographs are quite different from those of modern-day photos.

4.4 Is the network exploiting low-level cues to predict color?

Unlike many computer vision tasks that can be roughly categorized as low, mid or high-level vision, color prediction requires understanding an image at both the pixel and the semantic-level. Studies of natural image statistics have shown that the lightness value of a single pixel can highly constrain the likely color of that pixel: darker lightness values tend to be correlated with more saturated colors [27].

Could our network be exploiting a simple, low-level relationship like this, in order to predict color?[1] We tested this hypothesis with the simple demonstration

[1] E.g., previous work showed that CNNs can learn to use chromatic aberration cues to predict, given an image patch, its (x,y) location within an image [**?**].

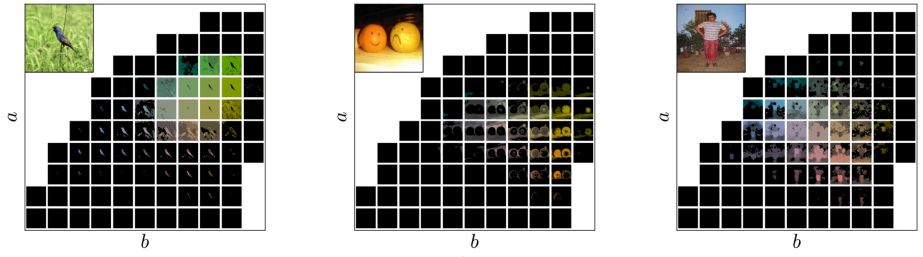

Fig. 14. The output probability distributions per image. The top-left image is final prediction of our system. The black sub-images are quantized blocks of the ab gamut. High probabilities are shown as higher luminance and are quantized for clarity. (a) Background of bird is predicted to be green or brown. Foreground bird has distribution across blue and red colors. (b) Oranges are predicted to be different colors. (c) The person's shirt and sarong has uncertainty across turqoise/cyan/orange and red/pink/purple colors, respectively. Note that despite the multimodality of the per-pixel distributions, the results after taking the annealed-mean are typically spatially consistent.

in Figure 13. Given a grayscale Macbeth color chart as input, our network was unable to recover its colors. This is true, despite the fact that the lightness values vary considerably for the different color patches in this image. On the other hand, given two recognizable vegetables that are roughly isoluminant, the system is able to recover their color.

In Figure 13, we also demonstrate that the prediction is somewhat stable with respect to low-level lightness and contrast changes. Blurring, on the other hand, has a bigger effect on the predictions in this example, possibly because the operation removes the diagnostic texture pattern of the zucchini.

4.5 Does our model learn multimodal color distributions?

As discussed above, formulating color prediction as a multinomial classification problem allows the system to predict multimodal distributions and can capture the inherent ambiguity in the color of natural objects. In Figure 14, we illustrate the probability outputs $\widehat{\mathbf{Z}}$ and demonstrate that the network does indeed learn multimodal distributions. The system output $\widehat{\mathbf{Y}}$ is shown in the top-left of Figure 14. Each block illustrates the probability map $\widehat{\mathbf{Z}}_q \in [0,1]^{H,W}$ given ab bin q in the output space. For clarity, we show a subsampling of the Q total output bins and coarsely quantize the probability values. In Figure 14(a), the system clearly predicts a different distribution for the background vegetation and the foreground bird. The background is predicted to be green, yellow, or brown, while the foreground bird is predicted to be red or blue. Figure 14(b) shows that oranges can be predicted to be different colors. Lastly, in Figure 14(c), the man's sarong is predicted to be either red, pink, or purple, while his shirt is classified as turquoise, cyan or light orange. Note that despite the multi-modality of the

Fig. 15. Applying our method to black and white photographs by Ansel Adams.

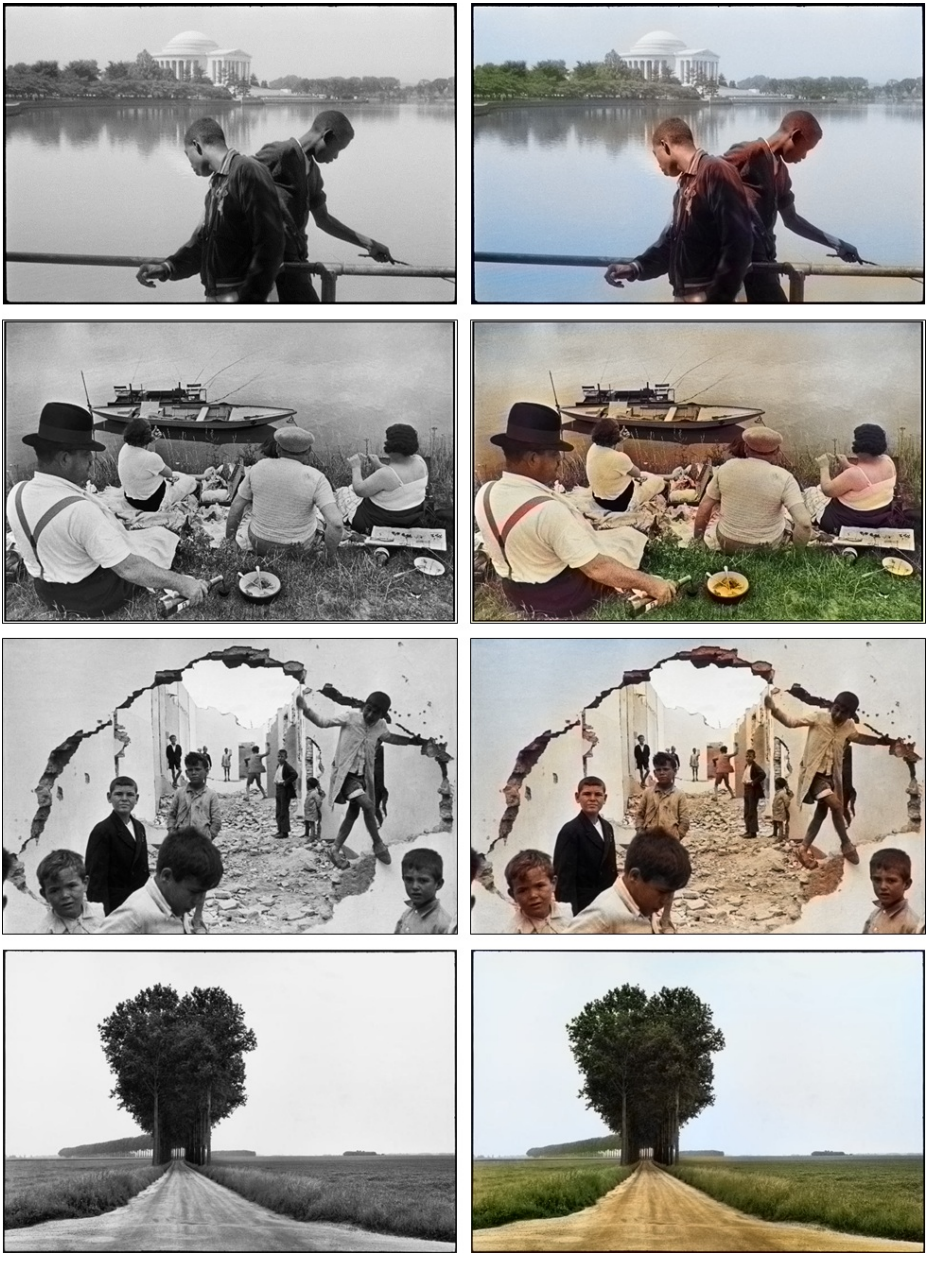

Fig. 16. Applying our method to black and white photographs by Henri Cartier-Bresson.

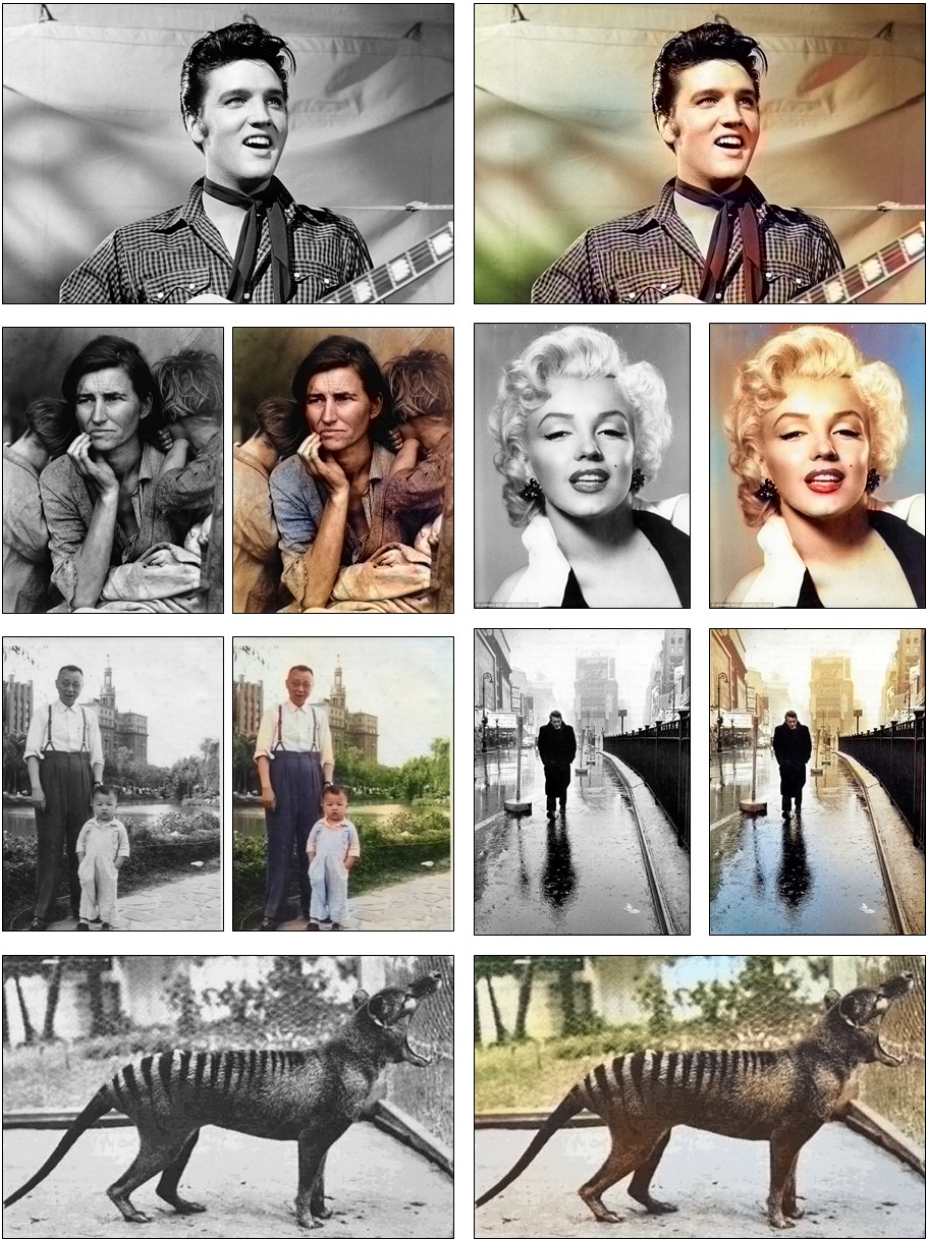

Fig. 17. Applying our method to legacy black and white photographs. Top to bottom, left to right: photo of Elvis Presley, photo of *Migrant Mother* by Dorothea Lange, photo of Marilyn Monroe, an amateur family photo, photo by Henri Cartier-Bresson, photo by Dr. David Fleay of *Benjamin*, the last captive thylacine which went extinct in 1936.

prediction, taking the annealed-mean of the distribution produces a spatially consistent prediction.

5 Conclusion

Colorizing grayscale images has been a long sought-after goal in computer graphics and is closely related to difficult pixel prediction problems in computer vision. Previous automatic approaches have failed to capture the full range and vibrancy of colors in the natural world. We have presented a method that addresses this issue and is able to produce colorful and diverse results. To do so, we utilize a classification framework that models the inherent multimodality in color prediction, introduce a class-rebalancing scheme to promote the learning of rare colors, and train a bigger model on more data than past methods. We test our algorithm by evaluating a low-level per-pixel accuracy metric, along with high-level semantic interpretability and perceptual realism metrics. The results demonstrate colorization with a deep CNN and a well-chosen objective function can come close to producing results indistinguishable from real color photos.

Acknowledgements

This research was supported, in part, by ONR MURI N000141010934, NSF SMA-1514512, and a hardware donation by NVIDIA Corp. We thank members of the Berkeley Vision Lab for helpful discussions. We thank Aditya Deshpande for providing help with comparisons to [10].

References

1. Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 415–423
2. Dahl, R.: Automatic colorization. In: http://tinyclouds.org/colorize/. (2016)
3. Charpiat, G., Hofmann, M., Schölkopf, B.: Automatic image colorization via multimodal predictions. In: Computer Vision–ECCV 2008. Springer (2008) 126–139
4. Ramanarayanan, G., Ferwerda, J., Walter, B., Bala, K.: Visual equivalence: towards a new standard for image fidelity. ACM Transactions on Graphics (TOG) **26**(3) (2007) 76
5. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
6. Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to greyscale images. ACM Transactions on Graphics (TOG) **21**(3) (2002) 277–280
7. Gupta, R.K., Chia, A.Y.S., Rajan, D., Ng, E.S., Zhiyong, H.: Image colorization using similar images. In: Proceedings of the 20th ACM international conference on Multimedia, ACM (2012) 369–378
8. Liu, X., Wan, L., Qu, Y., Wong, T.T., Lin, S., Leung, C.S., Heng, P.A.: Intrinsic colorization. In: ACM Transactions on Graphics (TOG). Volume 27., ACM (2008) 152

9. Chia, A.Y.S., Zhuo, S., Gupta, R.K., Tai, Y.W., Cho, S.Y., Tan, P., Lin, S.: Semantic colorization with internet images. In: ACM Transactions on Graphics (TOG). Volume 30., ACM (2011) 156
10. 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
11. Lowe, D.G.: Object recognition from local scale-invariant features. In: Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Volume 2., Ieee (1999) 1150–1157
12. Tola, E., Lepetit, V., Fua, P.: Daisy: An efficient dense descriptor applied to wide-baseline stereo. Pattern Analysis and Machine Intelligence, IEEE Transactions on **32**(5) (2010) 815–830
13. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3431–3440
14. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3) (2015) 211–252
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
16. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1–9
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
18. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2014) 580–587
19. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**(2) (2010) 303–338
20. Xie, S., Tu, Z.: Holistically-nested edge detection. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1395–1403
21. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: Computer vision–ECCV 2014. Springer (2014) 297–312
22. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)
23. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
24. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
25. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. Pattern Analysis and Machine Intelligence, IEEE Transactions on **35**(8) (2013) 1915–1929
26. Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets. arXiv preprint arXiv:1409.5185 (2014)
27. Chakrabarti, A.: Color constancy by learning to predict chromaticity from luminance. In: Advances in Neural Information Processing Systems. (2015) 163–171

28. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: Computer vision and pattern recognition (CVPR), 2010 IEEE conference on, IEEE (2010) 3485–3492
29. Ratliff, N.D., Silver, D., Bagnell, J.A.: Learning to search: Functional gradient techniques for imitation learning. Autonomous Robots **27**(1) (2009) 25–53