Combining Deep Convolutional Neural Networks with Markov Random Fields for Image Colorization

Luke Melas-Kyriazi George Han

Harvard University

Abstract

We construct and train a deep convolutional neural network for automatic image colorization, the task of generating a colored image from a black-and-white image input. We build upon recent CNN-based colorization work by Iizuka et al [4], Zhang et al [16], and Larsson et al [7] in constructing our network, combining different elements of each paper's approaches to colorization. We experiment with various architectures, loss functions, and training datasets with the aim of producing natural, vibrant colorizations on a diverse range of grayscale images. Our final model is a two-step approach combining a deep convolutional neural network with a Markov Random Field: the neural network predicts a distribution over colors for each pixel in a grayscale input image, and the Markov Random Field infers a final colorized image from these distributions. Finally, we apply our model to colorize a set of historical images and a scene from a black-and-white film.

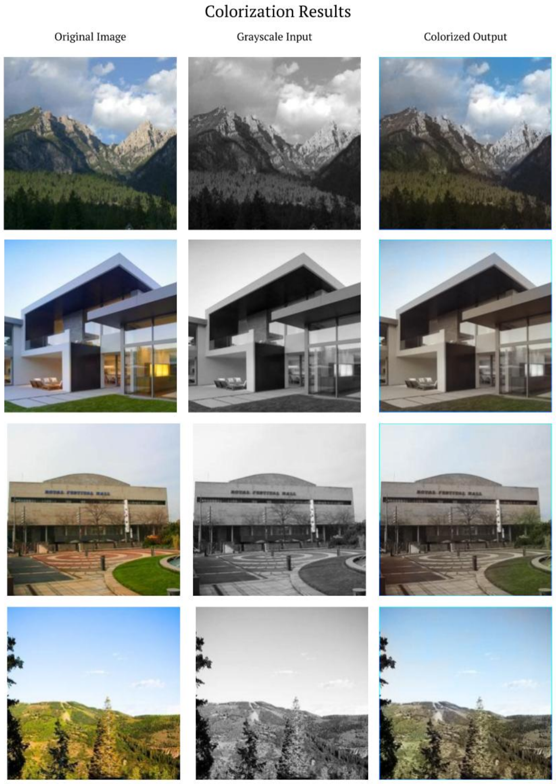

Figure 1. Outputs of our final colorization model.

1. Overview

1.1. Problem Outline

In image colorization, we aim to automatically generate a colored image from a grayscale image input. Precisely, given the lightness channel of an image (a $1 \times H \times W$ input), our objective is to infer the corresponding chrominance and hue channels (a $2 \times H \times W$ output). We aim to infer plausible, natural-looking images with consistent colorizations throughout the image frame. This task has a number of real-world applications, foremost of which is colorizing historical images and films.

Colorization poses a significant challenge to traditional computer vision techniques because a single black-and-white image may have a multitude of plausible colorizations. For example, since grayscale images of red, blue, and green dresses may be identical shades of gray, there may be multiple plausible colorizations of a dress in an input grayscale image. This ambiguity makes it difficult to formulate an objective function assessing the quality of a given colorization. As a consequence, traditional colorization methods struggled to produce natural colorizations [13] [3], instead relying on significant user input in the form of color scribbles [9] on the grayscale input or colored image examples [14].

Recent research on colorization has sought to produce vibrant, natural colorizations with deep convolutional neural networks (CNNs). Colorization is well-suited to deep learning-based models due to the abundance of training data: any colored image may be broken down into its lightness channel (the model input) and its color channels (the

	Iizuka et al.	**Larsson et al.**	**Zhang et al.**
Model Architecture	Two-stream convolutional network	Hypercolumn approach	Single-stream convolutional network
Training Dataset	Places365	ImageNet	ImageNet
Loss Function	Regression (mean squared error)	Classification (cross-entropy)	Classification (cross-entropy)

Table 1. Comparison of previous deep neural network-based approaches to image colorization.

ground truth). CNNs have found success in colorization in large part due to their ability to extract natural features from images and scale to large training datasets. Three papers published in 2016 (Iizuka et al. [4], Zhang et al [16], and Larsson et al [7]) introduce different CNN-based approaches to colorization, all of which demonstrate results surpassing traditional colorization methods (from a qualitative standpoint).

1.2. Project Outline

In our project, we build on the approaches employed in the three papers introduced above to construct a deep convolutional neural network for colorization. We experiment with multiple network architectures, settling on a fully-convolutional approach based on the ResNet image classification network. We train this model on a variety of datasets, including the Places365 dataset and CelebA dataset, using a mean squared error loss between our model output and the ground-truth color channels.

We then extend upon this work by constucting a second model combining a deep neural network with a Markov Random Field. Rather than output a colored image, the network outputs a distribution over a set of binned colors (25 bins for each color channel) for each pixel in the input image. The Markov Random Field (a Potts model) converts these distributions to a final colored image, with pairwise potentials enforcing that nearby pixels with similar lightness values have similar colorizations. As an additional benefit, our Markov Random Field can incorporate a global histogram (a set of user-defined colors) to bias the colorization toward a particular set of colors.

The structure of this paper is as follows: in the background section, we describe how previous colorization models differ with respect to neural network architecture, choice of loss function, and choice of training data. In the model section, we present the two models described above and analyze our architecture design choices. In the experiments section, we detail our training procedure, present results on the Places365 and CelebA datasets, and describe how global histograms may be incorporated into our Markov Random Field model. To conclude, we colorize a number of historical images and discuss areas of future research.

2. Background

Four recent papers in '16 -'17 highlight the recent advances in colorization:

1. Iizuka et al. - Let there be Color! Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization [4]
2. Larsson et al. - Learning Representations for Automatic Colorization [7]
3. Zhang et al, 2016 - Colorful Image Colorization [16]
4. Zhang et al, 2017 - Real-Time User-Guided Image Colorization with Learned Deep Priors [17]

While all four papers employ deep convolutional neural networks, each does so with a unique architecture.

Iizuka et al. construct an image classification network to extract global and mid-level image features, combine these features in a fusion layer, and pass the combination through a colorization (deconvolution) network to generate color channel outputs. Iizuka et al. train this model with a hybrid loss function: the loss is the sum of a classification loss (from the image classifier) and a colorization loss (mean squared error between the output colors and the ground truth color channels). Larsson et al. take the idea of combining global and mid-level features a step further by employing hypercolumns, vectors of concatenated features extracted from all levels of a classification network. For the classification network, Larsson et al. employ VGG-Gray, a variant of the VGG classifier trained on grayscale images. Zhang et al, 2016, opt for a fully-convolutional, single-stream architecture initialized with VGG-Gray. Their follow-up network in Zhang et al, 2017, also incorporates optional user-input by enabling users to specify the colors of individual pixels.

Whereas Iizuka et al. predict a colored image directly from their convolutional neural network, Larsson et al. and Zhang et al. output a distribution over a discrete set of color bins for each pixel in the input grayscale image. After outputting distributions, they train their networks with cross-entropy loss between their output and the ground truth (taken to be either a one-hot vector or a sharply-peaked normal distribution around the bin corresponding to each pixel's true color).

Larsson et al. and Zhang et al. then infer a final colored image from their per-pixel color distributions. After experimenting with sampling, taking means, and taking modes, Larsson et al. conclude that taking the mean of each pixel's color distribution yields the best qualitative results. Similarly, Zhang et al. take an annealed mean of each pixel's distribution; they parameterize the annealed mean by a temperature parameter T, where $T = 1$ yields the mean and $T = 0$ yields the mode, and find that $T = 0.38$ yields the best qualitative results.

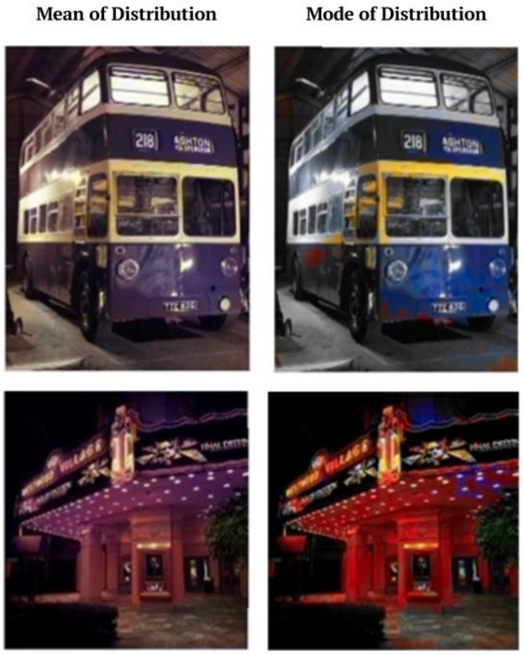

Table 2. Taking the mode of the distributions yields more vibrant colors, but leads to color inconsistencies across the image. *Images from Zhang et al.*

However, these averaging-based methods suffer from the same potential downsides as using a mean squared error loss function: multi-modal color distributions are averaged, yielding colorizations which are consistent across the image, but not as vibrant as may be desired. In addition, inference is performed on a per-pixel level without taking into account the image structure or presence of adjacent pixels. In our second model, we address this issue by proposing a Markov Random Field (Potts model) for inferring a final colored image from per-pixel distributions over color bins. In this model, pairwise potentials encourage nearby pixels with similar lightness values to be colorized similarly.

3. Model and Results

3.1. CNN Model

For our first model, we construct a deep convolutional neural network that outputs a colored image directly from a grayscale image input. We work with images in the CIE-L*A*B* colorspace, which represents a pixel as a tuple (L, a, b), where L is lightness, a represents the spectrum between magenta and green, and b represents the spectrum between yellow and blue. CIE-L*A*B* is the preferred colorspace in image colorization because it enables us to easily separate the lightness channel from the color channels and it has no discontinuities in its representation of the color channels (as opposed to a colorspace such as HSL, where the hue ranges from 0° to 360°).

We test a variety of neural network architectures for colorization, including a fully convolutional (single-stream) network, inspired by Zhang et al., and a network integrating global and mid-level features, inspired by Iizuka et al. Rather than train from scratch, as done by Iizuka et al., we begin by pre-training a ResNet-18 classifier with grayscale inputs, creating "ResNet-Gray". Tested on a subset of the ImageNet dataset [2], ResNet-Gray achieves 79.7% Top-5 classification accuracy, as opposed to 85.0% accuracy for (color) ResNet-18. This retraining not only enables us to train our colorizer much more quickly than otherwise possible, but also speaks to the quantity of information contained solely in the lightness channel of an image.

Classification Performance on the Places365 Dataset

Model	Top-1 Accuracy	Top-5 Accuracy
VGG	55.2%	84.9%
ResNet-18	54.7%	85.0%
ResNet-18-Gray	51.4%	79.7%

Our final colorization network architecture is shown in Figure 2. We initialize the beginning layers of the network with ResNet-18-Gray, but we do not fix these weights during training, enabling their fine-tuning for colorization. We train with an NVidia K80 GPU on the Azure cloud computing platform.

Given our use of ResNet-18-Gray, we found that integrating global features into our network, as proposed by Iizuka et al., did not improve our colorization performance (qualitatively or as measured by mean squared error). These global features and the corresponding classification loss used in Iizuka et al. may have played an important role

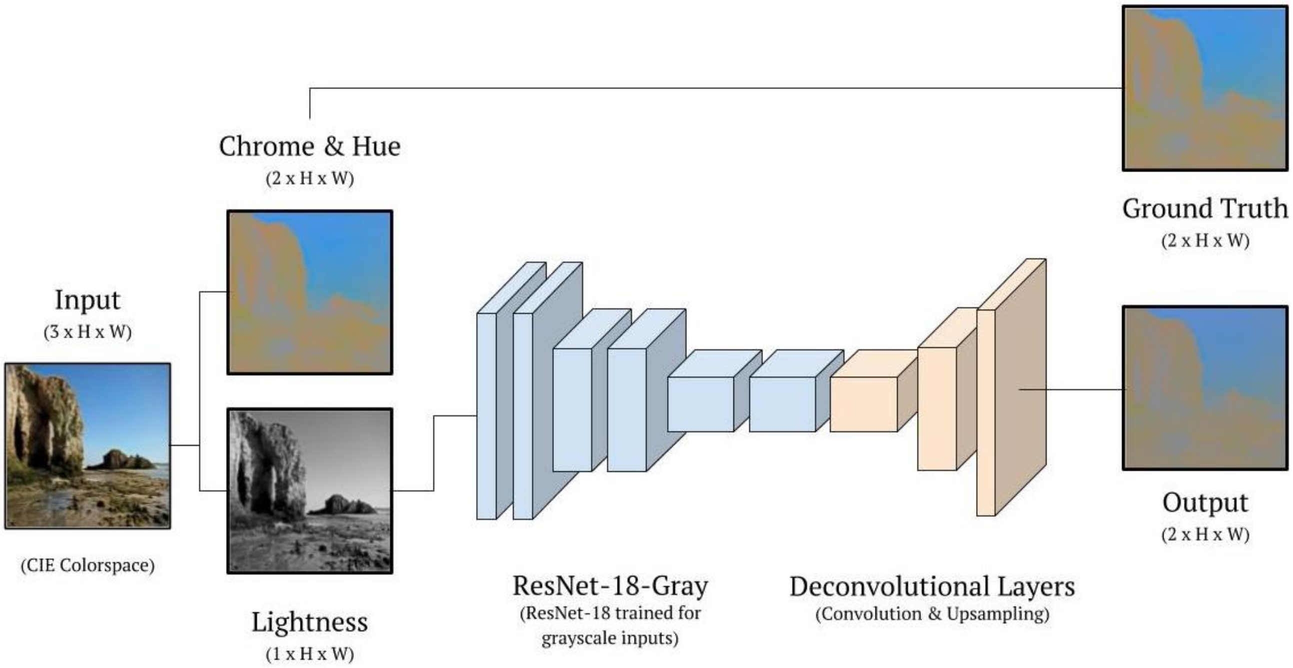

Figure 2. A diagram of our final single-stream network architecture.

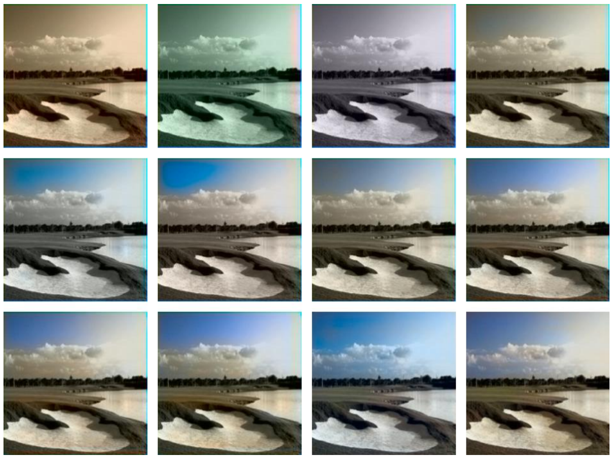

Figure 3. Output colorizations produced by our model over training epochs (shown above: epochs 1 to 12).

in training their network from a random (Xavier) initialization, but they made little impact on our network's colorizations. As a result, we opted for a simpler and faster-to-train single-stream network architecture. We trained with both mean squared error (L2) and mean absolute error (L1) loss functions, finding that mean squared error produced more vibrant (qualitative) colorizations. In summary, our first model is a fully-convolutional network based on the ResNet-18-Gray classifier, which outputs a colored image from an input grayscale image and is trained using a mean squared error loss function.

It is difficult to determine a quantitative metric for success on the task of image colorization, given the inherent ambiguity of colorizations noted in the introduction. Nonetheless, we may compare the performance of different variants of our model using per-pixel mean squared error (averaged across the a and b color channels). We present the average per-pixel errors of our different models in Table 3.

We show the progression of our model over its initial training epochs in Figure 3. The network begins by predicting the approximately same color for all pixels in the image, learning over time to assign blue colors to the sky, green colors to the golf course, and yellow colors to the sand.

3.2. CNN & MRF Model

For our second model, we present a two-step approach to colorization combining a deep convolutional neural network and a Markov Random Field.

As shown by our first model above, training a network with mean squared error can yield strong colorization results. However, as discussed in the Problem Overview, mean squared error has the undesirable property that it harshly penalizes plausible colors that differ from the ground truth. We address this property in our second model by outputing a distribution over a set of discrete color bins for each pixel in the input grayscale image, as done by Larsson et al. and Zhang et al.. We use 25 bins per color channel (the a and b channels), predicting a distribution over each channel separately. We train with cross-entropy error against the ground

Average Per-Pixel Mean Squared Error

	CNN trained on Places365 for 10 epochs	**CNN trained on Places365 for 40 epochs**	**CNN trained on CelebA for 10 epochs after Places365 for 40 epochs**
Per-pixel MSE on the Places365 validation set	0.0025	0.0019	0.0021
Per-pixel MSE on the CelebA validation set	0.0021	0.0021	0.0008

Table 3. Comparison of per-pixel mean squared error, averaged across the a and b channels (with colors from 0 to 1), on the models we trained. Note that we trained on subsets of the Places365 and CelebA datasets (so multiple epochs were possible). Also note that a network needs lower per-pixel errors on the Places365 dataset than the CelebA dataset to achieve qualitatively good results, as there are a much broader range of colors in the Places365 dataset (the majority of pixels in the CelebA dataset are skin tones).

truth, represented as a one-hot vector corresponding to the bin with the true pixel color.

Our neural network model architecture is the same as that described in the section above, except that the final convolutional layer outputs a distribution of colors (a tensor of size $50 \times H/4 \times W/4$) rather than a colored image (a tensor of size $2 \times H/4 \times W/4$). Note that we have 50 channels in the output as there are 25 bins for each color channel (a and b).

Although we began training this model, we found it more difficult and computationally intensive to train than the model based on mean squared error. As a result, for the purposes of the second part of this model, our Markov Random Field, we generated distributions over color values using the network trained by Zhang et al., using pretrained weights provided by the authors (who trained for multiple weeks). We believe that our network could generate similar color distributions with more training time, but for this project we used the outputs of their network as inputs to our Markov Random Field.

Given a distribution over color bins for each pixel in the grayscale input image, we infer a final colorization with a Markov Random Field (Potts model). We apply the same Potts model separately to the a and b color channels, combining the results of the Potts models and the original grayscale input to produce our final image.

We represent each pixel y_i as a categorical random variable (with 25 bins). The unitary potential θ_i of pixel y_i is the distribution over colors produced by our deep neural network times ten:

$$\theta_i(y_i = k) = 10 * \delta(x_i = k)$$

where x_i is given by the CNN

For each pair (i, j) of adjacent pixels, the binary potential enforces the smoothness of the coloring. Binary potentials between neighboring nodes are weighted by the difference in their lightness values L_i, to allow pixels representing different objects in the image (such as those lying on the edges of objects) to be colored differently. The binary potential is then:

$$\theta_{(i,j)} = w_{(i,j)} * s_{(i,j)} \quad \text{where}$$

$$s_{(i,j)} = \begin{cases} 10 & y_i = y_j \\ 3 & |y_i - y_j| \leq 2 \\ 0 & \text{otherwise} \end{cases} \quad \text{and}$$

$$w_{(i,j)} = e^{-\frac{1}{\sigma}(L_i - L_j)^2}$$

The idea of enforcing similar colorings only for pixels with similar lightness values was adopted from Levin et al. [9], which proposed an optimization-based colorization method relying on user input. In the expression above, σ is a hyperparameter controlling the scale of the lightness weighting, which we set to $1 \cdot 10^{-4}$. We perform inference on our Potts model with Coordinate Ascent Variational Inference [1].

We also experimented with other unary potential, binary potentials, and hyperparameter settings. We found that for the unary potentials, taking the square root of the distributions produced by the CNN yielded slightly smoother results than taking the distribution directly, although doing so led to slower convergence times.

The output of the MRF model are distributions for each pixel placing nearly all the mass of the distribution in a single color bin. From this output, we generate our final image by taking the mode (or mean, as they are essentially equivalent at this point) of each distribution. An example of the outputs of our model over eight epochs are shown in Figure 4. The image begins in grayscale, as the mean field model is initialized with a uniform distribution, and quickly progresses toward a colorized image.

After reaching convergence (usually approximately eight epochs), we found that the final images produced are (qualitatively) more similar to those produced by taking the modes of the original distributions than the means of the distributions. That is, they are more vibrant than those obtained by taking means, but also contain some colorization inconsistencies ("splotches" of color). We believe that although the binary potentials enforce some local smoothing, they also lead some regions in the image to be colored with exactly the same color values; this behavior results in small areas of uniform color followed by relatively harsh transitions to different colors in other regions of the image, creating color "splotches." This effect may be beneficial in an application such as image denoising, but leads to some unnatural artifacts in colorization [10] [6].

Nonetheless, the images produced by the Potts model appear slightly more consistent than those obtained by taking modes, while retaining the same vibrancy. In Figure 5, for example, the Markov Random Field model produces a vibrant colorization of the dog, and partially removes an unnatural bright red spot on the dog's tongue, which is present in the images produced by both taking means and taking modes.

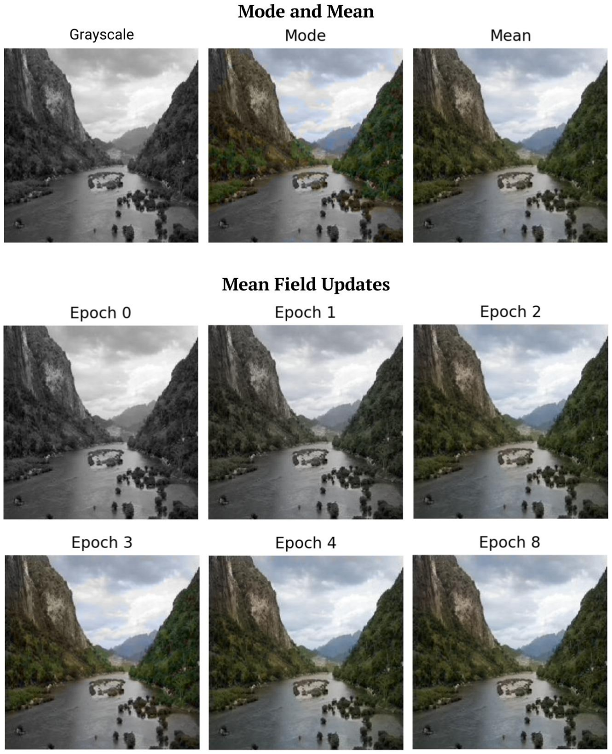

Figure 4. Output colorizations produced by our Potts model, running from grayscale to convergence.

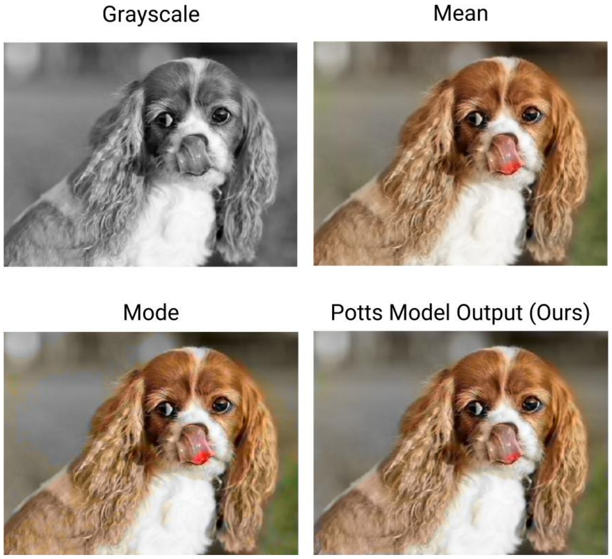

Figure 5. Output of our Potts Model compared to taking means or modes. Note that the model partially reduces the unnatural red mark on the dog's tongue.

4. Experiments

4.1. Datasets: Places365 & CelebA

We train our first model, initialized with our ResNet-18-Gray classifier, separately on three datasets: Places365 (indoor and outdoor scenes) [18], ImageNet (objects) [2], and CelebA (celebrity faces) [11]. We use the Adam optimizer with an initial learning rate of 0.1.

For most images, training on Places365 produces the most accurate colorizations, but for close-up portraits and skin tones, CelebA produces the best results. We compare the results of running both models on the validation images of both datasets, using per-pixel mean squared error. As seen in Figure 6, the model trained on CelebA accurately colorizes skin tones but struggles to colorize landscapes, whereas the model trained on Places365 accurately colors landscapes but struggles to colorize skin tones. We also try training our network on the combination of both datasets; we obtain colorizations that lie (qualitatively) between those of the separately-trained models on both validation sets.

4.2. Global Histogram Transfer

In addition to producing automatic colorizations, our second model can incorporate a global color histogram, a set of user-defined colors, to bias the final colorization. To incorporate this histogram, we modify our Potts Model as follows: we take the unary potential for pixel i to be the elementwise product of the color bin distribution produced by our neural network for pixel i and the global histogram,

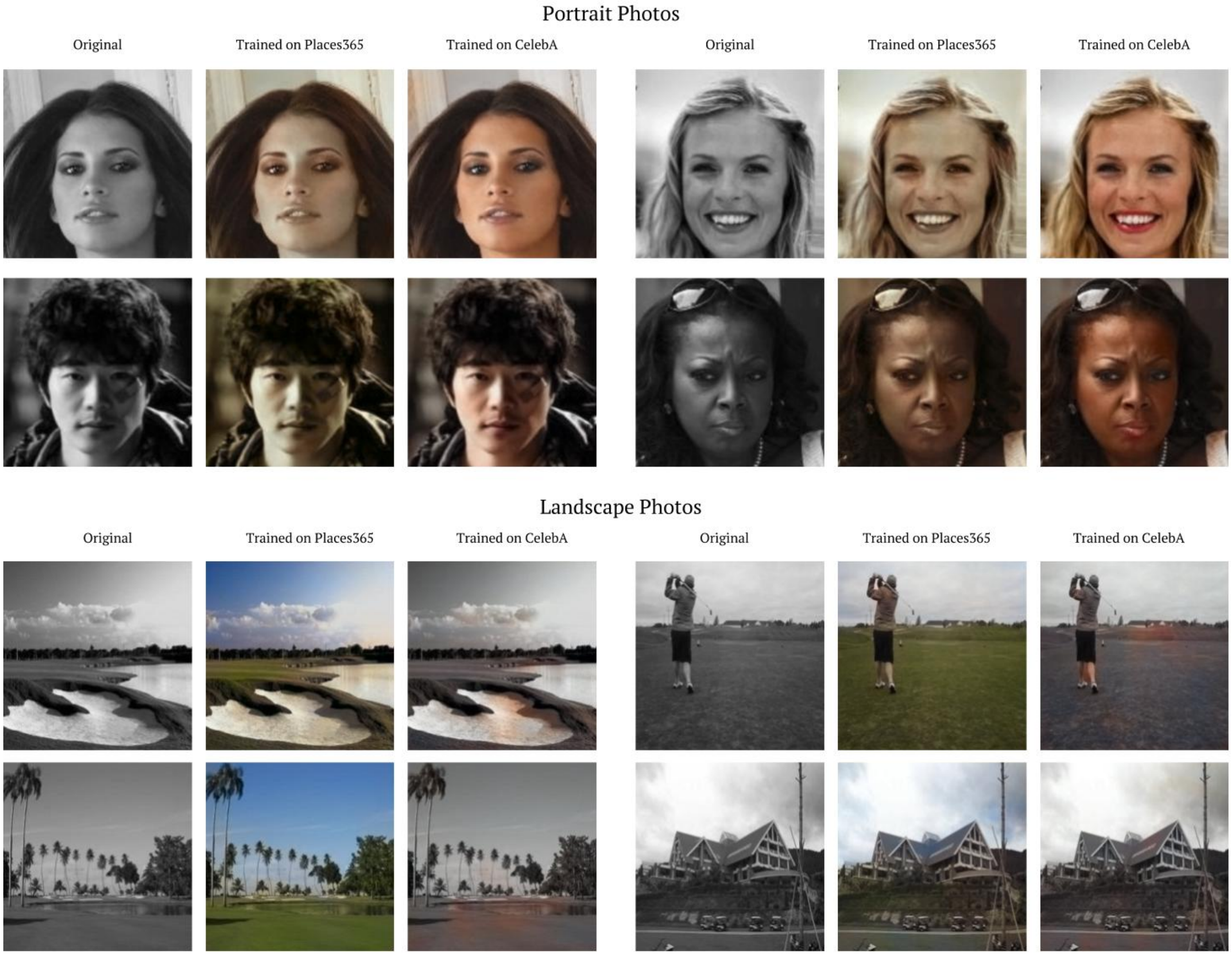

Figure 6. A diagram of our final single-stream network architecture.

normalized to sum to 1 over the 25 color bins.

In practice, this global histogram is often computed by taking the average of the color values presented in some other colored image. In the colorization literature, this technique is refered to as histogram transfer between images.

We find that transfering histograms from unrelated images usually only makes a small impact on the final colorized image outputted by our Potts Model. In certain cases, however, when the image being colorized contains objects of ambigious color, the global histogram transfer skews the result toward particular colors. In Figure 7, for example, the true color of the bird is unclear; the color distributions for the pixels composing the bird are relatively close to a uniform distribution, making the global histogram transfer somewhat (qualitatively) successful.

4.3. Historical Colorizations

Finally, we apply our first model to colorize a number of historical images. Colorizing historical images is challenging for a network trained on modern images due to the artifacts and textures introduced by old film cameras. Nonetheless, we run our model on a number of historical portraits and landscapes, shown in Figure 8 (at the of this paper). We believe that by training on old colored images, or by synthetically converting modern images to fake historical images before training our model, we may be able to achieve stronger performance on historical images.

We also ran our network on some hand-drawn black-and-white sketches, with our results shown in Figure 9; the network recognized and colorized some structures in the sketches, such as bushes and trees, although it did not colorize most objects in the sketches.

Finally, we colorize a scene from an old black-and-white

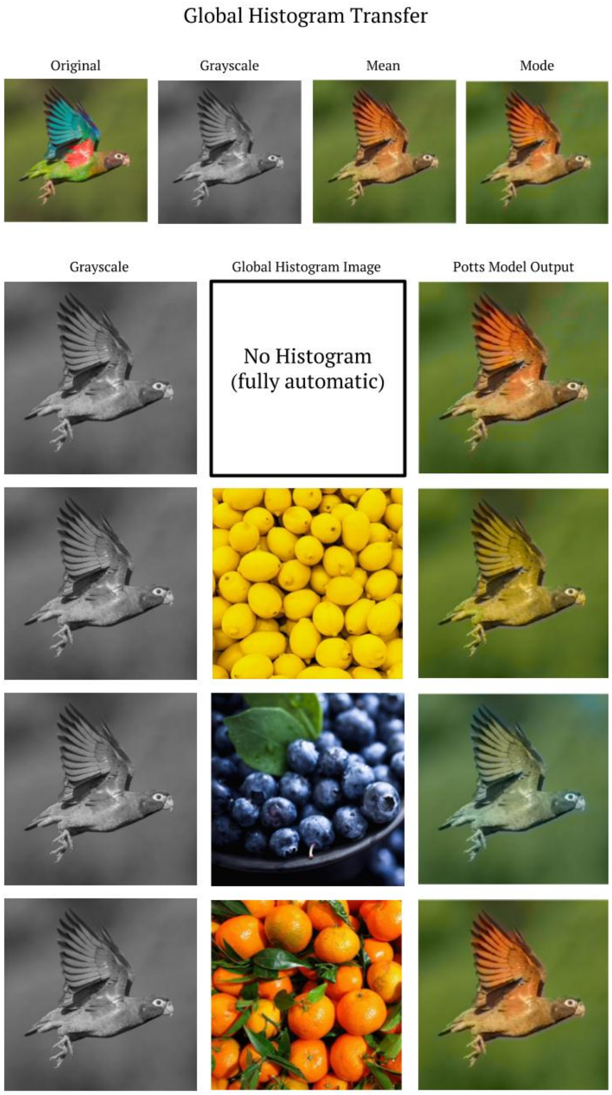

Figure 7. A demonstration of global histogram transfer.

film. We convert the video to still image frames, colorized each frame individually using our network, and recompiled the colored frames into a video. This frame-by-frame approach does not necessarily enforce consistency between frames, but still performs well. We specifically choose to colorize the scene "Charlie Chaplin in The Beach," as we trained our model on the Places365 dataset, which contains images of beaches. The results of our colorization may be viewed at the link below:

`https://youtu.be/LluZarKPY-o`

5. Conclusions and Future Work

In this project, we build and train a deep convolutional neural network for image colorization. Further, we propose a two-step colorization model that synthesizes a deep convolutional neural network with a Potts Model. This second model gives users an extra degree of control over their image through the input of a global color histogram. We train our colorizer on the Places365 and CelebA datasets, finding colorization performance to be strongly linked to the training dataset.

We look to extend upon this project in a number of different directions:

- Train a single network to perform well on both portraits (CelebA) and landscapes (Places365)
- Incorporate local user input into the model (as proposed by Zhang et al., 2017 [17])
- Use our colorization network for self-supervised representation learning (as proposed by Larsson et al., 2017 [8]).
- Incorporate an adversarial loss into our colorizer or experiment with colorization by generative adversarial networks [5]

Finally, we plan to colorize additional scenes from black-and-white films in the near future, and we hope to improve our network's performance on these scenes by training with synthetic historical photographs.

We have made our code and pre-trained weights for our model available on GitHub.

Acknowledgements

We would like to thank Professor Sasha Rush and teaching fellows Mark Goldstein, Rachit Singh, and Yuntian Deng for their assistance.

References

[1] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, (just-accepted), 2017.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[3] Jun-Hee Heu, Dae-Young Hyun, Chang-Su Kim, and Sang-Uk Lee. Image and video colorization based on prioritized source propagation. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 465–468. IEEE, 2009.

[4] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (TOG)*, 35(4):110, 2016.

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

[6] Zoltan Kato and Ting-Chuen Pong. A markov random field image segmentation model for color textured images. *Image and Vision Computing*, 24(10): 1103–1114, 2006.

[7] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European Conference on Computer Vision*, pages 577–593. Springer, 2016.

[8] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. *arXiv preprint arXiv:1703.04044*, 2017.

[9] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM transactions on graphics (tog)*, volume 23, pages 689–694. ACM, 2004.

[10] Stan Z Li. Markov random field models in computer vision. In *European conference on computer vision*, pages 361–370. Springer, 1994.

[11] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

[12] Qing Luan, Fang Wen, Daniel Cohen-Or, Lin Liang, Ying-Qing Xu, and Heung-Yeung Shum. Natural image colorization. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 309–320. Eurographics Association, 2007.

[13] Xiao-Hui Wang, Jia Jia, Han-Yu Liao, and Lian-Hong Cai. Affective image colorization. *Journal of Computer Science and Technology*, 27(6):1119, 2012.

[14] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. Transferring color to greyscale images. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 277–280. ACM, 2002.

[15] Liron Yatziv and Guillermo Sapiro. Fast image and video colorization using chrominance blending. *IEEE Transactions on Image Processing*, 15(5):1120–1129, 2006.

[16] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016.

[17] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017.

[18] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.

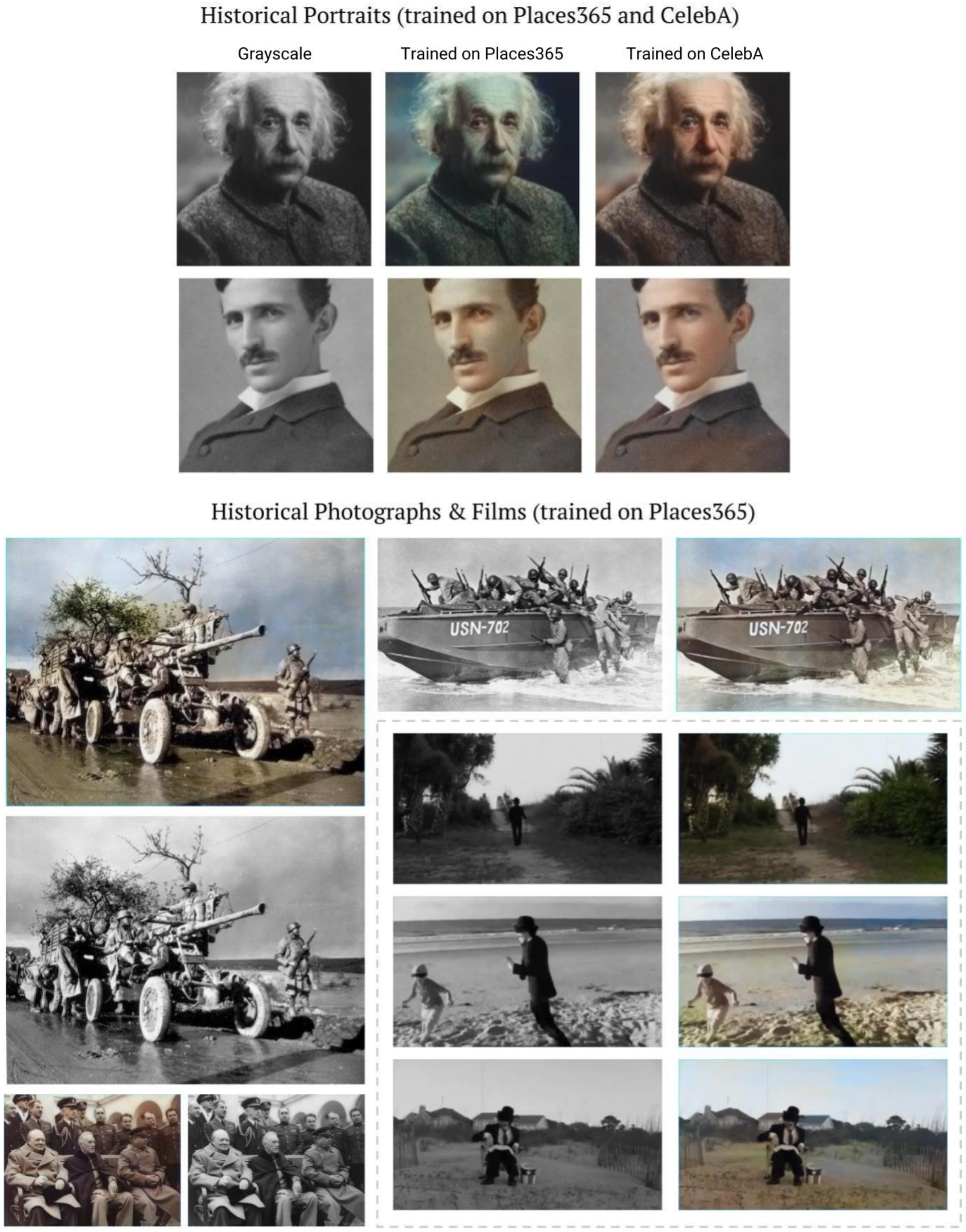
Historical Portraits (trained on Places365 and CelebA)
Grayscale
Trained on Places365
Trained on CelebA
Historical Photographs & Films (trained on Places365)
USN-702
USN-702

Figure 8. Colorizations of historical portraits, photographs from WWII, and a scene from a Charlie Chaplin film.

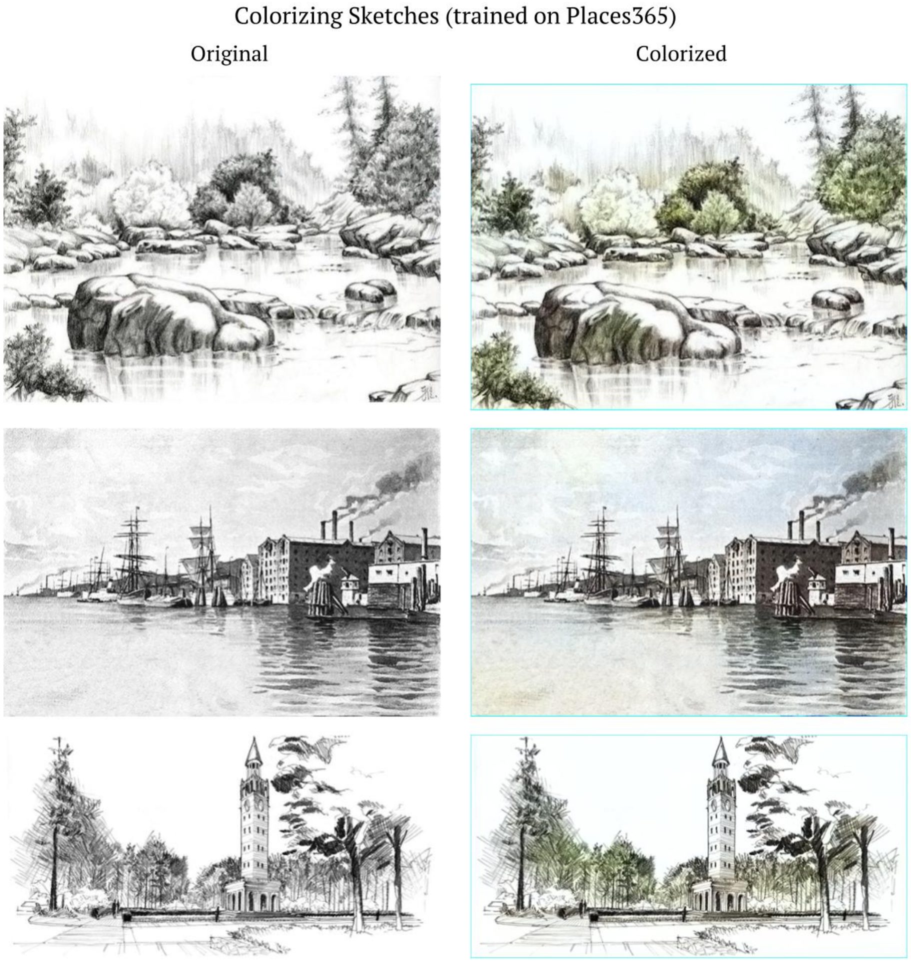

Figure 9. Colorizations of black-and-white sketches.