https://doi.org/10.2352/CIC.2023.31.1.24

A Simple Fast Resource-Efficient Deep Learning for Automatic Image Colorization

Tanmay Ambadkar[1], Jignesh S. Bhatt[2]
[1]The Pennsylvania State University, University Park, PA, USA
[2]Indian Institute of Information Technology Vadodara, Gujarat, India
tanmay@psu.edu, jignesh.bhatt@iiitvadodara.ac.in

Abstract

Colorization of grayscale images is a severely ill-posed inverse problem among computer vision tasks. We present a novel end-to-end deep learning method for the automatic colorization of grayscale images. Past methods employ multiple deep networks, use auxiliary information, and/or are trained on massive datasets to understand the semantic transfer of colors. The proposed method is a 38-layer deep convolutional residual network that utilizes the CIELAB color space to reduce the problem's solution space. The network comprises 16 residual blocks, each with 128 convolutional filters to address the ill-posedness of colorization, followed by 4 convolutional blocks to reconstruct the image. Experiments under challenging heterogeneous scenarios and using the Imagenet, Intel, and MirFlickr datasets show significant generalization when assessed visually and against PSNR, SSIM, and PIQE. The proposed method is relatively simpler (16 million parameters), faster (15 images/sec), and resource-efficient (just 50000 training images) when compared to the state-of-the-art.

Introduction

Images represent our visual perception through spatial patterns of brightness, shades, or colors. Many grayscale photographs were captured before the advent of a color camera and hence require *"colorization"*. Assigning color to an image is a challenging problem because much information has been lost. Understanding the image can help guess what color should be assigned, like blue to the sea or yellow to the sun. Moreover, the presence of multiple objects in an image, where each object can be made up of various colors, makes it a hard problem. Thus, given a grayscale image, estimating the corresponding color image is an ill-posed inverse problem with many possible solutions. Learning colorization function with a deep neural network is depicted in Fig. 1. Thus, The algorithm must first understand the image and guess what color can be assigned to parts based on learned semantics.

State-of-the-art methods typically train the networks with millions of image pairs to learn the semantics of an image. They use multiple GPUs to aid with training. This work proposes an efficient colorization network trained on significantly lesser image pairs on a readily available GPU. In summary, major contributions include:

1. A lightweight deep neural network (16 million parameters) comprised of convolutional residual blocks to learn the colorization function.
2. Prototype deep network with just 5% resources (50000 images) and 2 to 3 times faster colorization (15 images/sec) yields state-of-art performance.

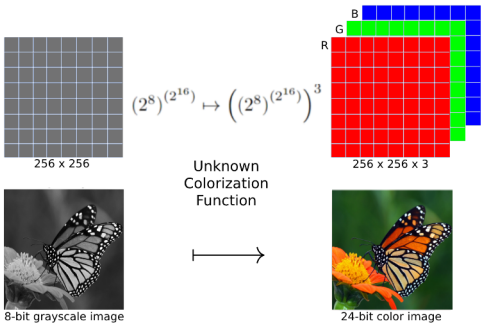

Figure 1. *Ill-posedness in learning image colorization function with a deep neural network $R^{n\times n} \mapsto (R\times R\times R)^{n\times n}$: For the given illustration, it turns out to be $3\% \times (2^8)^{(2^{16})} \mapsto 3\% \times \left((2^8)^{(2^{16})}\right)^3$ since statistically about 3% of possible distinct matrices represent visually meaningful spatial patterns referred to as the "images".*

Related Works

There are multiple approaches in which researchers have attempted to solve image colorization problem. Following is a quick review of three distinct categories:

1. Reference-based colorization: A grayscale image and a reference (color) image are presented to a model with very similar semantics as in the grayscale image. It attempts to identify the mapping and assigns colors from the reference to the grayscale image [17]. Recently, in [8], the authors employ deep learning to generate multiple reference images and use them for photo-realistic colorization. Such methods depend on generated reference images' availability and/or accuracy.
2. User-defined colorization: The user provides high-level scribbles that guide the colorization process. An early attempt [9] learns to pass on these scribbles to appropriate parts of the input grayscale image by optimizing a quadratic cost function. A promising approach requires the user to provide color hints to be propagated in algorithm [19]. This suggests the scribbles to the user or can consider scribbles from the user to propagate colors to various parts of an image. Such methods rely on the user's scribbles and/or require trained individuals to suggest appropriate colors for

Figure 2. *A few results of proposed method in challenging scenarios include colorization of print-and-scan legacy photographs and colorizing natural images in heterogeneous environments under varying light conditions.*

each image. Sugawara et al. [15] introduce a new way to colorize images by modeling chrominance using a global graph that connects important pixels and local graphs to connect the global graph to other pixels.

3. Automatic colorization: This is generally achieved by convolutional neural networks (CNN) since they capture invariant and equivariant [11] representations from a set of images. It helps extract semantic information for various image processing tasks, including colorization [2, 16, 18, 20, 14]. Methods in this category involve training CNNs on massive datasets such as ImageNet. Some models like [2, 16] use pre-trained networks, say, VGGNET and RESNET, for transferring learned semantics for the colorization task. In [18], the colorization is formulated as a classification problem wherein each pixel is assigned one of the 313 colors from the AB color space. An open-source colorization library [14] uses image-to-image generative adversarial networks; the community regularly updates it and serves as a good reference. Recent approaches like [13] use diffusion models for image-to-image translation tasks, one of which includes colorization. Zhou et al. [21] generate multiple color spaces with some randomness. It shows that good colorization can be achieved using a suitable color space.

One may notice that assigning colors to grayscale images with brightness variations over the foreground and background in a scene is challenging. It is found that most deep learning-based approaches either adopt weights from an existing model, are limited when attempting to colorize multiple objects within an image, or exhibit inconsistent colorization due to color bleeding from edges. As shown in Fig. 2, the proposed method found excellent results in colorizing legacy photographs (print-and-scan), natural imagery like jungles, mountains, humans, animals, and birds in the heterogeneous background, and so & so forth.

Proposed Method

In this section, we propose a simple deep neural network for the colorization of grayscale images and provide its architectural details along with a discussion on ill-posedness and complexity analysis. We further refer to Occam's Razor on achieving a simpler solution among multiple solutions, while achieving state-of-the-art performance. Given a grayscale image, it performs perception-consistent automatic colorization without any reference image and/or without user input for propagating color information. We construct an end-to-end deep convolutional residual network in CIELAB space. As shown in Fig. 3, it is a 38-layer deep residual convolutional network that uses a grayscale image (L channel) to estimate corresponding A and B channels [1] (images). The estimated A and B channels are then combined with the input grayscale (L) image to construct an estimated color image. The network captures the underlying semantics within the image to identify the object(s) and/or background and assigns suitable colors.

The CIELAB is deemed the closest color model of how humans perceive and process colors in visual scenes [10]. Humans can visualize the brightness of an image as captured by the L channel. This is considered the grayscale part of a color image. The A channel captures the red-green while the B channel captures the yellow-blue space of a color image. Our proposed method considers the given grayscale image as the L channel and hence remains to estimate corresponding A and B channels. This itself is a step towards addressing ill-posedness by reducing the solution space, while achieving a simpler solution.

Architecture

As shown in Fig. 3, the network first constructs 128 convolutional filters with a 9×9 block size (conv1) to process the input L channel. The output of the resulting 128 channels is sequentially passed to 16 residual blocks, which have 128 convolutional filters each. The output of these residual blocks is given to a convolutional layer with 128 filters with a 3×3 block size. This output is then element-wise summed to the output of the conv1. After this step, the channels are reduced gradually, i.e., 64, 32, 16, and finally, 2 channels. These are then passed to a sigmoid activation function to squash the output to values between 0 and 1.

There is a total of 16 residual blocks. A single block (Fig. 3 sidebar) consists of two convolutional layers having 128 filters with 3×3 block size, followed by a batch normalization after every layer and a parameterized ReLU function [4] after every

layer. The output is element-wise summed up with the input to a residual block.

Addressing ill-posedness

Comparing the ill-posedness of colorization (Fig. 1) and the proposed network (Fig. 3), one can see that the use of CIELAB space has the inherent advantage of the presence of the brightness channel as an input that permits easier and enhanced estimation of colors in the reconstructed colorized output image. It helps to make the colorization a better-posed problem by reducing the solution space to estimate the remaining 2 channels. Using multiple residual layers in our network helps generate deep contextual spatial features with its invariant and equivariant representation properties [11], and hence capture semantic information for colorization. The residual blocks enable building the deeper network with stable training and augment in extracting important shades or brightness variations to help the colorization process by the convolutional layers. In addition, the proposed network does not use pooling layers, which subsample the image, losing finer details. This results in better semantic transfer of fine-grained details, yielding natural colorization. Note that our network is end-to-end compared to recent SOTA methods [8, 15, 13]. A work in [21] attempts to transfer the information from a global color space to a gray-scale image with randomness to generate multiple results. It tests the hypothesis only on two sets of images, bedrooms, and churches. On the other hand, we conduct experiments on print and scan legacy photographs to natural images (humans, birds, lakes, animals, buildings) in heterogeneous environments with varying light conditions (Fig. 2). We also conduct ablation experiments with two different loss functions, i.e., l_2 and l_1, to calculate the error between predicted and true AB channels.

Complexity Analysis

It is challenging to assess the complexity of a neural network directly. However, attempts can be made to quantify [7]. The number of parameters in a network gives an insight into how much space it occupies, as it is directly related to the memory. The inference colorization speed (images/sec) helps us understand how fast a network can process inputs, giving us a hint at the time complexity. In addition, the number of training image pairs indicates the resource efficiency of a method.

The number of parameters is calculated by summing up the number of weights and biases in each layer. Here, the inference speed is the number of images colorized per second. The higher the speed, the better the performance in real-time applications.

Table 1: Analytical Statistics

Method	Network	Resource	Colorization Speed
	# Params (million) ↓	# Training image pairs (million) ↓	# images/sec↑
Zhang [18]	32	1.3	8
Zhang [19]	34	1	5
Deoldify [14]	42	1.2	3
Proposed Method	**16**	**0.05**	**15**

Table 1 lists the analytical statistics of proposed method and comparisons with SOTA. We see that the SOTA use the entire ImageNet dataset with 1.3 Million training pairs. The proposed method cuts down on that number by using only 50000 images or 5% of the ImageNet dataset. Along with this, the number of parameters indicate the space occupied by a model. Referring to Table 1, the proposed network uses half the number of parameters to achieve a similar performance. Having lesser parameters and training images mean that the model can be trained on a relatively moderate hardware, making the model easily accessible.

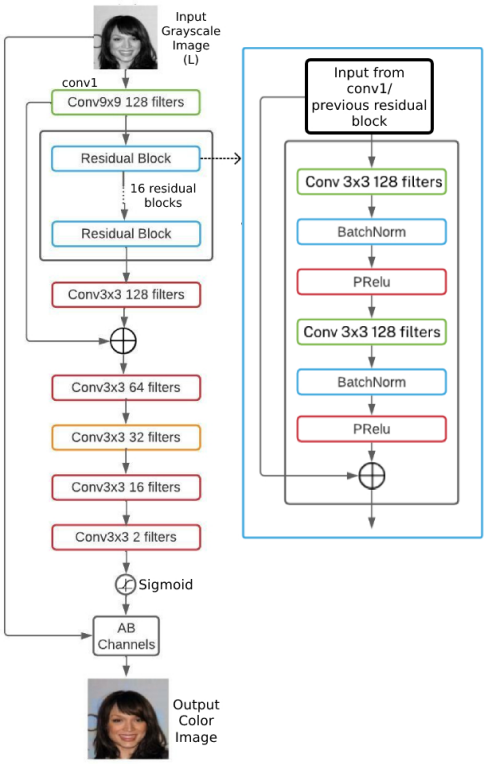

Figure 3. *Proposed 38-layer end-to-end deep convolutional residual neural network for automatic image colorization in CIELAB.*

Inference speed is further calculated to assess the speed of a method in many applications. It is clear that higher inference (colorization) speed shows a more efficient model. This is inversely related to the number of parameters and the pre and post processing being done on the images. Keeping the image size constant (144×144) for all models and the hardware same, we obtain the number of images colorized per second. One can see from Table 1 that the proposed model nearly doubles the inference speed when compared to the SOTA models. At 15 images/sec, the network can be used in many real-time colorization applications like video colorization, or it can be deployed on the edge/low powered hardware for inference at lowered cost.

Invoking Occam's Razor

State-of-the-art colorization algorithms currently deal with low-resolution images. Therefore, the available algorithms attempt to fill up the missing details with auxiliary images, making the method complicated. On the other hand, in this work, the proposed algorithm only needs to look at the coarse details to understand the color semantics. In addition, the major focus is on the subject of the image, which a convolutional neural network can isolate. When given enough diverse samples to understand the semantics of common objects, the proposed colorization al-

gorithm can learn the pattern and guess the color of unseen images exhibiting similar characteristics. A lightweight network is employed with far fewer image pairs for training to show that semantic understanding is achieved for this task and the network produces state-of-the-art results.

Experimental Results

This section presents the results obtained for the proposed method and compares performance with SOTA methods. For a fair comparison,The results are compared with end-to-end networks [19, 18, 14] for which the weights/codes are available in the public domain. The proposed network is trained on 50000 images randomly sampled from the ImageNet dataset. The color ground truths are produced as available in [3, 5, 6]. All images are resized to 144×144 for training. The Adam optimizer with a learning rate 0.0001 is used for 200 epochs. PyTorch is used as our library for defining and training the network. [1] The performances are assessed with 1000 test images randomly sampled from the ImageNet, Intel Image Classification, and MirFlickr25K test datasets. The visual results are shown first and then quantitative comparisons using peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and perception-based image quality evaluator (PIQE) [12].

Fig. 4 shows visual results of image colorization. Here, we display 5 grayscale images, their corresponding color ground truths, and estimated colorized outputs using different methods. As seen in Fig. 4, the proposed network provides more saturated colors when compared to SOTA. It generates consistent colorization as found in the color ground truth. On a close inspection of results in Fig. 4, one can see that the proposed model trained with l_1 loss performs visually better colorization than trained with l_2 loss. It can also capture minute details while the model trained with l_2 loss has missed. For instance, refer to the bird image in Fig. 4. It can be observed that the background is not colored completely in the output of the l_2 model. The l_1 model can assign the green background the l_2 model missed. Consider another instance of the tiger image. Other methods produce desaturated colors that are not similar to the ground truth or the actual color of the tiger. The proposed method produces the right colors and a saturated image, making it pleasing to the human eye. It shows that humans prefer boosted and saturated images to dull images. Referring to Fig. 4, similar better visual results by the proposed method can be seen for the lake, human, and building images. One may also refer to Fig. 2 for more visual results on challenging scenarios.

It is a known fact that quantifying the image quality is challenging, and one often needs to weigh the measures depending on the application. Here, we rely on SSIM, PIQE, and PSNR. Table 2 lists quantitative comparisons with recent SOTA using three datasets. It can be seen from Table 2 that the proposed network outperforms all related SOTA methods. In addition, one can see that the network trained with the l_1 loss model performs better in SSIM and PSNR, while network trained with l_2 loss performs better with respect to PIQE. Since PIQE is a blind metric assessing human perception using a mean squared implementation, it is closer to the l_2 loss and thus performs better. This is evident from our experiment conducted with the different losses. Let us consider the image of the tiger from Fig. 4. The colors produced by the reference SOTA methods are dissimilar to the color ground truth. This reduces their structural similarity and increases the difference with the ground truth, thus reducing the PSNR. Note that the proposed simple network, performs consistently better and addresses the ill-posedness of colorization without needing any auxiliary information.

Table 2: Quantitative comparisons with end-to-end colorization networks

Metric	Method	ImageNet	Intel	MirFlickr25
SSIM ↑	Zhang (2017) [19]	0.91	0.95	0.89
	Zhang (2016) [18]	0.85	0.93	0.86
	DeOldify [14]	0.89	0.91	0.87
	Proposed Method (l_2)	0.91	0.94	0.89
	Proposed Method (l_1)	**0.92**	**0.95**	**0.91**
PIQE ↓	Zhang (2017) [19]	13.89	9.65	16.86
	Zhang (2016) [18]	13.84	9.77	16.94
	DeOldify [14]	13.45	14.59	15.57
	Proposed Method (l_2)	**12.61**	**8.88**	**14.97**
	Proposed Method (l_1)	12.76	9.15	15.29
PSNR ↑	Zhang (2017) [19]	23.7 ± 4.3	24.9 ± 5.7	22.9 ± 5.1
	Zhang (2016) [18]	21.8 ± 3.4	23.5 ± 3.2	20.9 ± 3.7
	DeOldify [14]	21.9 ± 0.1	22.7 ± 3.2	20.9 ± 4.51
	Proposed Method (l_2)	22.7 ± 3.8	24.6 ± 5.7	22.2 ± 4.5
	Proposed Method (l_1)	$\mathbf{24.2 \pm 3.7}$	$\mathbf{25.7 \pm 0.1}$	$\mathbf{23.6 \pm 6.7}$

Conclusion and Future scope

This paper has made automatic image colorization a better-posed problem by presenting a novel 38-layer deep residual network. Compared to existing methods, it is a simpler, faster and resource-efficient end-to-end network that does not need any auxiliary information, and is more suitable for video colorization as well as in many hardware constrained scenarios. Through visual inspection and quantitative experiments, it is demonstrated that the proposed network performs better than current state-of-the-art methods. We observed that the network trained with the l_1 loss had produced more saturated and pleasing colors, while the network trained with the l_2 loss sometimes generated muted colors. Besides, the proposed network has the potential for generalizing to other tasks. In the future, one may pretrain the model using image-to-image and image-to-text tasks to learn better semantic representations before applying them to colorization.

References

[1] Arash Akbarinia and Raquel Gil-Rodríguez. "Color Conversion in Deep Autoencoders". In: *Journal of Perceptual Imaging* 4.2 (2021), pp. 20401-1–20401-1. DOI: 10.2352/J.Percept.Imaging.2021.4.2.020401. URL: https://library.imaging.org/jpi/articles/4/2/art00002.

[2] Federico Baldassarre, Diego González Morın, and Lucas Rodés-Guirao. "Deep Koalarization: Image Colorization using CNNs and Inception-ResNet-v2". In: *CoRR* abs/1712.03400 (2017). arXiv: 1712.03400. URL: http://arxiv.org/abs/1712.03400.

[3] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *Computer Vision and Pattern Recogni-*

[1] The weights of the proposed method and more ablation experiments are available on https://github.com/TanmayAmbadkar/ImageColorNet-Residual-Colorization.

Figure 4. *Visual results and ablation analysis of proposed method for perceptual comparison with state-of-the-art.*

tion (2009), pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[4] Kaiming He et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *CoRR* abs/1502.01852 (2015). arXiv: 1502.01852. URL: http://arxiv.org/abs/1502.01852.

[5] Mark J. Huiskes and Michael S. Lew. "The MIR Flickr Retrieval Evaluation". In: *MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval*. Vancouver, Canada: ACM, 2008.

[6] Intel. *Intel Image Classification*. Jan. 2019. URL: https://www.kaggle.com/datasets/puneet6060/intel-image-classification.

[7] RICH LEE and ING-YI CHEN. "The Time Complexity Analysis of Neural Network Model Configurations". In: *2020 International Conference on Mathematics and Computers in Science and Engineering (MACISE)*. 2020, pp. 178–183. DOI: 10.1109/MACISE49704.2020.00039.

[8] Chenyang Lei, Yue Wu, and Qifeng Chen. "Towards Photorealistic Colorization by Imagination". In: *CoRR* abs/2108.09195 (2021). arXiv: 2108.09195. URL: https://arxiv.org/abs/2108.09195.

[9] Anat Levin, Dani Lischinski, and Yair Weiss. "Colorization Using Optimization". In: *ACM Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH)*. SIGGRAPH '04 (2004), pp. 689–694. DOI: 10.1145/1186562.1015780. URL: https://doi.org/10.1145/1186562.1015780.

[10] Ming Ronnier Luo. "CIELAB". In: *Encyclopedia of Color Science and Technology* (2014), pp. 1–7. DOI: 10.1007/978-3-642-27851-8_11-1. URL: https://doi.org/10.1007/978-3-642-27851-8_11-1.

[11] Piduguralla Manaswini and Jignesh S. Bhatt. "Towards glass-box CNNs". In: *CoRR* abs/2101.10443 (2021). arXiv: 2101.10443. URL: https://arxiv.org/abs/2101.10443.

[12] Venkatanath N et al. "Blind image quality evaluation using perception based features". In: *Twenty First National Conference on Communications (NCC)* (2015), pp. 1–6. DOI: 10.1109/NCC.2015.7084843.

[13] Chitwan Saharia et al. "Palette: Image-to-Image Diffusion Models". In: *CoRR* abs/2111.05826 (2021). arXiv: 2111.05826. URL: https://arxiv.org/abs/2111.05826.

[14] Antoine Salmona, Lucía Bouza, and Julie Delon. "DeOldify: A Review and Implementation of an Automatic Colorization Method". In: *Image Processing On Line* 12 (2022). https://doi.org/10.5201/ipol.2022.403, pp. 347–368.

[15] Mamoru Sugawara et al. “Local and Global Graph Approaches to Image Colorization”. In: *IEEE Signal Processing Letters* 27 (2020), pp. 765–769. DOI: 10.1109/LSP.2020.2994817.

[16] Domonkos Varga and Tamás Szirányi. “Fully automatic image colorization based on Convolutional Neural Network”. In: *International Conference on Pattern Recognition* (2016), pp. 3691–3696. DOI: 10.1109/ICPR.2016.7900208.

[17] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. “Transferring Color to Greyscale Images”. In: *ACM Transactions on Graphic* 21 (July 2002), pp. 277–280. DOI: 10.1145/566570.566576.

[18] Richard Zhang, Phillip Isola, and Alexei Efros. “Colorful Image Colorization”. In: *European Conference on Computer Vision* (2016), pp. 649–666. DOI: 10.1007/978-3-319-46487-9_40.

[19] Richard Zhang et al. “Real-Time User-Guided Image Colorization with Learned Deep Priors”. In: *ACM Transactions on Graphics* 36.4 (2017), pp. 1–11.

[20] Jiaojiao Zhao et al. “Pixel-level Semantics Guided Image Colorization”. In: *CoRR* abs/1808.01597 (2018). arXiv: 1808.01597. URL: http://arxiv.org/abs/1808.01597.

[21] Jinjie Zhou et al. “Progressive Colorization via Iterative Generative Models”. In: *IEEE Signal Processing Letters* 27 (2020), pp. 2054–2058. DOI: 10.1109/LSP.2020.3037690.

Author Biography

Tanmay Ambadkar *is pursuing his M.S. in Computer Science and Engineering from Pennsylvania State University, USA. He graduated with a B.Tech from the Indian Institute of Information Technology Vadodara, India. He specializes in Deep Learning in computer vision problems and reinforcement learning.*

Jignesh S Bhatt *received his B.E. in EC from GEC Modasa (2000), M.Tech. in CS from SVNIT Surat (2008), and PhD in ICT from DAIICT Gandhinagar (2015). He has over 16 years of teaching experience and currently working at IIIT Vadodara (Gandhinagar). He has a long association working on various research projects with the Departmnet of Space, ISRO, India. His research interests include computer vision and deep learning for remote sensing, medical imaging, and defense applications.*