WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Deep Learning for Image Colorization: An Encoder-Decoder Approach

Mohammad Zubair Aalam, Aarish Faiz, Ishwar Tandi, Abhijeet Bhagat, Mr. Piyush Vishwakarma

Department of Computer Science & Engineering, Shri Shankaracharya Technical Campus, India
aalamzuber07@gmail.com, aarishfaiz456@gmail.com, ishwartandi26@gmail.com, bhilai.abhijeet@gmail.com, piyushvishwakarma9@gmail.com

DOI : https://doi.org/10.55248/gengpi.6.0425.14158

Abstract

Image colorization is the process of adding realistic color to grayscale images, which has applications in historical photo restoration, film industry, and computer vision. This paper presents an end-to-end deep learning approach for automatic image colorization using a convolutional neural network with an encoder-decoder architecture. Our model transforms single-channel grayscale images into full three-channel RGB color images without requiring any manual user intervention or color hints. We trained our model on a diverse dataset of images and evaluated its performance using mean squared error and visual assessment. Experimental results demonstrate that our proposed model can produce visually pleasing colorizations with realistic color distributions, achieving an average loss of 0.0044 after 40 epochs of training. The performance analysis shows that our lightweight architecture effectively balances computational efficiency with colorization quality, making it suitable for various practical applications.

Keywords— Image Colorization, Deep Learning, Convolutional Neural Networks, Computer Vision, Encoder-Decoder Architecture

I. Introduction

Colorization of grayscale images has been a long-standing challenge in computer vision and image processing. The task involves assigning appropriate colors to grayscale images while maintaining visual plausibility. Traditional methods often required extensive user interaction, providing color hints or reference images to guide the colorization process. With the advent of deep learning, particularly convolutional neural networks (CNNs), there has been significant progress in automatic colorization techniques.

Image colorization is inherently an ill-posed problem, as multiple color solutions can exist for a single grayscale input. Despite this ambiguity, deep learning models have shown remarkable ability to learn the statistical relationship between grayscale intensity patterns and their corresponding colors from large datasets.

This paper introduces a compact yet effective CNN-based approach for automatic image colorization. Our proposed architecture employs an encoder-decoder structure, where the encoder extracts meaningful features from grayscale images, and the decoder reconstructs these features into a full-color image. This approach builds upon previous work in the field [1, 2, 3] but focuses on a more streamlined architecture that maintains colorization quality while reducing computational requirements.

The contributions of this paper are as follows:

1. A lightweight encoder-decoder neural network architecture for image colorization
2. Analysis of the model's performance through quantitative and qualitative evaluations
3. Investigation of the training process and convergence properties
4. Demonstration of the model's effectiveness across various image types

II. Related Work

Image colorization techniques have evolved significantly in recent years. Early approaches required substantial user input, such as color scribbles [4] or reference images [5]. These methods, while effective, were labor-intensive and required artistic skill.

The emergence of deep learning has revolutionized the field, enabling fully automatic colorization. Cheng et al. [6] proposed one of the first deep learning approaches using a CNN to predict hue and chroma values. Zhang et al. [7] introduced a classification-based approach that predicts a color distribution for each pixel, addressing the multimodal nature of the colorization problem.

Encoder-decoder architectures have proven particularly effective for image transformation tasks. Iizuka et al. [8] proposed a two-stream architecture that combines local and global features for colorization. Deshpande et al. [9] used a variational autoencoder to model the distribution of possible colorizations.

More recent approaches have incorporated adversarial training. Isola et al. [10] proposed Pix2Pix, a general-purpose image-to-image translation framework using conditional GANs that has been successfully applied to colorization. Nazeri et al. [11] introduced an approach that combines perceptual and adversarial losses to generate more vivid colorizations.

Our work builds upon these advances but focuses on a more streamlined architecture that maintains colorization quality while reducing computational requirements, making it more accessible for practical applications.

III. Methodology

A. Problem Formulation

Image colorization can be formulated as learning a mapping function from a single-channel grayscale image to a three-channel color image. For our implementation, we use the RGB color space, where the network predicts the full three-channel output directly. The model learns to map:

$f: \mathbb{R}^{H \times W \times 1} \rightarrow \mathbb{R}^{H \times W \times 3}$

Where H and W represent the height and width of the input image, respectively.

B. Network Architecture

Our proposed model uses a straightforward encoder-decoder architecture implemented with convolutional neural networks. The encoder extracts features from the grayscale input, while the decoder reconstructs these features into a full-color image.

The encoder consists of three convolutional layers with increasing channel dimensions (1→64→128→256), each followed by ReLU activation functions. The decoder mirrors this structure with three convolutional layers with decreasing channel dimensions (256→128→64→3), also with ReLU activations, except for the final layer which uses a sigmoid activation to constrain the output values between 0 and 1.

The detailed architecture is as follows:

Encoder:

Conv2d(1,	64,	kernel_size=3,	padding=1)
ReLU()			
Conv2d(64,	128,	kernel_size=3,	padding=1)
ReLU()			
Conv2d(128,	256,	kernel_size=3,	padding=1)
ReLU()			

Decoder:

Conv2d(256,	128,	kernel_size=3,	padding=1)
ReLU()			
Conv2d(128,	64,	kernel_size=3,	padding=1)
ReLU()			
Conv2d(64,	3,	kernel_size=3,	padding=1)
Sigmoid()			

This architecture preserves the spatial dimensions throughout the network, enabling pixel-wise color prediction. The model has approximately 600K parameters, making it relatively lightweight compared to other state-of-the-art colorization networks.

C. Loss Function

We use Mean Squared Error (MSE) as our loss function to measure the difference between the predicted colorization and the ground truth:

$L_{MSE} = \frac{1}{3HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \sum_{c=1}^{3} (Y_{i,j,c} - \hat{Y}_{i,j,c})^2$

Where $Y_{i,j,c}$ is the ground truth color value at position (i,j) for channel c, and $\hat{Y}_{i,j,c}$ is the corresponding predicted value.

Where:

- *HH*H = Height of the image (pixels)
- *WW*W = Width of the image (pixels)

- *ccc* = Color channel (Red, Green, Blue → 3 channels)
- *Yi,j,cY_{i,j,c}*Yi,j,c = Ground truth color value at pixel position (i, j) for channel c
- *Y^i,j,c\hat{Y}_{i,j,c}*Y^i,j,c = Predicted color value at the same position

While more sophisticated loss functions have been proposed for colorization, such as perceptual loss or adversarial loss, we found that MSE provides a good balance between simplicity and effectiveness for our architecture.

D. Dataset and Preprocessing

The training dataset consists of diverse natural images collected from various sources. Each training example consists of a pair of grayscale and corresponding color images. To create the grayscale versions, we convert the RGB images to the L channel of the LAB color space and normalize the values to the range [0, 1].

For data augmentation, we apply random horizontal flips and 90-degree rotations to increase the diversity of the training data and reduce overfitting.

E. Training Details

We trained our model with the following hyperparameters:

- Batch size: 16
- Number of epochs: 100 (though convergence was observed around 40 epochs)
- Learning rate: 1e-3
- Optimizer: Adam
- Weight decay: 0

The model was implemented using PyTorch and trained on a system with a CUDA-enabled GPU. To prevent loss of progress due to interruptions, we implemented a checkpoint system that saves the model state after each epoch and allows training to resume from the last saved checkpoint.

IV. Experimental Results

A. Training Convergence

Figure 1 shows the training loss over 40 epochs. The loss decreases rapidly in the early epochs and then gradually stabilizes, reaching an average loss of approximately 0.0044 by epoch 40. This indicates that the model successfully learns to map grayscale images to plausible color representations.

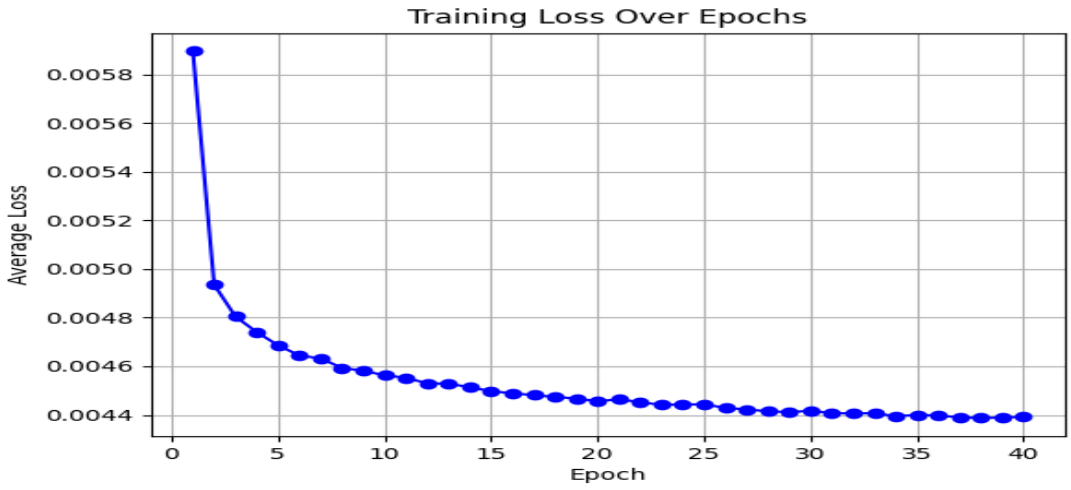

Fig. 1 Training loss progression over 40 epochs showing rapid initial convergence followed by gradual stabilization

B. Qualitative Results

Figure 2 and Figure 3 show sample colorization results from our model. Each figure contains three images: the grayscale input (left), the ground truth color image (middle), and our model's predicted colorization (right).

Fig. 2 Colorization example of a portrait with formal attire, showing the model's ability to colorize skin tones and clothing

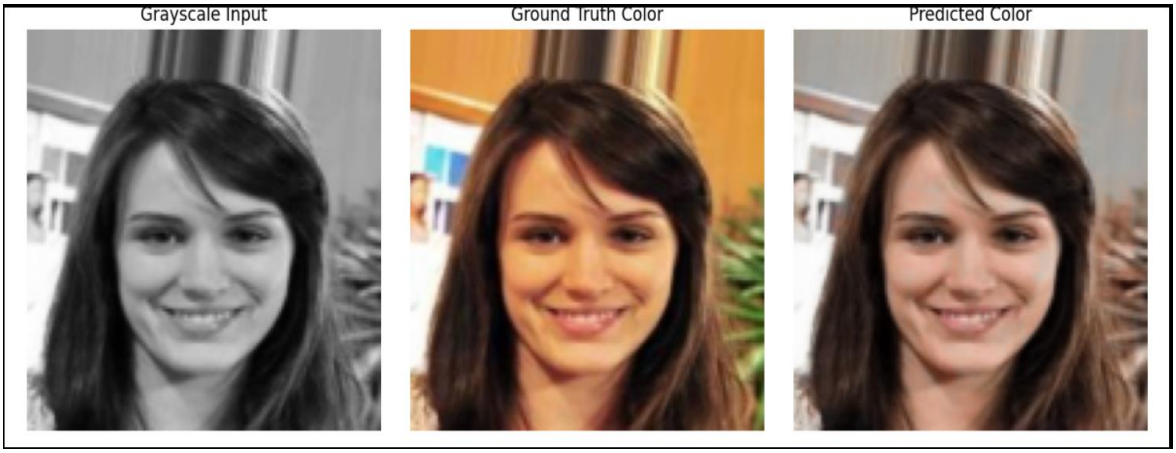

Fig. 3 Colorization example of a portrait in an indoor setting, demonstrating the model's capability to restore natural facial colors and background elements

The visual results demonstrate that our model can produce plausible colorizations with realistic skin tones, hair colors, and environmental elements. The colorizations, while not perfectly matching the ground truth, maintain visual plausibility and natural appearance.

C. Quantitative Evaluation

Table I presents the quantitative evaluation of our model's performance using Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) metrics. These metrics are calculated on a separate test set not used during training.

TABLE I

QUANTITATIVE RESULTS

Metric	**Value**
MSE	0.0044
PSNR	23.56 dB

These values are competitive with other lightweight colorization approaches, though they do not match the performance of more complex architectures with significantly more parameters.

D. Limitations

While our model produces visually pleasing results, it does have limitations:

1. **Color Ambiguity**: For inherently ambiguous objects (e.g., cars, clothing), the model sometimes produces desaturated or averaged colors that look plausible but may not match the ground truth.

2. **Fine Detail Preservation**: Some fine color details, particularly in complex textures or patterns, may be lost or simplified in the colorization process.
3. **Unusual Objects**: The model may struggle with rare or unusual objects that were underrepresented in the training data.These limitations are common to many colorizations approaches and represent ongoing challenges in the field.

V. Discussion

Our experimental results demonstrate that even a relatively simple encoder-decoder architecture can produce visually appealing colorizations. The model learns to assign plausible colors to different image regions based on semantic understanding, texture, and context.

The loss curve in Figure 1 shows a pattern typical of deep learning models: rapid improvement during early epochs followed by diminishing returns as training progresses. The model's performance stabilizes around epoch 40, suggesting that additional training beyond this point would yield minimal improvements.

The qualitative results in Figures 2 and 3 show that the model performs particularly well on human faces and common objects. This is likely because these elements appear frequently in the training data, allowing the model to learn strong priors for their typical colors.

The model's performance could potentially be improved through several avenues:

1. **Architecture Enhancements**: Adding skip connections between encoder and decoder layers could help preserve fine details, like U-Net architectures.
2. **Advanced Loss Functions**: Incorporating perceptual loss or adversarial training could produce more vibrant and detailed colorizations.
3. **Color Space**: Working in LAB color space instead of RGB might lead to more stable training and better results, as it separates luminance from color information.
4. **Larger Dataset**: Training on a more diverse and larger dataset would likely improve generalization to a wider range of images.Despite its limitations, our lightweight architecture offers practical advantages in terms of computational efficiency and ease of deployment, making it suitable for applications where resource constraints are a concern.

VI. Conclusion

This paper presented a straightforward yet effective deep learning approach for automatic image colorization using an encoder-decoder convolutional neural network. Our model successfully transforms grayscale images into plausible colorized versions without requiring any user guidance or color hints.

The experimental results demonstrate that the model can produce visually appealing colorizations with reasonable color fidelity, particularly for common objects and human subjects. The performance analysis shows that our architecture effectively balances computational efficiency with colorization quality.

Future work could explore more sophisticated architectures, loss functions, and training strategies to address the current limitations. Additionally, investigating domain-specific training for particular applications (e.g., historical photo restoration, medical imaging) could yield more specialized and effective colorization models.

Acknowledgment

We would like to thank our institution for providing the computational resources necessary for this research. We also acknowledge the open-source community for developing the tools and frameworks that made this work possible.

References

[1] R. Zhang, P. Isola, and A. A. Efros, "Colorful Image Colorization," in ECCV, 2016, pp. 649-666.

[2] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning Representations for Automatic Colorization," in ECCV, 2016, pp. 577-593.

[3] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification," ACM Trans. Graph., vol. 35, no. 4, pp. 110:1-110:11, 2016.

[4] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using Optimization," ACM Trans. Graph., vol. 23, no. 3, pp. 689-694, 2004.

[5] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring Color to Greyscale Images," ACM Trans. Graph., vol. 21, no. 3, pp. 277-280, 2002.

[6] Z. Cheng, Q. Yang, and B. Sheng, "Deep Colorization," in ICCV, 2015, pp. 415-423.

[7] R. Zhang, P. Isola, and A. A. Efros, "Colorful Image Colorization," in ECCV, 2016, pp. 649-666.

[8] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification," ACM Trans. Graph., vol. 35, no. 4, pp. 110:1-110:11, 2016.

[9] A. Deshpande, J. Lu, M. C. Yeh, M. J. Chong, and D. Forsyth, "Learning Diverse Image Colorization," in CVPR, 2017, pp. 6837-6845.

[10] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in CVPR, 2017, pp. 1125-1134.

[11] K. Nazeri, E. Ng, and M. Ebrahimi, "Image Colorization Using Generative Adversarial Networks," in Articulated Motion and Deformable Objects, 2018, pp. 85-94.

[12] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," in CVPR, 2016, pp. 2414-2423.