JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Historical Colorification using U-Net and CGAN

Pushpalata Verma[1]*, Vedant Fuley[2], V Jogendra Rao[3]
1*Assistant Professor: Bhilai Institute of Technology, Raipur
2,3 B. Tech Scholar: Bhilai Institute of Technology, Raipur

Abstract: The technique of turning grayscale photos into colored ones is known as image colorization, and it has several uses, such as enhancing visual media and restoring old photographs. Regression loss functions or classification loss functions can be used by an automatic colorization method to accomplish this transition. The classification loss function, on the other hand, might cause color overflow and necessitate substantial calculation for color categories and balance weights of the ground truth, whilst the regression loss function frequently produces brownish colors. This study uses a deep learning model built on the U-Net architecture to suggest a sophisticated method for image colorization. Our goal is to get beyond the drawbacks of conventional techniques by training the model on a sizable dataset of paired color and greyscale images. Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) are used to assess the model's performance; it received scores of 24.7595 and 0.9260, respectively. When compared to traditional algorithms, these results show a notable improvement in the quality and realism of colorized photographs. This development demonstrates how deep learning methods may produce high-quality colorization, opening the door for further advanced applications across a range of domains.

Keywords **- Image Colorization, Regression Loss, Classification Loss, U-Net Architecture, GAN (Generative Adversarial Networks).**

I. INTRODUCTION

In several fields, such as medical imaging, night vision, electron microscopy, satellite remote sensing, and the restoration of historical photographs, image colorization has become an essential method for processing greyscale images. Because it is inherently ambiguous to assign colors to distinct locations, colorization is still a difficult and complex problem despite its significance. One article of clothing, for example, could be red, blue, brown, or any other color, making it more challenging to identify the colors precisely. Consequently, colorization remains a major problem in image processing activities.

The two main categories of traditional colorization techniques are color transfer using reference images [5–8] and color expansion through neighboring pixels [1–4]. However, both methods necessitate a significant amount of manual labor and mostly depend on the precision of the color marking or the choice of relevant reference photos. Numerous automatic colorization techniques based on Convolutional Neural Networks (CNNs) have been presented in recent years due to the growth of deep learning. The colorization process can be automated with the help of these algorithms. However, the majority of these methods make use of regression loss functions (such L1 and L2) [9–21], which analyze greyscale picture features and add color channels to produce the final colorized output. Although this approach has been usually successful, it frequently produces colorization with unsaturated and brownish tones, which reduces the vibrancy of the images that are produced.

Using a classification loss function, Zhang et al. [22] developed a colorization technique to solve this problem and provide more vivid and saturated hues. However, this method created the issue of color overflow, which causes unwanted color distortions in the resulting photos.

We provide a novel technique to address these problems, including the unsaturated and brownish coloration and the overflow problems related to classification loss functions. By calculating color categories and balancing the weights of the color channels, our method aims to increase color prediction accuracy while minimizing the problems associated with both regression and classification-based approaches and cutting down on the amount of time needed for these networks' training.

II. LITERATURE REVIEW

A. Conventional Colorization Techniques

Traditional colorization techniques, which fall into two primary categories—color mapping using reference images and color propagation using nearby pixels—heavily rely on human input. Despite being fundamental, these techniques are often labor-intensive and might not scale well for automated applications.

1) Color Propagation Using Neighboring Pixels

Levin et al. [1] first proposed the color propagation approach, showing that adjacent pixels with comparable greyscale intensities frequently have similar colors. They were able to apply the manually assigned colors throughout the entire image thanks to this finding. Yatziv et al. [2] expanded on this concept by introducing a weighted distance function to improve color propagation based on pixel proximity, thus refining the colorization process. Using textural information in

the image, Qu et al. [3] and Luan et al. [4] considerably decreased the process's processing cost. Although this method yielded acceptable colorizations, it had drawbacks, such as color misinterpretations brought on by erroneous or badly positioned initial color assignments, especially close to image boundaries.

2) Color Mapping Using Reference Images

Welsh et al. [5] introduced the color mapping technique, in which they compared a greyscale image to a color reference image and then applied the reference colors to the greyscale image's corresponding pixels. Although less manual input is required with this method, selecting a suitable reference image is still necessary. This was enhanced by Irony et al. [6] through the use of high-resolution reference photos and color transfer based on textural similarity. Liu et al. [7] used the internet to locate color photographs that were aesthetically comparable to greyscale images in order to get around the problem of finding appropriate reference images. This was further enhanced by Wang et al. [8], who used semantic analysis to determine the best reference images based on the greyscale image's content. Even with these improvements, the final coloration still mainly depended on the transfer procedure and the reference image, which frequently produced uneven results.

B. Deep Learning-Driven Colorization Methods

Deep learning has led to the development of fully automated colorization techniques. Regression loss functions and classification loss functions are the two main categories into which these techniques are usually divided.

1) Regression Loss Functions

A dominant approach in deep learning-based colorization is the use of regression loss functions (such as L1 and L2 losses). These methods focus on predicting color values for pixels by estimating the a and b channels in the Lab color space. Cheng et al. [9] combined CNN feature extraction with bilateral filtering to enhance the quality of colorization. Larsson et al. [10] used a deep VGG network to capture semantic features of the image and guided the colorization process based on hue and chroma histograms of individual pixels. Iizuka et al. [11] designed a two-channel CNN to extract global and local features separately, which were then fused to improve colorization accuracy. In a similar vein, Nazeri et al. [12] utilized Conditional Generative Adversarial Networks (cGANs) for colorization, which significantly improved the results. Patricia et al. [15] introduced ChromaGAN, which not only generates colorized images but also outputs category distributions for better color predictions. Other approaches, like those by Su et al. [19], Wu et al. [20], and Jin et al. [21], used GANs and additional semantic features to improve colorization. Despite these successes, regression loss functions remain prone to issues, particularly when it comes to assigning colors to various objects in the image. This uncertainty often results in brownish and unsaturated colorization effects.

2) Classification Loss Functions

An alternative approach to the regression loss function is the classification loss function, which was explored by Zhang et al. [22]. In their method, they created 313 color categories based on the a and b channels of the Lab color space. By calculating the geometric distance between the a and b values of each pixel and the nearest 32 color categories, they were able to build a probability distribution for each pixel's color category. This method used Gaussian weighting to select the color category with the highest probability, resulting in more vivid and saturated colors. To further enhance the colorization quality, they balanced the weights using the ImageNet dataset's color category distribution. However, although this technique produced more vibrant results, it suffered from color overflow due to the lack of effective fusion between global and local image features, making the colorization network inefficient and prone to over-saturation. Additionally, the method required long training times due to the large computational load involved in calculating the color categories and balancing the weights.

In conclusion, while deep learning has significantly advanced colorization techniques, challenges persist. Regression-based methods tend to produce unsaturated and brownish results, while classification-based methods offer more vibrant colorization but are hindered by issues like color overflow and excessive computational requirements. Researchers continue to seek innovative solutions to improve both the quality and efficiency of image colorization algorithms.

III. METHODOLOGY

A. Overview

In this study, we propose an end-to-end deep learning approach for colorizing grayscale images using a U-Net-based generator network. The objective is to predict the missing chromatic channels (a and b) in the Lab color space, given the luminance (L) channel. This method involves transforming a single-channel grayscale image $x_l \in \mathbb{R}^{H\times W}$ into a full-color image $x_{lab} \in \mathbb{R}^{H\times W\times 3}$ by predicting the a and b channels. In the Lab color space, the L channel represents brightness, while the a and b channels correspond to the color components ranging from red to green and from yellow to blue, respectively. Our approach leverages the power of convolutional neural networks (CNNs) and adversarial training through a Generative Adversarial Network (GAN) structure, using a U-Net architecture for the generator and a PatchGAN for the discriminator.

B. Generator Architecture

The generator network *G* is based on U-Net architecture, which is known for its effectiveness in pixel-level image generation tasks. The core of the generator is a convolutional encoder-decoder structure, where the encoder learns features from the input grayscale image (L channel), and the decoder reconstructs the predicted a and b channels.

1) U-Net Architecture:

a. Encoder: This part gradually reduces the spatial dimensions of the input while increasing the number of feature channels. It consists of several convolutional layers with ReLU activations and downsampling operations, enabling the network to learn hierarchical feature representations.

b. Decoder: The decoder gradually upsamples the feature maps produced by the encoder, ultimately producing the output color channels (a and b). Skip connections from the encoder to the decoder help preserve spatial information during the upsampling process.

In this specific implementation, we use a ResNet-based encoder (ResNet-18) combined with a U-Net decoder, leveraging the power of pretrained weights from the ResNet architecture to enhance performance in feature extraction. The decoder uses transposed convolutions to upsample the feature maps and generate high-resolution outputs.

2) Output Layer:

The output layer of the generator network is designed to predict the a and b channels, which are concatenated to the L channel to form the final colorized image in the Lab color space. The output of the generator is passed through a Sigmoid activation to ensure that the predicted color values fall within a specific range.

C. Loss Functions

To train the generator network, we use a combination of L1 Loss and GAN Loss.

a. **L1 Loss:** L1 loss measures the pixel-wise difference between the predicted a and b channels and the ground truth values. It is defined as:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\| y - G(x) \|_1]$$

Here, x is the grayscale image input, z as the input noise for the generator, y is the 2-channel output (2 color channels of a real image), and G(x) is the generated color channels by the generator. L1 Loss is used to ensure that the generated colors are close to the true color values, encouraging the generator to produce realistic outputs.

Using only L1 loss allows the model to still colorize images, but it tends to be overly conservative, often resorting to neutral colors like "gray" or "brown." This happens because when the model is uncertain about which color to apply, it opts for the average color to minimize the L1 loss. This behavior is similar to the blurring effect seen in super-resolution tasks. Additionally, L1 loss is preferred over L2 loss (mean squared error) because it helps mitigate the tendency to generate grayish images.

b. **GAN Loss:** GAN loss helps the generator learn to produce more realistic colorization outputs by encouraging it to fool the discriminator into classifying the generated images as real. The GAN loss is based on the output of a Patch-GAN discriminator, which classifies patches of the image as real or fake. The loss function used here is BCEWithLogitsLoss, which combines binary cross-entropy loss with a sigmoid activation, making it suitable for binary classification tasks.

Let *x* represent the grayscale image, *z* be the noise input for the generator, and *y* denote the desired 2-channel output from the generator (which can also correspond to the two-color channels of a real image). In this case, *G* refers to the generator model and *D* is the discriminator. The loss function for our conditional GAN can then be expressed as:

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[logD(x, y)] + \mathbb{E}_{x,z}\left[\log\left(1 - D\left(x, G(x, z)\right)\right)\right]$$

c. Combined Loss:

The total loss used to train the generator is a weighted sum of the L1 loss and the GAN loss:

$$G^* = arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$$

Here λ is a coefficient that controls the relative influence of the two losses in the overall loss function (naturally, the discriminator loss does not include the L1 loss).

D. Training Strategy

During training, the generator and discriminator networks are updated alternately. The discriminator is first trained to differentiate between real and generated images. Then, the generator is trained to produce colorized images that can fool the discriminator. This adversarial process continues iteratively, with the generator improving its output and the discriminator becoming more adept at detecting fake images.

The discriminator is trained to minimize the loss which consists of the real loss (real images labeled as real) and the fake loss (generated images labeled as fake). The generator is trained to minimize the loss, combining the adversarial loss and the L1 loss.

The optimization is done using the Adam optimizer, with different learning rates for the generator and discriminator to maintain stability during training. The generator's learning rate is typically set to 2×10^{-4}, and the discriminator's learning rate is set to 2×10^{-4} with betas (β1=0.5, β2=0.999) (β1=0.5, β2=0.999).

IV. RESULTS

A. Experimental Details

This project, which focuses on image colorization using a GAN-based architecture, was trained on Google Colab using a TPU for accelerated processing. The dataset used for training and validation is a subset of the COCO dataset, with 10,000 randomly selected images, split into 8,000 for training and 2,000 for validation. The input image size is standardized to 256x256 pixels to ensure consistency in the model's learning process. A batch size of 16 was used during training, with 50 epochs for the initial generator pretraining phase and further fine-tuning with adversarial training. The model utilizes a residual U-Net architecture for the generator and a Patch-Discriminator for the discriminator, with the GAN loss and L1 loss combined to guide the colorization task.

B. Quantitative Analysis

To quantitatively assess the performance of our colorization network, we use SSIM and PSNR as evaluation metrics. SSIM measures the similarity between a generated color image and the corresponding real image by considering factors like brightness, contrast, and structure. It is particularly sensitive to local structural differences between the two images. The SSIM value ranges from 0 to 1, where a higher SSIM value indicates greater similarity between the images. The SSIM is defined as follows:

$$l(x,y) = (2\mu_x \mu_y + C1)/({\mu_x}^2 + {\mu_y}^2 + C_1)$$

$$c(x,y) = (2\sigma_x\sigma_y)/({\sigma_x}^2 + {\sigma_y}^2 + C_2)$$

$$s(x,y) = (\sigma_{xy} + C_3)/(\sigma_x\sigma_y + C_3)$$

$$SSIM(x,y) = l(x,y)^{\alpha} * c(x,y)^{\beta} * s(x,y)^{\gamma}$$

In this context, μ_x and μ_y represent the mean values of images x and y, respectively; σ_x and σ_y denote the variance of image x and y, respectively; σ_{xy} refers to the covariance of image x and y; C_1, C_2, C_3 are constant; and $\alpha, \beta, and\ \gamma$, indicate the weights or importance assigned to each module.

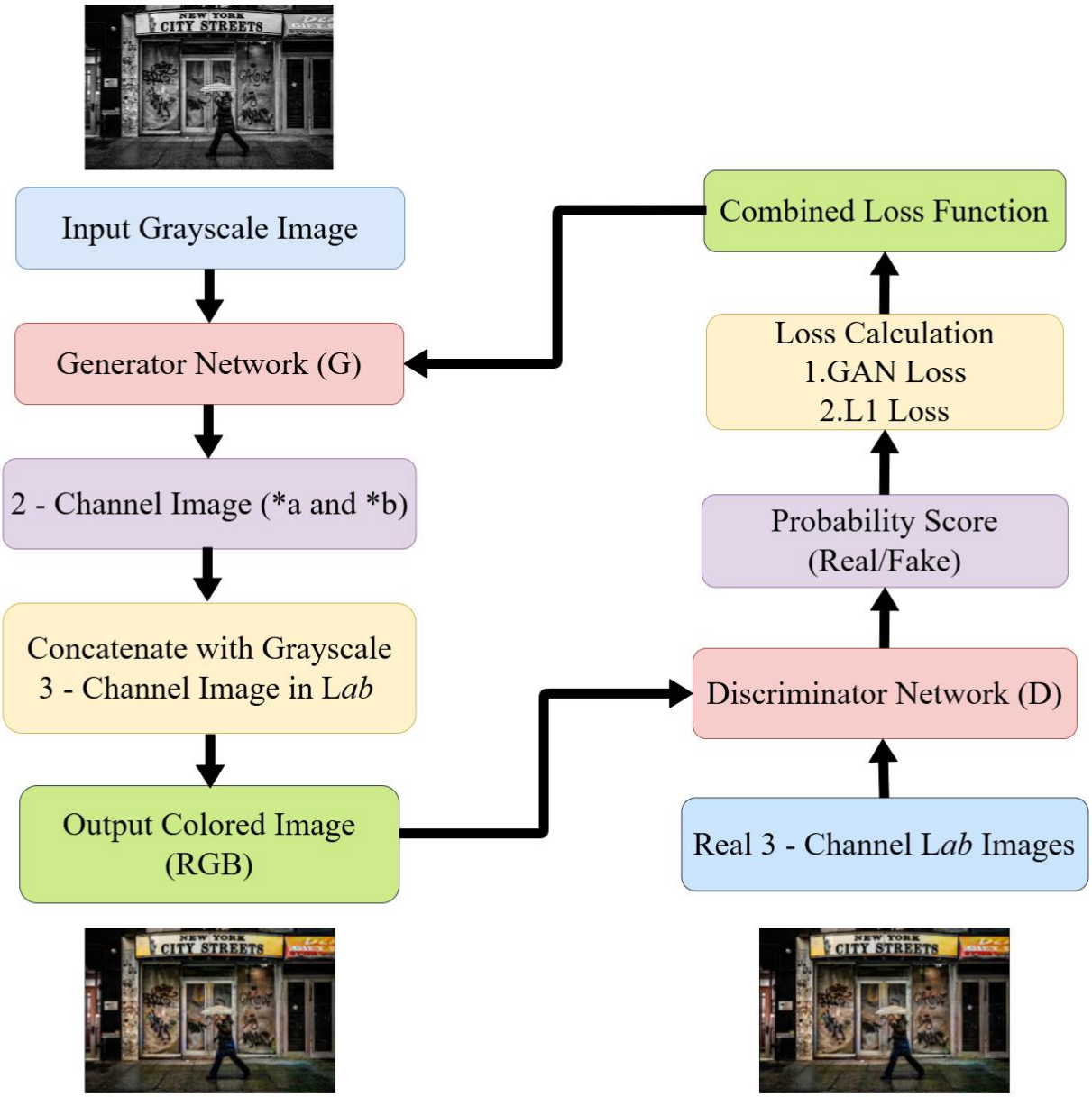

Figure 1 Flow chart depicting how the Image is converted from Grayscale Image to a Colorized Image.

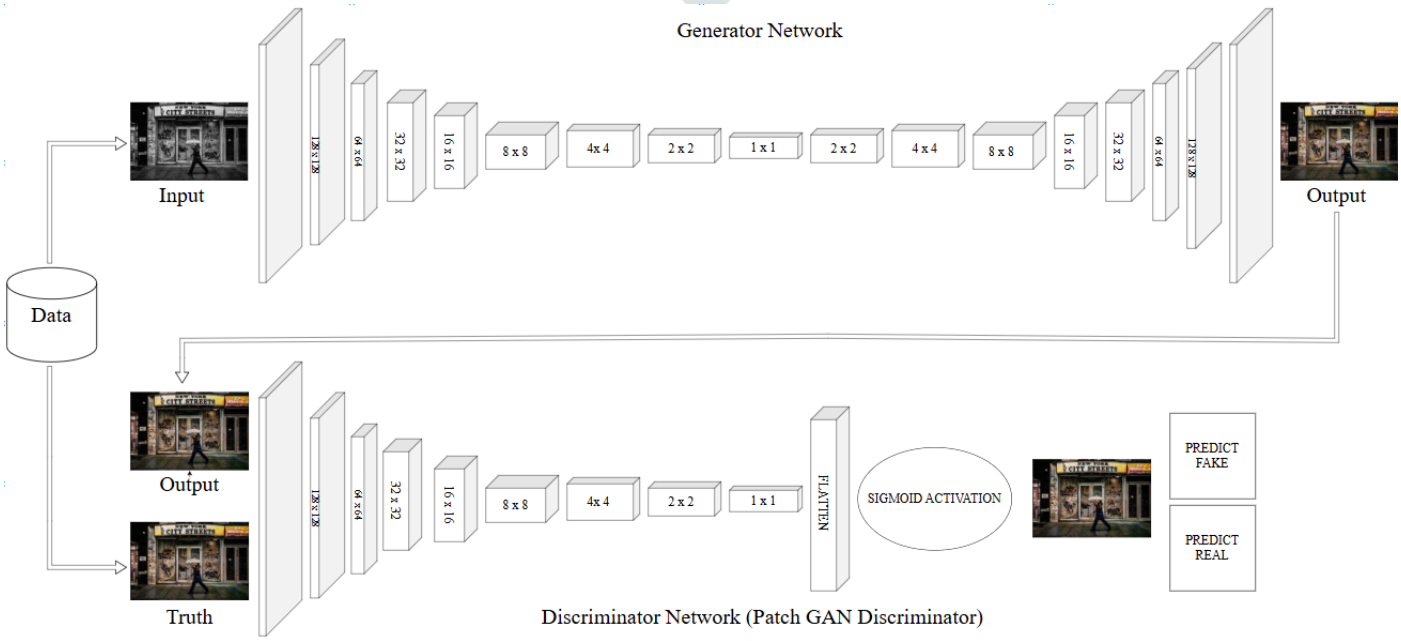

Figure 2 Generator and PatchGAN Discriminator

Table 1 Quantitative Analysis

Method	PSNR	SSIM
Zhang et al. [22]	21.7910	0.8915
Deoldify [18]	23.5372	0.9144
Iizuka et al. [11]	23.6362	0.9173
H. Shafiq et al. [5]	24.157	0.941
Ours	24.7595	0.9260

Table 1 displays the Quantitative analysis of the colorization effect shows that our network outperforms other colorization algorithms when comparing PSNR and SSIM values.

Figure 3. Comparison of Grayscale and Colorized Images. The Grayscale image is shown on the left, and the colorized output produced by the model is displayed on the right.

V. Conclusion

In conclusion, our advanced deep learning-based approach to image colorization, which leverages a U-Net architecture and a GAN framework, has demonstrated significant improvements in the quality and realism of colorized images. By training on a comprehensive dataset of paired grayscale and color images, the model effectively overcomes the limitations of traditional methods, such as the brownish and unsaturated tones produced by regression loss functions and the color overflow issues associated with classification loss functions. The quantitative evaluation metrics, specifically Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM), yielded impressive scores of 24.7595 and 0.9260, respectively. These results underscore the model's capability to generate high-quality, realistic colorizations.

Our methodology not only enhances the visual appeal of colorized images but also holds significant potential for diverse applications, including the restoration of historical photographs, enhancement of visual media, and real-time colorization for digital media and augmented reality. The integration of advanced architectures, larger datasets, and temporal consistency further paves the way for future advancements. This research highlights the promising role of deep learning techniques in achieving superior image colorization, offering valuable contributions to the fields of digital restoration, media production, and beyond.

VI. Acknowledgment

We would like to thank Prof. Pushpalata Verma, our supervisor, for their invaluable guidance, continuous encouragement, and expert advice throughout the course of this project. Their expertise in Computer Vision, Image Processing, ML, and AI greatly enhanced our research. We also acknowledge the Department of Computer Science of Bhilai Institute of Technology, Raipur, providing the necessary resources and infrastructure to carry out this research.

REFERENCES

[1] Anat Levin, Dani Lischinski, et.al, "Colorization using optimization", ACM Trans. Graph. 2004, 23, 689–694.
[2] L. Yatziv, G. Sapiro, et.al, "Fast image and video colorization using chrominance blending", IEEE Trans. Image Process. 2006, 15, 1120–1129.
[3] Yingge Qu, Tien-Tsin Wong, et.al, "Manga Colorization" ACM Trans. Graph. 2006, 25, 1214–1220.
[4] Qing Luan, Fang Wen, et.al, "Natural Image Colorization", In Proceedings of the 18th Eurographics Conference on Rendering Techniques, Grenoble, France, 25 June 2007; pp. 309–320.
[5] Tomihisa Welsh, Michael Ashikhmin, et.al, "Transferring color to greyscale images", ACM Trans. Graph. 2002, 21, 277–280.
[6] Revital Irony, Daniel Cohen-Or, et.al, "Colorization by Example", In Proceedings of the Eurographics Symposium on Rendering (2005), Konstanz, Germany, 29 June–1 July 2005.
[7] Xiaopei Liu, Liang Wan, et.al, "Intrinsic Colorization", In Proceedings of the ACM SIGGRAPHAsia2008, New York, NY, USA, 1 December 2008.
[8] Xiao-Hui Wang, X.; Jia Jia, et.al, "Affective Image Colorization", Journal of Computer Science and Technology 2012, 27, 1119–1128.
[9] Zezhou Cheng, Qingxiong Yang, et.al, "Deep Colorization", In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 415–423.
[10] Gustav Larsson, Michael Maire et.al, "Learning Representations for Automatic Colorization", In Lecture Notes in Computer Science; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Cham, Switzerland, 2016; Volume 9908.
[11] Santoshi Iizuka, Edgar Simo-Serra, et.al, "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. 2016, 35, 1–11.
[12] Kamyar Nazeri, Eric Ng, et.al, "Image Colorization Using Generative Adversarial Networks", In International Conference on Articulated Motion and Deformable Objects; Springer: Berlin/Heidelberg, Germany, 2018; Volume 10945, pp. 85–94.
[13] Phillip Isola, Jun-Yan Zhu, et.al, "Image-to-Image Translation with Conditional Adversarial Networks", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
[14] Yun Cao, Zhiming Zhou, et.al, "Unsupervised Diverse Colorization via Generative Adversarial Networks" In Joint EuropeanConference on Machine Learning and Knowledge Discovery in Databases; Springer: Berlin/Heidelberg, Germany, 2017; pp. 151–166.
[15] Patricia Vitoria, Lara Raad, et.al, "ChromaGAN: Adversarial Picture Colorization with Semantic Class Distribution", In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, 1–5 March 2020; IEEE: New York, NY, USA, 2020; pp. 2434–2443.
[16] Richard Zhang, Jun-Yan Zhu, et.al, "Real-time user-guided image colorization with learned deep priors", ACM Trans. Graph. 2017, 36, 1–11.
[17] Jiaojiao Zhao, Jungong Han, et.al, "Pixelated Semantic Colorization", International Journal of Computer Vision 2019, 128, 818–834.
[18] Antic, J. Jantic/Deoldify: "A Deep Learning Based Project for Colorizing and Restoring Old Images (and Video!)", Available online:https://github.com/jantic (accessed on 16 October 2019).
[19] Jheng-Wei Su, Hung-Kuo Chu, et.al, "Instance-Aware Image Colorization", In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7965–7974.
[20] Yanze Wu, Xintao Wang, et.al, "Towards Vivid and Diverse Image Colorization with Generative Color Prio", In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 14377–14386.
[21] Xin Jin, Zhonglan Li, et.al, "Focusing on Persons: Colorizing Old Images Learning from Modern Historical Movies", In Proceedings of the 29th ACM International Conference on Multimedia, Virtual Event, 20–24 October 2021; pp. 1176–1184.
[22] Richard Zhang, Phillip Isola, et.al, "A. Colorful Image Colorization", In European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2016; pp. 649–666.
[23] Olaf Ronneberger, Phlipp Fischer, et.al, "U-Net: Convolutional networks for biomedical image segmentation", In Medical Image Computing and Computer-Assisted Intervention 2015; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
[24] Sung-Jin Cho, Seo-Won Ji, et.al, "Rethinking Coarse-to-Fine Approach in Single Image Deblurring", In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 4621–4630.
[25] Hamza Shafiq, Bumshik lee et.al, "Image Colorization Using Color-Features and Adversarial Learning", Digital Object Identifier. 2022
[26] Zhang, R., Isola, P., et al "Colorful Image Colorization." IEEE Transactions on Pattern Analysis and Machine Intelligence. 2024
[27] Aryan Raj Tiwary, Aditya KR. Gupta et.al, "Image Colorization and Restoration using Deep Learning", Artificial Intelligence and Machine learning Techniques in Image Processing and Computer Vision, page-14, 2024
[28] Hong-an Li, Diao Wang et.al, "Image Color Rendering based on frequency channel attention GAN", Signal, Image and Video Processing, Volume 18, pages 3179-3186, 2024
[29] Himanshu Kumar, Abeer Banerjee, et.al, "ParaColorizer – Realistic image colorization using parallel generative networks", The Visual Computer, Volume 40. Pages 4039-4054, 2024
[30] Hamza Shafiq, Bumshik lee et.al, "Image Colorization Using Color-Features and Adversarial Learning", Digital Object Identifier. 2023
[31] Y. Lu, X. Huang, et.al, ''ColorGAN: Automatic image colorization with GAN", in Proc. IEEE 3rd Int. Conf. Inf. Technol., Big Data Artif. Intell. (ICIBA), vol. 3. Chongqing, China, May 2023, pp. 212–218, doi: 10.1109/ICIBA56860.2023.10164924.

Snitha Shetty, Radha Shri Vamsi et.al, "Image Colorization and Detection using Machine Learning", International Journal of Creative Research Thoughts (IJCRT), Volume 11, Issue 5 May 2023.