Multimedia Tools and Applications
https://doi.org/10.1007/s11042-023-15330-z

# TIC: text-guided image colorization using conditional generative model

**Subhankar Ghosh[1] · Prasun Roy[1] · Saumik Bhattacharya[2] · Umapada Pal[3] · Michael Blumenstein[1]**

Received: 26 July 2022 / Revised: 22 September 2022 / Accepted: 6 April 2023


**Abstract**

Image colorization is a well-known problem in computer vision. However, due to the ill-posed nature of the task, image colorization is inherently challenging. Though several attempts have been made by researchers to make the colorization pipeline automatic, these processes often produce unrealistic results due to a lack of conditioning. In this work, we attempt to integrate textual descriptions as an auxiliary condition, along with the grayscale image that is to be colorized, to improve the fidelity of the colorization process. To the best of our knowledge, this is one of the first attempts to incorporate textual conditioning in the colorization pipeline. To do so, a novel deep network has been proposed that takes two inputs (the grayscale image and the respective encoded text description) and tries to predict the relevant color gamut. As the respective textual descriptions contain color information of the objects present in the scene, the text encoding helps to improve the overall quality of the predicted colors. The proposed model has been evaluated using different metrics like SSIM, PSNR, LPISPS and achieved scores of 0.917, 23.27,0.223, respectively. These quantitative metrics have shown that the proposed method outperforms the SOTA techniques in most of the cases.

✉ Subhankar Ghosh
subhankar.ghosh@student.uts.edu.au

Prasun Roy
prasun.roy@student.uts.edu.au

Saumik Bhattacharya
saumik@ece.iitkgp.ac.in

Umapada Pal
umapada@isical.ac.in

Michael Blumenstein
Michael.Blumenstein@uts.edu.au

1 Faculty of Engineering and IT, University of Technology Sydney, Ultimo, NSW, Australia

2 E&ECE Department, Indian Institute of Technology Kharagpur, Kharagpur, India

3 CVPR Unit, Indian Statistical Institute, Kolkata, India

Published online: 11 October 2023

**Keywords** Image colorization · Text-guided generation · GAN

## 1 Introduction

Old legacy movies and historical videos are in black and white format. When the video was captured, there was no suitable technology to preserve color information. Black and white or grayscale images can be restored by new real-life colorization, which gives life to old pictures and videos. The main aim of colorization is to add color to a black and white image or grayscale image such that the newly generated image is visually appealing and meaningful. In recent years, based on generative adversarial networks (GANs) [13], a variety of colorization techniques have been proposed, and the state-of-the art performance has been reported on current databases [8, 11, 31]. These colorization techniques differ in many aspects, such as network architecture, different types of loss functions, learning strategies, etc. However, the existing colorization [15, 21, 27, 33, 34] processes mostly follow unconditional generation where the colors are predicted only from the grayscale input image. This might lead to ambiguous results as the prediction of color from a grayscale information is inherently ill-posed. To increase the fidelity in the colorization pipeline, a text-guided colorization pipeline is proposed where some color descriptions about the objects present in the grayscale image can be provided as auxiliary conditions to achieve more robust colorized results (Fig. 1).

The major contributions of our work are as follows.

- A novel GAN pipeline is proposed that exploits textual descriptions as an auxiliary condition.
- We extensively evaluate our framework using qualitative and quantitative measures. In comparison with the state-of-the-art (SOTA) algorithms, it is found that the proposed method generates results with better perceptual quality.
- To the best of our knowledge, this is the first attempt to integrate textual information into an end-to-end colorization pipeline to improve the quality of generation. The textual color description acts as additional conditioning to increase the fidelity in the final colorized output.

It is important to note that the SOTA text-based colorization method [6] is not an end-to-end model. It first tries to estimate a color palette from the textual description and then attempts to colorize the input grayscale image. The proposed method is an end-to-end model that completely circumvents the necessity of any intermediate color palette estimation.

The rest of the paper is organized as follows. Section 2 introduces the SOTA colorization techniques. In Section 3, the proposed colorization framework is discussed in detail. Section 4 presents the experimental settings that are used to train and evaluate the pipeline. We present our results and compare our proposed framework with the SOTA algorithms using qualitative and quantitative metrics in Section 5. Finally, in Section 6, the paper is concluded by pointing out the overall findings of the proposed work, its limitations and future prospects.

## 2 Related work

Image colorization methods have been the primary focus of significant research over the last two decades. Most of these methods were influenced by conventional machine

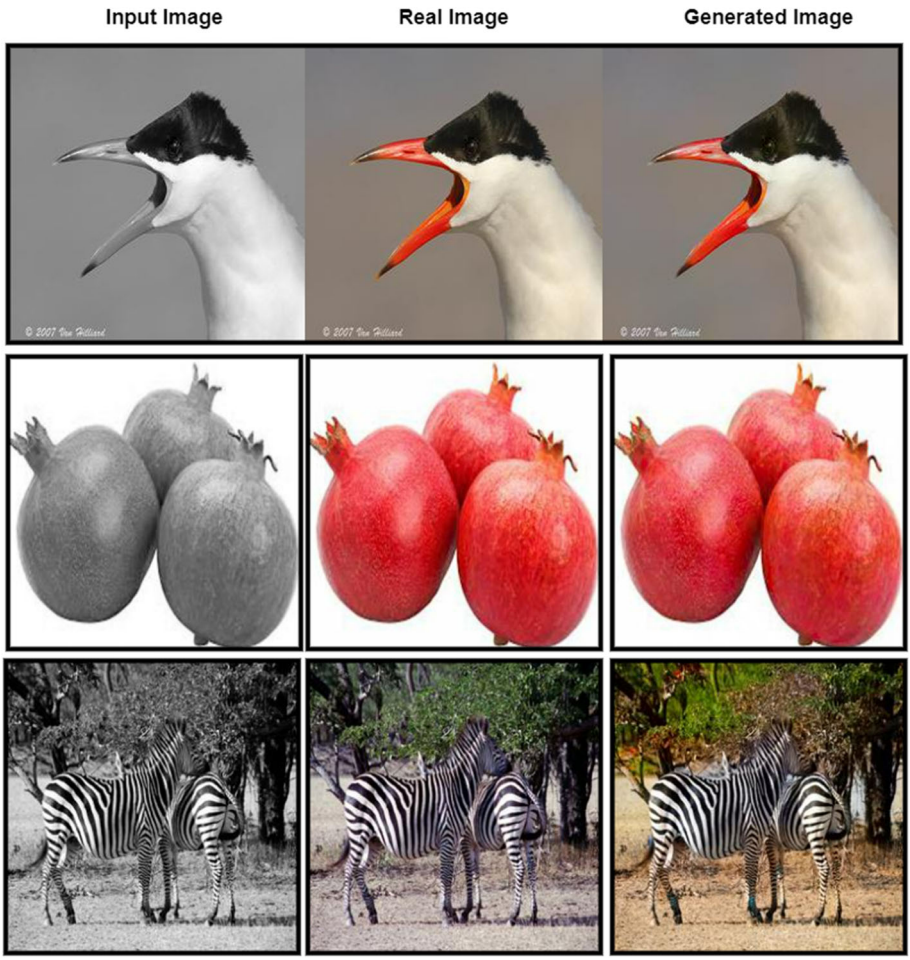


**Fig. 1** Images generated by the proposed algorithm: the first column indicates the input grayscale images; the second column shows the ground truth color images and the third column illustrates the respective colorized outputs of the proposed model. [Best viewed with 300% zoom in the digital version]

learning approaches [7, 14, 19]. In the last few years, the trend has shifted to deep learning (DL)-based approaches due to the success of DL-based approaches in different fields [5, 24, 26, 28]. Recently, DL-enabled automatic image colorization systems have shown an impressive performance in the colorization task [6, 9, 10, 19, 28, 35]. For the attention based mechanism [1–4] the authors are used in literature for crowd management work.

Deep colorization [10] was the first network to incorporate DL for image colorization. In training, five Fully Connected layers are used followed by ReLU activation with the least-squares error as a loss function. In this model, the first layer neurons depend on the features extracted from the gray scale patch. The output layers have only two neurons, i.e., the U and V channel. During testing, grayscale image features are extracted from three levels, i.e., low-level, mid-level, and high-level. The sequential gray values, DAISY feature [26] and

semantic labeling are extracted at the low level, mid-level and high level, respectively to complete the task.

Deep depth colorization [9] used a pre-trained ImageNet [11] network for colorizing the image using their depth information of RGB. First, the network is designed for object recognition by learning a mapping from the depths to the RGB channel. Pre-trained weights are kept frozen in the network, and this pre-trained network is merely used as a feature extractor.

Wang et al. [28] proposed SAR-GAN to colorize Synthetic Aperture Radar images. The cascaded generative adversarial network is used as the underlying architecture. SAR-GAN was developed with two subnets; one is the speckling subnet, and another is the colorization subnet. The speckling subnet generates the noise-free SAR image, and the colorization subnet further processes it to colorize the images. The speckling sub-network consists of 8 convolution layers with BatchNorms and element-wise division in the residual network. The colorization subnet utilizes an encoder-decoder architecture with 8 convolution layers and skips connection. The Adam [17] optimizer is used for training the entire network. The SAR-GAN utilizes hybrid loss with l1 loss and adversarial loss.

The text2color [6] model consists of two conditional adversarial networks: the Text to Palette Generation network and the Palette-based colorization network. The Text to Palette Generation network is trained using the palette and text dataset. The text to Palette Generation networks generator learns the color palette from the text and identifies the fake and real color palette. Huber loss is used as a loss function in this network. The palette-based colorization network is designed using the U-NET architecture where the color palette is used as a conditional input to the generator architecture. The authors had designed the discriminator using a series of convo2d and LeakyRelu [22] modules, followed by a fully connected layer to classify the colorized image as real or fake. In this method, the number of color palettes is 4 to 6. The generated color image entirely depends on the number of the color palette.

In Colorful image colorization [35], authors proposed the 1st CNN architecture for the colorization. In this method, the authors used a cross Chanel encoder for colorization. The colorization method cannot color each object with the appropriate color.

In colorization with optimization [19], authors used some color scribbles to colorized the image. The author proposed a quadratic cost function and obtained an optimization problem that can be solved efficiently using standard techniques. If the number of colors in the image is huge, then the colorization technique can not maintain the color properly.

Towards Vivid and Diverse Image Colorization with Generative Color Prior [32], authors first used a pre-trained GAN for the feature matching. After that authors generated the various color by changing the latent space for the next GAN network. If the pre-train GAN produced miss leading feature, then the colorization network generated the unnatural colorization of the image.

In Colorization transfer [18], author used conditional auto aggressive transformer for generation the image in low resolution. After that, the network consists of two parallel networks, one for course colorization and one for fine colorization.

In the proposed work, a novel end-to-end deep model takes an input grayscale image, and a textual embedding [23] and the model tries to colorize the input image using the textual embedding as side information. The textual embedding is fused with a low dimensional representation of the input image using residual in residual dense block (RRDB) [30] to impose the conditioning.

## 3 Methodology

Image colorization aims to generate a color image from a grayscale image. Typically, deep learning tools use RGB images as ground truth for image generation. In the proposed method, RGB images are converted into the CIE LAB color space, where we need to find only the 'A' and 'B' channels instead of three channels of RGB. The input text is converted, containing the color information of the image, to a word vector using the word2vec. The size of the word vector is 256, and the input size of the image is $256 \times 256$, which is the 'L' channel of the LAB color space. We add the 'L' channel with the AB channel of the image, which the Generator predicts, to reconstruct a fully colorized image. The discriminator signifies the visual authenticity of the image in a patch-based manner.

**Motivation** Though image colorization has a wide range of applications, the majority of the existing methods do not provide any direct control over the colorization process. The scribble-based techniques, that can regulate the final colorized output demand extensive human intervention. Though it has been observed that text-based generative pipelines are extremely user-friendly and give direct control in the generation process, there are only a few attempts to design text-based colorization algorithm due to the inherent complexity of the overall methodology. Most of the existing text-based colorization algorithms try to predict the color palette, and perform the colorization process. To the best of our knowledge, there is no existing end-to-end model available that can exploit the flexibility and richness of text-based generation pipeline which is the prime motivation to propose the model. Regarding the proposed architecture, though largely unexplored, we observe that the RRDB modules perform well in ill-posed inverse problems [12]. Thus, we designed an encoder-decoder-based generative architecture keeping RRDB in the generator.

### 3.1 Generator

The idea of the proposed Generator (Fig. 2) is that the text color information is fused with the grayscale image ($L^i$) at the last downsample step of the network. The input L image is first resized to a fixed size of $256 \times 256$. The overall generator has two pathways- an image pathway, through which the image information flows in the network, and the text pathway, through which the text color information flows as a conditional input. Both pathways finally meets in the Residual in Residual Dense Block (RRDB). For the image path, each resolution level has two convolution layers. The down-sampling follows the last convolution by 2 to move to a new resolution. A $3 \times 3$ kernel size with 64 filters are used in each convolution block. After each convolution, batch-normalization is performed , and each convolution block has ReLU activation. We also process the text vector($S^i$) by two fully connected layers of sizes 256 and 4096. The text features is resized and computed by the last fully connected layer to 1x64x64 and perform an element-wise dot product between the image features and the text features to impose a text-guided conditioning. The text conditioned features are then fed to a Residual in Residual Dense Block (RRBD) before forwarding to the expanding part of the generator. The RRBD block consists of several dense layers with skip connections. The output of each dense block is scaled by $\beta$ before feeding it to the next dense block. Each Dense block consists of a convolutional layer, followed by BN and leaky ReLU activation with the residual connection. As shown in Fig. 3, skip connections are introduced to tackle the problem of a vanishing gradient. The output of the RRBD block is used as input to the convTranspose2d layer with a 64 filter. In the expanding pathway,three up-sampling oparations are used that work in four different resolutions. To increase the feature information

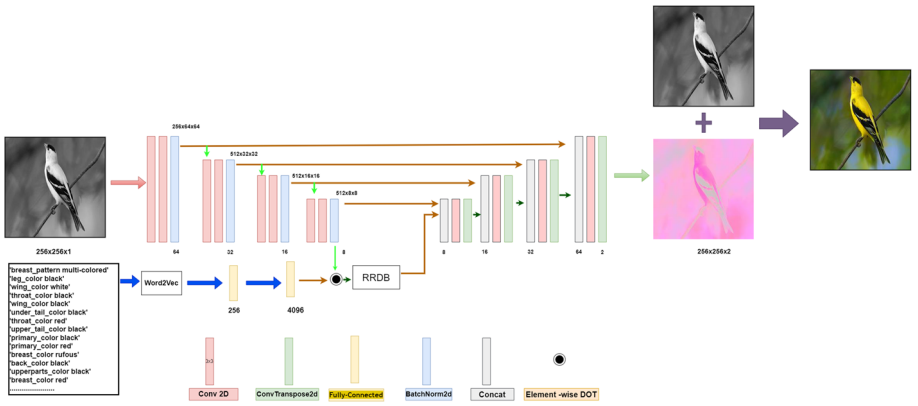


**Fig. 2** The block diagram of the proposed architecture. The network predicts the color components of the image, which is combined with the intensity image to produce the final colorized image. [Best viewed with 300% zoom in the digital version.]

in the expanding path, after each up-sampling layer, the available features are concatenated with the same resolution in the contracting path. The convolution blocks in the expanding path are similar to the convolution blocks at the contracting paths, and The number of filters are decreased by two as we move to the higher resolution. At the highest resolution, two filters are applied with kernel size 1x1 to generate the estimated $AB$ channel of the color image. At the end of the proposed network, The color image is computed by adding the generated $AB$ and the input grayscale image($L^i$). The proposed Generator is illustrated in Fig. 2.

### 3.2 Discriminator

For the colorization task, it is required that the discriminator can detect the local quality of a generated colorized image. Thus, The PatchGAN Discriminator $D$ use to judge the quality of the generated image. The discriminator penalizes the generated structure at the patch level resulting in a high-quality single level generation. The grayscale image are stack ($L^i$)

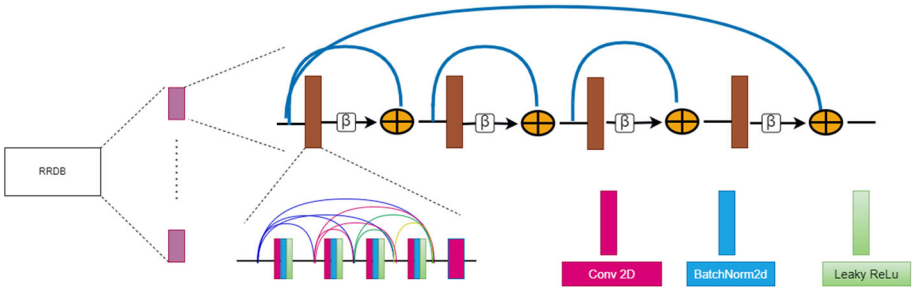


**Fig. 3** The block diagram of the Residual in Residual Dense Block(RRBD) architecture. [Best viewed with 300% zoom in the digital version.]

with either a target image ($T^i$) or with a estimated image ($E^i$) where $T^i$ and $E^i$ are the $AB$ channel of the color image. The ($L^i$, $T^i$) stack is labeled as real and the ($L^i$, $E^i$) stack is labeled as fake. In this way, we enforce discrimination on image transition rather than the image itself. In our model, the Patch discriminator takes a three-channel input dimension 256 × 256. The discriminator has three convolution blocks with 64, 128 and 256 filters, respectively, in each block with filter dimension 4 × 4. In the first two convolution blocks, the filter has stride 2, whereas, for the last two blocks, 1x1 stride is used . Each convolution layer is followed by batch-normalization and leaky-ReLU activation. After the convolution blocks, one filter of kernel size 4x4is applied with stride 1 to compute the final response. The average of the final response is the output of the discriminator.

### 3.3 Training

As mentioned in, the PatchGAN discriminator focuses more on the high frequency information. Thus to keep the fidelity of low frequency information in the colorized image, $L_1$ loss use in the generator G which is calculated as

$$\mathcal{L}_1^G = \|E^i - T^i\|_1 = \|G(L^i, S^i) - T^i\|_1 \tag{1}$$

$$\mathcal{L}_1^G = \sum_{i=1}^{d} \|x_i - y_i\| \tag{2}$$

where $x_i$ and $y_i$ are the $i$-th elements of $d$-dimensional vectors $\mathbf{x}$ and $\mathbf{y}$, respectively. As the generator trained in an adversarial manner,the adversarial or the GAN loss of the generator and the discriminator define as:

$$\mathcal{L}_{GAN}^G = \mathcal{L}_{BCE}(D(L^i, G(L^i, S^i)), 1) \tag{3}$$

$$\mathcal{L}_{GAN}^D = \mathcal{L}_{BCE}(D(L^i, T^i), 1) + \mathcal{L}_{BCE}(D(L^i, G(L^i, S^i)), 0)$$

$$\mathcal{L}_{BCE} = (y \log(p) + (1 - y) \log(1 - p)) \tag{4}$$

where $L_{GAN}^G$ and $L_{GAN}^D$ denote adversarial Generator loss adversarial discriminator loss, respectively. To increase the visual quality of the image, perceptual loss is used to train the generator. In $\mathcal{L}_{BCE}$, y is the label and p is predicted probability of the point.

$$\mathcal{L}_{p_\rho}^G = \frac{1}{h_\rho w_\rho c_\rho} \sum_{x=1}^{h_\rho} \sum_{y=1}^{w_\rho} \sum_{z=1}^{c_\rho} \|\phi_\rho(E^i) - \phi_\rho(T^i)\|_1 \tag{5}$$

where $\mathcal{L}_{p_\rho}^G$ is the perceptual loss computed at the $\rho^{\text{th}}$ layer, $\phi_\rho$ is the output from the $\rho^{\text{th}}$ layer of a pretrained VGG19 model, and $h_\rho$, $w_\rho$ and $c_\rho$ are the height, width and the number of channels at that layer, respectively.

The total generator loss $\mathcal{L}^G$ can be defined as

$$\mathcal{L}^G = arg \min_G \max_D \lambda_1 \mathcal{L}_{GAN}^G + \lambda_2 \mathcal{L}_{P_4}^G + \lambda_3 \mathcal{L}_1^G \tag{6}$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{7}$$

## 4 Experimental details

The PyTorch framework is used to build the model, and perform our experiments. The Adam [17] optimizer is used to train both the generator and discriminator up to 350K iterations with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate is $1 \times 10^{-4}$ with a decay of 0. All the leaky-ReLU activations have negative slope coefficients of 0.2. We select $\lambda_1 = 1$, $\lambda_2 = 1$ and $\lambda_3 = 1$.

While training the discriminator $D$, by concatenating $L^i$ with either $T^i$ or $E^i$ and give that as a input. Both $D$ and $G$ are trained iteratively, *i.e.*, we keep $D$ fixed while training G and vice versa. As the training process of GAN is highly stochastic, the network weights is stored at the end of each iteration. At the time of inference, The discriminator network is droped and generate the A,B channels only using the generator network.

| | |
|---|---|
| 'breast_pattern multi-colored' | 'upperparts_color white' |
| 'leg_color black' | 'back_color white' |
| 'wing_color white' | 'belly_color white' |
| 'throat_color black' | 'wing_pattern multi-colored' |
| 'wing_color black' | 'leg_color grey' |
| 'under_tail_color black' | 'underparts_color white' |
| 'throat_color red' | 'crown_color rufous' |
| 'upper_tail_color black' | 'eye_color black' |
| 'primary_color black' | 'forehead_color rufous' |
| 'primary_color red' | 'underparts_color red' |
| 'breast_color rufous' | 'tail_pattern multi-colored' |
| 'back_color black' | 'nape_color red' |
| 'upperparts_color black' | 'forehead_color red' |
| 'breast_color red' | 'breast_color white' |

**Fig. 4** A typical example in our dataset: each sample contains a color image and corresponding color descriptions of the bird. To use the image while training the network, the color image is converted to LAB color space, and use the 'L' image as the input

### 4.1 Datasets

To evaluate the performance of our model, three popular datasets, Caltech-UCSD Birds 200 [31], MS COCO [20] and Natural color Dataset(NCD) [5] is used.

The Birds dataset (Fig. 4) contains 6032 bird images with their color information. The dataset is split into two parts (train, test). The total number of images for training is 5032, and the remaining images are used in the test set.

The total number of images in the NCD set are 730 fruit images. 600 images are used for training and the remaining 130 images for testing. The class label is converted to one single color, like the tomato's class is converted into red and used as color information for training and testing.

From the MS COCO [20] dataset, 39k images are used for training and 6225 images for testing. In COCO stuff [8], the text description of the images are available. In each text description, The sentence(s) related to color information of an object is an auxiliary information to the network. By collecting all such sentences and use it as the final auxiliary information for the respective image.

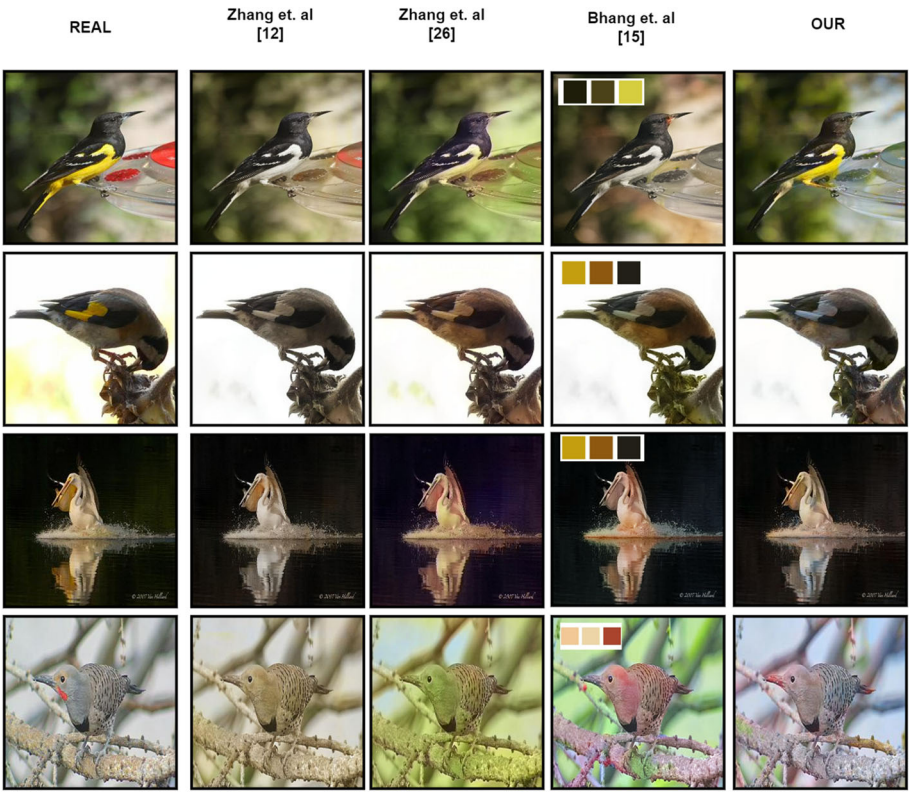


**Fig. 5** Qualitative comparison results: The first column contains ground truth images, the second column, third and fourth columns contain the results generated by the SOTA algorithms, and the last column shows the results generated by the proposed algorithm. [Best viewed with 300% zoom in the digital version.]

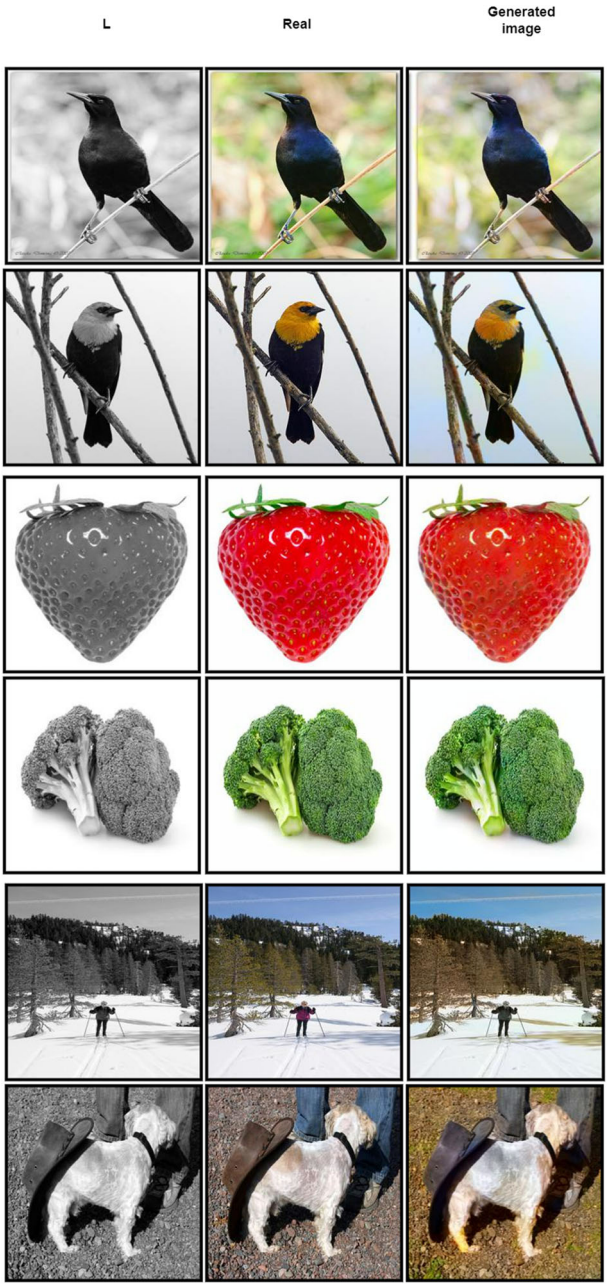


**Fig. 6** Images generated by the proposed algorithm from theCaltech-UCSD Birds 200 [31], NCD [5] and MS COCO stuff [8] Dataset: the first column contains the grayscale images, the second column contains the ground truth images and the third column shows the colorized outputs of the proposed model. [Best viewed with 300% zoom in the digital version.]

**Table 1** Quantitative comparison among different colorization methods – Zhang et al. [35], Zhang et al. [36], Bhang et al. [6] and our method. The bold emphasis indicates the best result in that metric

| Method | SSIM ↑ | PSNR ↑ | LPIPS (vgg) ↓ | LPIPS (sqz)↓ |
|---|---|---|---|---|
| Zhang et al. [35] | 0.903 | 22.94 | 0.253 | 0.143 |
| Zhang et al. [36] | 0.892 | 22.15 | 0.231 | 0.129 |
| Bhang et al. [6] | 0.912 | 22.99 | 0.228 | **0.127** |
| Our | **0.917** | **23.27** | **0.223** | 0.133 |

## 5 Experimental results

To understand the overall performance of the proposed framework by performing an extensive set of experiments to evaluate the quality of the final colorized images. In Fig. 5, we compare our algorithm with [35, 36] and [6]. As shown in the figure, the proposed algorithm colorized the grayscale images with higher fidelity. Although the existing methods have colorized the grayscale images successfully, however, the colors are often significantly different to the actual ground truth. The colorized images produced by the SOTA algorithms are also less colorful. As the proposed method utilizes the textual description


**Fig. 7** Validation of the importance of the textual encoding: first column contains the grayscale images, second column contains the ground truth images, the third and fourth columns show the results generated without and with the textual encoding, respectively. [Best viewed with 300% zoom in the digital version.]

as auxiliary information, our algorithm generates more realistic and colorful images from the respective grayscale input images. Evaluate the network's performance by generating the color images from three public databases. Figure 6 shows image samples from the UCSD Bird dataset [31]and MS COCO [8] dataset and the Natural Color Dataset [5], respectively.

To further validate the effectiveness of the proposed model, To evaluate the quality of the generated images using quantitative metrics as well. Average PSNR, SSIM [29], LPIPS(vgg) [25, 37] and LPIPS(sqz) [16] measures is used to compare the similarity of the generated images with the ground truth. As shown in Table 1, the proposed algorithm outperforms the SOTA algorithms in SSIM, PSNR, LPIPS(vgg) measures.

### 5.1 Ablation study

To further validate the textual description's importance by training a new model without using the textual information. As shown in Fig. 7, without the textual conditioning, the proposed pipeline fails to colorize the grayscale images properly. In Fig. 8, represents that the textual description can be used for the recolorization task. In Fig. 8 (a), grayscale image was colorized with the actual textual description of the ground truth. In Fig. 8 (b), by keeping the grayscale image unchanged and have used the textual description of a different image. It is observed that the proposed framework is able to follow textual conditioning and can produce significantly different colorized outputs from the same grayscale image based on the textual encoding. In the ablation study, RRBD networks ablation is by changing the nuber of RRBD. In Fig. 9 shows the generator images with 1,32 and 64 RRBDs. The result of the forth column (generated by 64 RRBDs)is the best result of this ablation study. To validate the understating some quantitative result is generated in the Table 2.

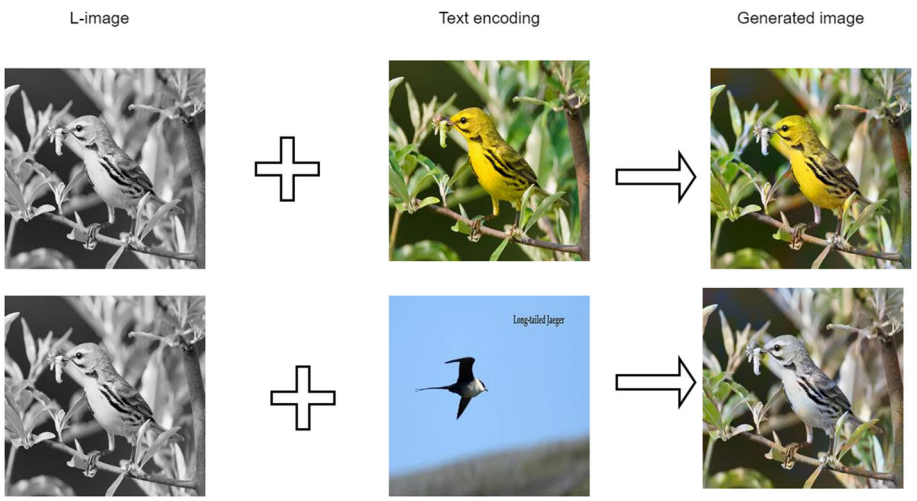


**Fig. 8** Recolorization: The first column shows the grayscale images, the second column shows the images whose textual descriptions are used as conditioning. The third column shows the final colorized images. [Best viewed with 300% zoom in the digital version.]

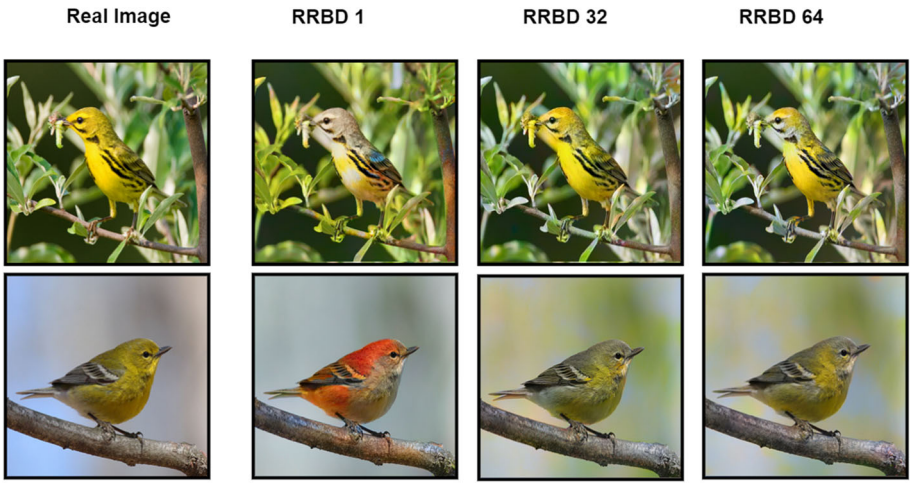


**Fig. 9** The first column shows the Real images, the second column shows the generator images with 1 RRBD. The third column shows the generator images with 32 RRBDs. The fourth column shows the generator images with 64 RRBDs [Best viewed with 300% zoom in the digital version.]

## 6 Conclusions

In this paper, we proposed a novel image colorization algorithm that utilizes textual encoding as auxiliary conditioning in the color generation process.It is found that the proposed framework exhibits higher color fidelity compared to the state-of-the-art algorithms. We have also demonstrated that the proposed framework can also be used for recolorization purposes by modulating textual conditioning. It is important to note that we have considered only textual conditioning of foreground objects in this work. In the given setting, though the proposed algorithm outperforms the SOTA methods, as the textual descriptions mostly depict the foreground objects ignoring the backgrounds; our method exhibits less fidelity for the background colors. This problem can obviously be resolved by adding additional color descriptions for the background. We also observed that as the textual descriptions define the colors of the objects coarsely, to fill the gaps, the proposed method generates certain colors which are not there in the respective ground truths. Thus, in certain cases, our method produced less colorful backgrounds (Fig. 10), which establishes the necessity of a more exhaustive textual description for the grayscale images in the future. Efforts should also be made to design a more robust colorization process for the background for which textual descriptions are not rich.

**Table 2** Quantitative comparison among different number of RRBDs in ablation study. The bold emphasis indicates the best result in that metric

| Method | SSIM ↑ | PSNR ↑ | LPIPS (vgg) ↓ | LPIPS (sqz)↓ |
|---|---|---|---|---|
| RRBD 1 | 0.910 | 22.99 | **0.222** | 0.138 |
| RRBD 32 | 0.912 | 23.13 | 0.228 | 0.159 |
| RRBD 64 | **0.917** | **23.27** | 0.223 | **0.133** |

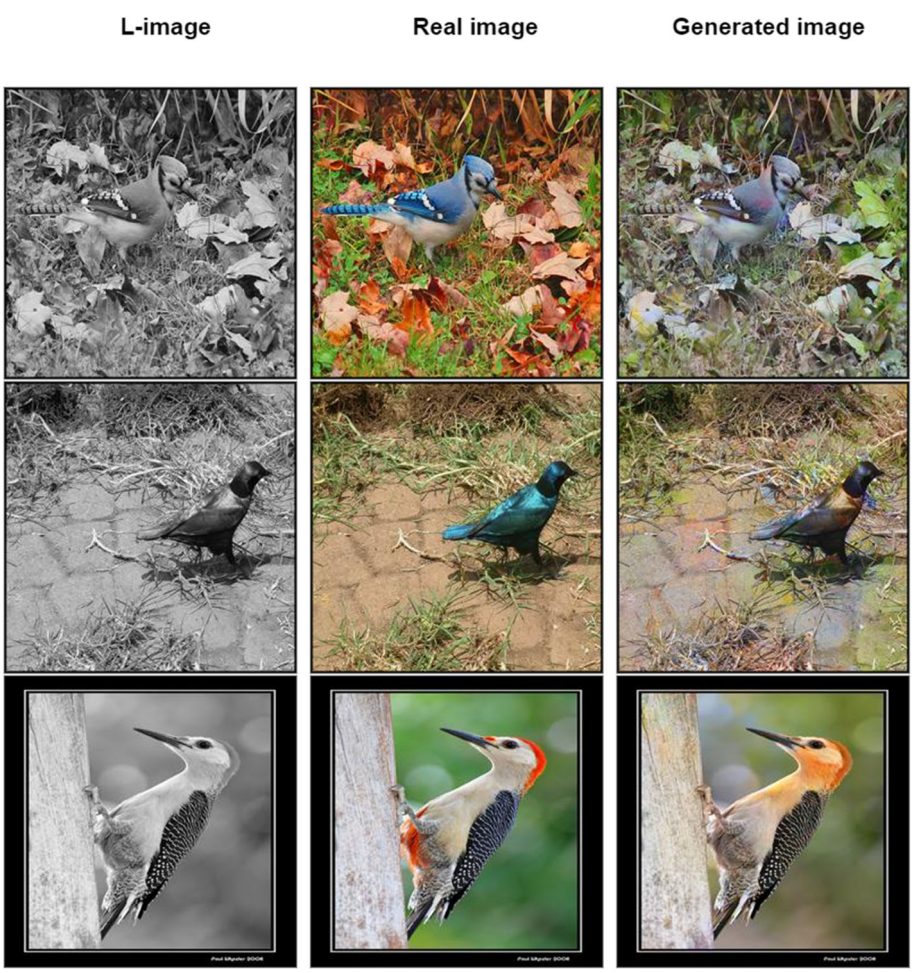


**Fig. 10** Some of the failure cases. [Best viewed with 300% zoom in the digital version.]

**Funding** Open Access funding enabled and organized by CAUL and its Member Institutions

**Data Availability** Data will be made available on reasonable request.

## Declarations

Financial interests: The authors declare they have no financial interests.

**Conflict of Interests** The authors declare that they have no conflict of interest

## References


1. Ali A, Zhu Y, Chen Q, Yu J, Cai H (2019) Leveraging spatio-temporal patterns for predicting citywide traffic crowd flows using deep hybrid neural networks. In: 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS). IEEE, pp 125–132
2. Ali A, Zhu Y, Zakarya M (2021) A data aggregation based approach to exploit dynamic spatio-temporal correlations for citywide crowd flows prediction in fog computing. Multimed Tools Applic 80:31401–31433
3. Ali A, Zhu Y, Zakarya M (2021) Exploiting dynamic spatio-temporal correlations for city-wide traffic flow prediction using attention based neural networks. Inf Sci 577:852–870. https://doi.org/10.1016/j.ins.2021.08.042, https://www.sciencedirect.com/science/article/pii/S0020025521008483
4. Ali A, Zhu Y, Zakarya M (2022) Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction. Neural Netw 145:233–247. https://doi.org/10.1016/j.neunet.2021.10.021, https://www.sciencedirect.com/science/article/pii/S0893608021004123
5. Anwar S, Tahir M, Li C, Mian A, Khan FS, Muzaffar AW (2020) Image colorization: a survey and dataset. arXiv:2008.10774
6. Bahng H, Yoo S, Cho W, Park DK, Wu Z, Ma X, Choo J (2018) Coloring with words: guiding image colorization through text-based palette generation. In: ECCV
7. Bastos R, Wynn WC, Lastra A (2013) Run-time glossy surface self-transfer processing
8. Caesar H, Uijlings JRR, Ferrari V (2018) Coco-stuff: thing and stuff classes in context. In: 2018 IEEE/CVF Conference on computer vision and pattern recognition, pp 1209–1218
9. Carlucci FM, Russo P, Caputo B (2018) $(de)^2co$: deep depth colorization. IEEE Robotics and Automation Letters
10. Cheng Z, Yang Q, Sheng B (2015) Deep colorization. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp 415–423
11. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on computer vision and pattern recognition, pp 248–255
12. Deora P, Vasudeva B, Bhattacharya S, Pradhan PM (2020) Structure preserving compressive sensing mri reconstruction using generative adversarial networks. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp 2211–2219
13. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: The Conference on Neural Information Processing Systems (NIPS)
14. Huang Y-C, Tung Y-S, Chen J-C, Wang S-W, Wu J-L (2005) An adaptive edge detection based colorization algorithm and its applications. In: MULTIMEDIA '05
15. Huang S, Jin X, Jiang Q, Liu L (2022) Deep learning for image colorization: current and future prospects. Eng Appl Artif Intell 114:105006
16. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K (2016) SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5mb model size. arXiv:1602.07360
17. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
18. Kumar M, Weissenborn D, Kalchbrenner N (2021) Colorization transformer. arXiv:abs/2102.04432
19. Levin A, Lischinski D, Weiss Y (2004) Colorization using optimization. In: SIGGRAPH 2004
20. Lin T-Y, Maire M, Belongie SJ, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: ECCV
21. Luo F, Li Y, Zeng G, Peng P, Wang G, Li Y (2022) Thermal infrared image colorization for nighttime driving scenes with top-down guided attention. IEEE Trans Intell Transp Syst 23:15808–15823
22. Maas AL (2013) Rectifier nonlinearities improve neural network acoustic models
23. Mikolov T, Chen K, Corrado GS, Dean J (2013) Efficient estimation of word representations in vector space. In: ICLR
24. Perazzi F, Pont-Tuset J, McWilliams B, Gool LV, Gross MH, Sorkine-Hornung A (2016) A benchmark dataset and evaluation methodology for video object segmentation. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 724–732
25. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: The International Conference on Learning Representations (ICLR)
26. Tola E, Lepetit V, Fua PV (2008) A fast local descriptor for dense matching. In: 2008 IEEE Conference on computer vision and pattern recognition, pp 1–8
27. Treneska S, Zdravevski E, Pires I, Lameski P, Gievska S (2022) Gan-based image colorization for self-supervised visual feature learning. Sensors (Basel, Switzerland), 22

28. Wang P, Patel VM (2018) Generating high quality visible images from sar images using cnns. In: 2018 IEEE Radar Conference (RadarConf18), pp 0570–0575
29. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. In: IEEE Transactions on Image Processing (TIP)
30. Wang X, Yu K, Wu S, Gu J, Liu Y, Dong C, Qiao Y, Loy CC (2018) Esrgan: enhanced super-resolution generative adversarial networks. In: The European Conference on Computer Vision Workshops (ECCVW)
31. Welinder P, Branson S, Mita T, Wah C, Schroff F, Belongie S, Perona P (2010) Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology
32. Wu Y, Wang X, Li Y, Zhang H, Zhao X, Shan Y (2021) Towards vivid and diverse image colorization with generative color prior. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp 14357–14366
33. Wu DNBH, Gan J, Zhou J, Wang J, Gao W (2022) Fine-grained semantic ethnic costume high-resolution image colorization with conditional gan. Int J Intell Syst 37:2952–2968
34. Xiao Y, Jiang A, Liu C, Wang M (2022) Semantic-aware automatic image colorization via unpaired cycle-consistent self-supervised network. Int J Intell Syst 37:1222–1238
35. Zhang R, Isola P, Efros AA (2016) Colorful image colorization. In: ECCV
36. Zhang R, Zhu J-Y, Isola P, Geng X, Lin AS, Yu T, Efros AA (2017) Real-time user-guided image colorization with learned deep priors. ACM Transactions on Graphics (TOG) 36:1–11
37. Zhang R, Isola P, Efros AA, Shechtman E, Wang O (2018) The unreasonable effectiveness of deep features as a perceptual metric. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)