IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 January 2023, accepted 13 March 2023, date of publication 3 April 2023, date of current version 2 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3263948

RESEARCH ARTICLE

A Novel Unbiased Deep Learning Approach (DL-Net) in Feature Space for Converting Gray to Color Image

MRITYUNJOY GAIN AND RAMESWAR DEBNATH, (Member, IEEE)
Computer Science and Engineering Discipline, Khulna University, Khulna 9208, Bangladesh

Corresponding author: Rameswar Debnath (rdebnath@cseku.ac.bd)

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery grants, and by an NSERC Collaborative Research and Training Experience (CREATE) grant on Software Analytics Research (SOAR).

ABSTRACT Gray to Color conversion causes difficulties because of the nature of its intrinsic multi-modality. Despite recent significant advancements in this domain by numerous learning-based approaches, there still have two drawbacks: 1) implausible color assignment and 2) contextual ambiguity. Recently deep learning models are being used for colorization as they outperform others. In a training image, desaturated color components are greater than saturated color components due to the larger background areas (clouds, pavement, dirt, walls, etc.) compared to the focused objects. This imbalanced feature representation biases the learning model in favor of major features. However, small regions with specific colors are the region of interest. To solve this problem, we proposed the Deep Localization Network (DL-Net) by modifying the mean squared error backpropagation algorithm. We compute chromatic component-based Local Losses (LLs) which are the primary component of the proposed DL-Net. The LL employs priority on rare semantic components of the original image features. It works to improve diverse-range dependency modeling in an effort to reduce contextual ambiguity and color leakage that promotes the production of more plausible coloring. With a number of current methodologies, we contrast our proposed approach. The experimental findings demonstrate that our proposed method produces good colorization of images and outperforms other methods in terms of SSIM, MSE, and PSNR quality criteria.

INDEX TERMS Imbalance feature, mean square error, local loss function, deep localized network, colorization.

I. INTRODUCTION

Color visuals are more perceptible to human sight. Instead of viewing grayscale images, people experience higher satisfaction and pleasure to view colored ones. Images from antiquity, medicine, and astronomy are typically drab and unable to depict their accurate interpretations and expressions. It is vital to colorize the image in order to gain a greater understanding of its semantics.

For image colorization, researchers used a variety of methods. User-guided colorization [1], [2], [3] and data-driven colorization [4], [5], [6], [7], [8], [9], [10] are the two primary categories into which existing techniques for coloring grayscale photos can be divided. The amount of human interaction needed for traditional user-guided colorization to correctly color the image is too tremendous [4], [5], [9]. The user-guided approach has lost popularity in favor of data-driven strategies [4], [5], and [9] since they are simpler and involve less human work. A reference color image is needed in a data-driven approach in order to add color to the grayscale image. Using the reference image as a guide, the user manually selects color values. Image colorization methods based on deep network learning are also growing in popularity today.

Convolutional Neural Network (CNN) is a deep learning network. The CNN can efficiently extract different picture

The associate editor coordinating the review of this manuscript and approving it for publication was Shunfeng Cheng.

properties and categorize them for colorization [11]. Convolution Neural Networks (CNNs) are composed of two distinct alternating layer types: convolutional and sub-sampling layers [11]. The first convolutional layer in a CNN extracts primitive features of network traffic while the subsequent convolutional layers deduce more sophisticated features. The activation unit in a CNN represents the results of the convolution operation of the input data with a kernel. The convolution layer is followed by a max-pooling layer for the dimensionality reduction of data. Finally, the dense layer classifies the output classes combining all complex features identified by convolutional layers.

Through several underlying network layers, Deep Neural Networks (DNNs) extract representative features and hidden structural knowledge from data by training. To optimize the model's parameters, the feedback is generated by the loss function in every epoch during training. The loss function is the measurement of the disparity between the predicted output and the ground-truth value, i.e., error. In each epoch, networks update the weights of the model in proportion to the error. The backpropagation algorithm gives the same importance to the misclassification errors of data instances from each class. The training process adapts the classifier in favor of the majority class for imbalance class distributions [12]. In imbalance distribution, harder instances from classes with fewer observations produce lower class probabilities by the models. However, correct instances have greater SoftMax probabilities than those misclassified and out-of-distribution instances [13]. Thus, the training process with an imbalance class distribution does not hinder the model performance of clear class separation tasks but it affects the instances that are inherently more difficult to classify.

Since the training process with an imbalance class distribution typically underestimates the class probability estimates of minority class instances, thus learning models misclassify the minority class observations for hard instances [14]. Therefore, the predicted class probabilities are unreliable in imbalanced class distributions.

In many real-world problems like microarray, medical images, color visualization, sequencing, etc., feature distributions have greater importance than class distributions. In these problems, the same scenarios arise when feature distributions are imbalanced. Therefore, the training process is biased to the larger feature subsets in imbalanced feature distribution, and characteristics of fewer feature subsets disappear in the resultant models. Handling feature imbalance is of great importance because the fewer feature subsets are the feature of interest concerning the learning task.

There are many existing methods dealing with the colorization problem. But in the colorization process, we see that desaturated color components are far more prevalent than saturated color components in training images, which dramatically influences the training process and causes saturated color components to skew toward desaturated components in the targeted image. Sometimes the colors of smaller items blend in with the background in the targeted image.

The class imbalance is usually handled by re-sampling the dataset to make it class balanced or by rescaling the data samples or using the weighted function for imposing higher weight on the minority class [12]. In the training process, the features of a sample determine the gradient directions on the loss function. The feature imbalance problem is usually seen in the computer vision domain. An image is composed of blocks of different colors where the blocks are of different sizes. However, all blocks have the same importance. The features of a sample determine the input dimension of a learning model. For colorization models, the output dimension is also the same as the input dimension. Feature rescaling or reproducing is not possible. The feature imbalance issues can be handled by adjusting color block loss (local loss) gradients.

In this context, we propose a novel learning algorithm where the feature set of instances is first clustered into several feature subsets (similar groupings). Instead of calculating a global loss function, loss functions are calculated for every subset of features. The backpropagation algorithm is developed based on local loss functions. In contrast to conventional models which propagate global loss as feedback into hinder layers where each mismatch gets the same importance, the proposed algorithm propagates losses computed by a subset of features as feedback into the respective nodes of the hinder layers in which that subset of features is extracted. We can assign weights with each loss function according to the priority of the clustered feature subset. To focus more on any feature cluster, higher weights can be assigned to training. This learning process can impose local features to extract more realistic features from feature sequences. The proposed algorithm can give an unbiased model for highly imbalanced features. Experimental results show that the proposed method outperforms existing methods in terms of SSIM, MSE, and PSNR, and produces improved colorized images.

The rest of the paper is divided into four sections. Section II contains the literature review. Section III discusses the proposed methodology in detail. Section IV shows the experimental results. Finally, the conclusion is written in Section V.

II. LITERATURE REVIEW

A. BACKGROUND

In this subsection, we will focus on the idea of how colorization tasks work. In the early stage, image colorization was performed by user interaction. Early methods mainly depended on user-complex doodling (such as dots or strokes) to direct the coloring process due to the multimodal difficulty of image colorization. User-interacted colorization is roughly divided into scribbles and examples based. It is typically unrealistic to assume that one or more reference photos will have enough color information to provide acceptable colorization results. In recent times, colorization is performed by data-driven methods. A vast number of source photos can be used to train in data-driven methods because those work without user interaction. In terms of colorization, this entails automatically discovering hues that naturally go with actual items. By expanding the network layers and increasing the training samples, the methods produce better outcomes. These methods are briefly described below.

1) SCRIBBLE-BASED COLORIZATION

One of the oldest methods of colorization is the scribble-based method. It is a user-guided colorization method. It interpolates colors to the crucial section or part of the image depending on user-specific scribbles. The authors in [2] provided a technique based on optimization for spreading the user-specific color scribbles to every pixel in the image. The nearby pixels with the same intensity levels had been colored with the same color from the user's color scribbles, giving them a gray appearance. For this work, they employed the quadratic cost function. Yatziv and Sapiro [3] suggested a method for determining pixel color by combining the color information from various scribbles. By preventing the color from flowing over the boundaries of the objects, Huang et al. [1] improved the method in [3].

2) EXAMPLE-BASED COLORIZATION

Example-based colorization transfers color components using a reference image that is connected to the input image. It is also a user-guided colorization method. Welsh et. al. [7] offered a semi-automatic process for converting a ground-truth reference image to a grayscale image. This technique compares the reference image with the brightness value and texture information of the gray image. The user analyzes the brightness values in the vicinity of each reference picture pixel and assigns the weight to grayscale image pixels that correspond. According to Sousa et. al. [6], each pixel in a grayscale image is given a color based on how intense it is in a reference color image with a similar subject matter. On the basis of superpixels, Gupta et. al. [8] derived features by matching from both the input image and reference image. These techniques have considerable difficulties despite producing amazing outcomes. These techniques are effective if the user can provide a suitable reference image from the internet or the natural world that contains the desired colors, but this is a laborious operation.

3) LEARNING-BASED COLORIZATION

Learning-based colorization is data-driven and automatic. The automatic means the model colorizes images without any user interaction after feeding the input image to the model. For this reason, the automatic colorization method based on learning has become more popular as a solution to the issue of user guidance dependency. These methods are familiar for the mapping between the color image and the grayscale input in a sizable dataset. A network is comprised of sums of nonlinearly transformed linear models and it is trained to approximate the non-linear functions between the input and output. First, it derives complex features by linear combinations of the inputs, and next, it derives the model's target function as a non-linear function from the derived features. Different model architectures are used for image colorization.

B. RELATED WORK

Dahl et. al. [15] proposed an automatic method to produce full-color channels for gray images using 4 pre-trained layers from VGG16 [16]. They used hypercolumns [17] with CNN for this task. They constructed a color output image by forwarding the input image to the VGG network, extracting features, and finally concatenating them. Hwang and Zhao [18] designed and built an automatic grayscale image colorization method based on the baseline regression model. Baldassarre et. al. [19] proposed a method that combines Deep CNN with Inception-ResNet-v2 [20]. This Inception-ResNet-v2 model is pre-trained and is used for high-level feature extraction. They trained from scratch in their Deep CNN model. An et. al. [4] proposed a model with the help of a VGG-16 CNN model, which relied on classification. Cross entropy loss and color rebalancing were used in this approach. Qin et. al. [21] used ResNet [22] for their colorization method. Their method combines classified information and image features.

Zhang et. al. [23] proposed an automatic colorization using CNN. Zhang's method is a classification model where the whole color range is divided into groups (classes). They used class rebalancing during training time to increase the variety of colors on the output image. Zhang et. al. [24] fused low-level cues with high-level semantic information. Iizuka et. al. [5] developed an end-to-end technique that jointly learns global and local image features. This method exploits classification labels (category of images) for increasing model performance. Qin et. al. [25] used a dense network for extracting texture and detailed features from the image as it has a small amount of information loss than that of other CNN architectures. Su et. al. [26] proposed an instance colorization network, where they extracted both object-level and full-image features for colorization. Dai et. al. [27] proposed an encoder-decoder model consisting of the local pyramid attention (LPA) module and the spatial semantic modulation (SSM) module. They used the LPA module for producing a range of scales of local features and spatial semantic modulation for plausible color generation. Xu and Ding [28] proposed a model of automatic image colorization which is based on semantic segmentation technology. They utilized a semantic segmentation network to quicken the convergence edges of the image. Wu et. al. [29] proposed a generative adversarial network-based model which used fine-grained semantic information for image colorization. They built an ethnic costume dataset covering four Chinese minority groups and applied a coloring model based on Pix2PixHD.

Hesham et. al. [30] proposed a colorization model using a scaled-YOLOv4 detector. They detected different objects from multi-object images and applied colorization to them. Guo et. al. [31] proposed a GAN-based bilateral Res-U-net model for image colorization. Liu et. al. [32] proposed a super-resolution network with color awareness that combines the concepts of picture colorization and super-resolution to enhance panchromatic pictures' spectral and spatial resolution. Kumar et. al. [33] proposed a pairing model of the Siamese network and convolutional neural network for image colorization. By integrating the networks, they looked into the possibility of improving colorization. Özbulak et. al. [34] used Capsule Network (CapsNet) for image colorization. The generative and segmentation properties of the original CapsNet are proposed for the image classification problem, but they altered the network and used it to colorize the images. Kong et al. [35] introduced an adversarial edge-aware model that integrates multitask output with semantic segmentation for image colorization. They employed a generator that learns colorization under chromatic ground truth values and extracts deep semantic characteristics from a given grayscale. For training, they also incorporated adversarial loss, segmentation loss, and semantic difference loss in terms of human vision. Wu et. al. [36] proposed a GAN-based model. For retrieving bright colors, they made use of the enhanced and varied color priors contained in pre-trained Generative Adversarial Networks. They used a GAN encoder to first find matching features that are similar to exemplars, and after that modulate these features into the colorization process. Nguyen-Quynh et al. [37] suggested an encoder-decoder image colorization model by exploiting both global and local priors. Bahng et. al. [38] proposed a model consisting of two conditional generative adversarial networks. The first network turns text into a palette, while the second network colorizes images using those palettes. Liang et. al. [39] proposed a colorization network based on the cycle generative adversarial network (CycleGAN) model, which combines a perceptual loss function and a total variation (TV) loss function to secure colorized medical images and to improve the quality of synthesized images. Using generative adversarial networks (GANs), Treneska et. al. [40] suggested a self-supervised technique that produces color images. They also employed transfer learning for visual understanding. Using deep convolution GAN, Wu et. al. [41] provided a new technique for coloring remote sensing images. The GAN generator collects detailed picture characteristics. The generator and discriminator successfully optimize one another to produce color images. Sugawara et. al. [42] used graph signal processing in their colorization method. Two separate networks are used; the first is a global graph that connects the key pixels on an image, and the second is a local graph that connects the global graph to each individual pixel. A color image is retrieved using the hierarchical combination of these two graphs. Afifi et. al. [43] presented the HistoGAN technique for controlling the color of GAN-generated images. They proposed a histogram feature that specifies the colors of GAN-generated images by altering the recently created StyleGAN architecture [44]. Gain et. al. [45] proposed an encoder-decoder CNN architecture with filtering-based rebalancing techniques to colorize images. Three models are developed based on the indoor, outdoor, or human image type. A user will choose a mode according to the image type. Larsson et. al. [46] proposed an automatic deep neural network model which is trained with semantic features.

Though these methods can solve some problems of reliable colorization, still object color matching and proper color saturation are unsolved and ongoing research issues.

III. PROPOSED METHODOLOGY

In this paper, the proposed network is built on encoder-decoder architecture. CIE L*a*b* color space [47] is a useful tool for color manipulation. The L*a*b* color model decouples the intensity components (represented by lightness L∗) from the color-carrying information (represented by a* for red-green and b* for yellow-blue). The lightness information can be separated from color information in L*a*b* space highly compared to any other color model. The lightness information contains the main image features. Lightness information can be mapped into the gray level (intensity) and vice versa [47]. The lightness channel L* is defined as model input $W \in R^{H\times W\times 1}$ and the other two a*b* color channels as the model output $Y\in R^{H\times W\times 2}$. $X\in R^{H\times W\times 2}$ is the ground truth color channels. Where H and W are height and weight respectively. We assume the task is to learn the mapping function f: $W \in Y$. The predicted a*b* channels Y is combined with the input L* channel X to estimate the color image $Z= (W, Y)$. The model's mean square loss function is:

$$\mathrm{MSE} = \frac{1}{N}\sum_{N} (X\,(i,j) - Y\,(i,j))^{2}\ [48] \tag{1}$$

The intrinsic ambiguity and multimodality of the colorization problem make this loss function vulnerable. The mean of the set is the best way to solve the Euclidean loss if an object can take on a range of different a*b* values. The background colors such as clouds, dirt, pavement, and walls cover most of the areas of the images. The averaging error effect favors mostly covered color values in the ground truth image and as a result, the predicted color values are strongly biased towards the ground truth colors. For this reason, the distribution of a*b* values is biased towards the colors that are mostly found. To solve this problem, we introduce the Local Loss (LL) module in our Deep Localization (DL-Net) Network. This prevents the domination of major color values over minor color values. Our key insight is that a clear color-ground separation can dramatically improve colorization performance. The proposed loss calculation equation and strategy are shown in Equation 2 and Figure 1 correspondingly. According to the figure, the whole image is divided into five color groups. The loss functions of all groups are computed. Each loss function is propagated through the nodes where those color values are extracted. This learning process imposes local features to extract more

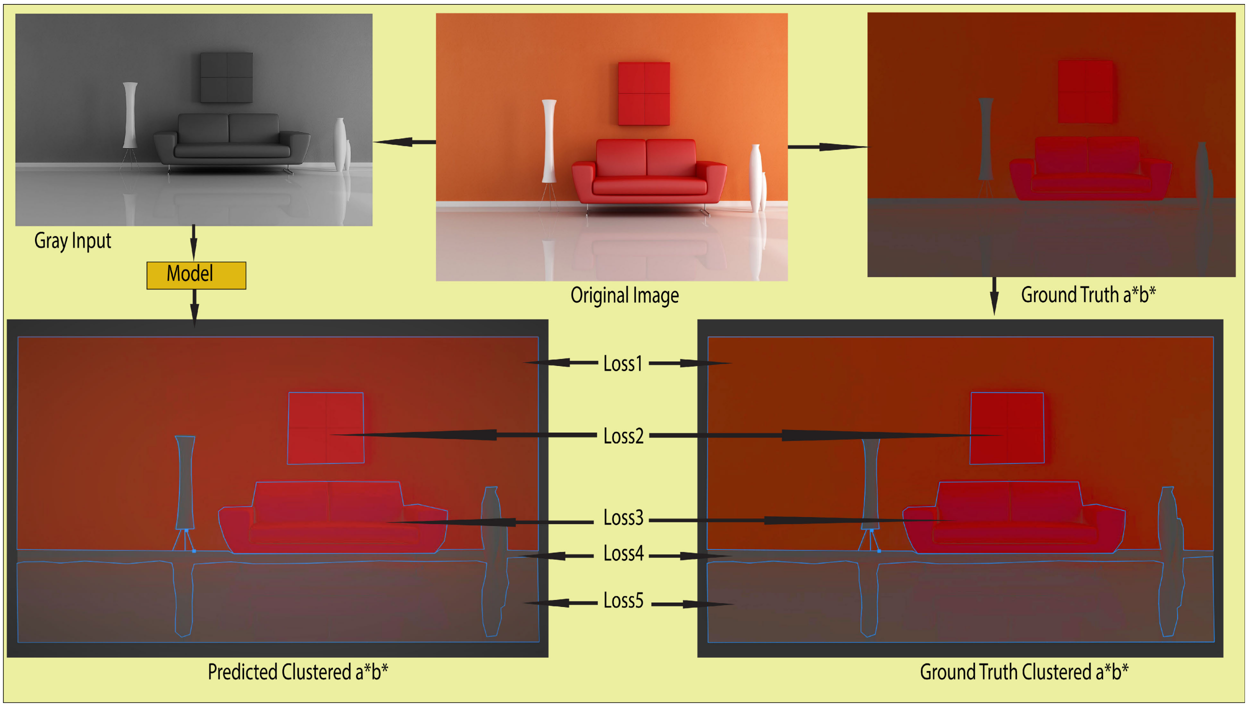


FIGURE 1. Loss functions according to color groups.

realistic color values. The proposed learning method, based on cluster-based losses, allows the colorization network to learn rare color representations for accurate colorization and to minimize the domination of the majority over the minority.

$$\text{Localized MSE} = \frac{\mathbf{1}}{N_1}\sum\nolimits_{N_1}(\boldsymbol{X}\,(\boldsymbol{i},\boldsymbol{j})-\boldsymbol{Y}\,(\mathbf{i},\boldsymbol{j}))^{\mathbf{2}} + \frac{\mathbf{1}}{N_2}\sum\nolimits_{N_2}(\boldsymbol{X}\,(\boldsymbol{i},\boldsymbol{j})-\boldsymbol{Y}\,(\mathbf{i},\boldsymbol{j}))^{\mathbf{2}} + \frac{\mathbf{1}}{N_3}\sum\nolimits_{N_3}(\boldsymbol{X}\,(\boldsymbol{i},\boldsymbol{j})-\boldsymbol{Y}\,(\mathbf{i},\boldsymbol{j}))^{\mathbf{2}} + \ldots\ldots\ldots + \frac{\mathbf{1}}{N_n}\sum\nolimits_{N_n}(\boldsymbol{X}\,(\boldsymbol{i},\boldsymbol{j})-\boldsymbol{Y}\,(\mathbf{i},\boldsymbol{j}))^{\mathbf{2}} \tag{2}$$

N= Number of pixels = $N_1 + N_2 + \ldots\ldots + N_n$
n = Numbers of clusters.

In this proposed encoder-decoder model the encoder is a Densely Connected Convolutional Network (DenseNet) [49] and the decoder is a conventional CNN [50]. The DenseNet extracts image features from gray images and the conventional CNN outputs the a*b* color channels. The DenseNet can extract high-level features which are very suitable for this colorization problem. The general structure of the proposed model is shown in Figure 2. After getting the output (a*b∗) of the deep learning model we applied the rebalancing technique [45] on a*b* channel values. Different regions of color channels are extracted and the chroma values are adjusted according to region wise using filtering techniques. It improves the image quality. The deep network architecture of our proposed model is shown in Figure 2. We will describe elaborately our model below.

A. FEATURE EXTRACTOR (DENSENET)

The DenseNet has a strong connection among its layers. It has less semantic information loss during feature extraction than other CNN architectures and reduces the gradient vanishing problem. The DenseNet concatenates its output from the previous layer with all of the future layers. We modify the first convolutional layer to make the model suitable for grayscale input. We discard the last linear layer to create a $\frac{H}{32} \times \frac{W}{32} \times 1024$ feature representation from DenseNet. These features are used as input in the colorization network, which is a CNN. The different convolutional layers and outputs of DenseNet are shown in Table 1.

B. COLORIZATION NETWORK (CNN)

The network takes $\frac{H}{32} \times \frac{W}{32} \times 1024$ feature representation as input and applies a series of convolutional and up-sampling layers. For up-sampling, we use the basic nearest-neighbor technique. The network outputs are $H \times W \times 2$ a*b* tensor. The different convolutional layers and their outputs are presented in Table 2.

C. GLOBAL LOSS FUNCTION AND GRADIENT DESCENT

Given a set of training data:

$$\mathrm{X} = \{(\mathrm{x}_1,\ \mathrm{y}_1),\ (\mathrm{x}_2,\ \mathrm{y}_2),\ (\mathrm{x}_3,\ \mathrm{y}_4),\ \ldots\ldots(\mathrm{x}_\mathrm{N},\ \mathrm{y}_\mathrm{N}) \tag{3}$$

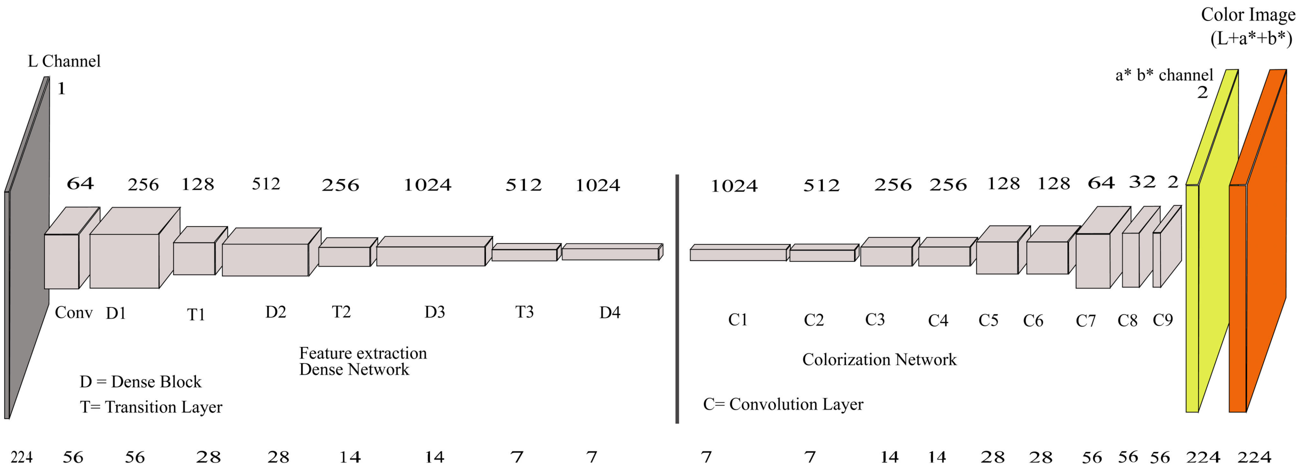


FIGURE 2. The structure of our proposed deep learning model.

TABLE 1. The model structure of the proposed Dense Network.

Layers	Output Size	DenseNet-121	Outputs
Convolution	112 × 112		64
Pooling	56 × 56		64
Dense Block 1	56 × 56	1 × 1 conv 3 × 3 conv × 6	256
Transition 1	56 × 56	1 × 1 × 128 conv	128
	28 × 28		
Dense Block 2	28 × 28	1 × 1 conv 3 × 3 conv × 12	512
Transition 2	28 × 28	1 × 1 × 256 conv	256
	14 × 14		
Dense Block 3	14 × 14	1 × 1 conv 3 × 3 conv × 24	1024
Transition 3	14 × 14	1 × 1 × 512 conv	512
	7 × 7		
Dense Block 4	7 × 7	1 × 1 conv 3 × 3 conv × 16	1024

TABLE 2. The model structure of the proposed Colorization Network.

Layers	Output Size	Kernel	Stride	Outputs
Conv-1	7 × 7	3 × 3	1 × 1	1024
Conv-2	7 × 7	3 × 3	1 × 1	512
Conv-3	14 × 14	3 × 3	1 × 1	256
Conv-4	14 × 14	3 × 3	1 × 1	256
Conv-5	28 × 28	3 × 3	1 × 1	128
Conv-6	28 × 28	3 × 3	1 × 1	128
Conv-7	56 × 56	3 × 3	1 × 1	64
Conv-8	56 × 56	3 × 3	1 × 1	32
Conv-9	56 × 56	3 × 3	1 × 1	2

where,

x_j = Ground truth pixel value
y_j = Predicted output pixel value by network
N = Number of pixels of each batch

Global Loss Function:

$$\mathbf{E(X,\ W)} = \frac{1}{N}\sum_{i,j=0}^{H,W}\left(X\left(i,j\right) - Y\left(i,j\right)\right)^2 \tag{4}$$

$$= \frac{1}{N}\sum_{i,j=0}^{H,W}\left(X_{ij} - f\left(a_{ij}^k\right)\right)^2 \tag{5}$$

N = H×W=Number of pixels
E(X, W) = Error function
W = Weight parameters of CNN
a_{ij}^k = Predicted value without activation
X_{ij} = Ground truth value for corresponding Predicted a_{ij}^k
f = Activation function
Then, the weight parameters are changed according to:

$$W^{i+1} = W^i - \alpha\frac{\partial E}{\partial W^i} \tag{6}$$

where

α = Learning Rate; i = Iteration
Gradient descent:

$$\frac{\partial E}{\partial W_{PQR}^k} = \sum_{i=0}^{h-l_1}\sum_{j=0}^{w-l_2}\sum_{z}\frac{\partial E}{\partial a_{Zij}^k}\times\frac{\partial a_{Zij}^k}{\partial W_{PQR}^k}\ [51] \tag{7}$$

Input Feature map dimension = h × w
Weight kernel dimension = $l_1 \times l_2$
Output Feature map dimension = (h - l_1 +1) (w - l_2+1)
k = Layer position of network
P = Kernel position of a layer
Q × R = row × column of a kernel
Z = output feature set position of a layer

$$a_{Zij}^k = \left(\sum_{l_1}\sum_{l_2}\left(w_{Pl_1l_2}^k\sum_{z}O_{z,i+l_1,j+l_2}^{k-1}\right)\right) + b_P^k \tag{8}$$

where, $\boldsymbol{W}$ = Weight value; O = Output value; b = bias value

$$\frac{\partial a_{Zij}^{k}}{\partial W_{PQR}^{k}} = \frac{\partial}{\partial W_{PQR}^{k}}\left[\left(\sum_{l_1}\sum_{l_2}(\,w_{Pl_1l_2}^{k}\sum_{z}O_{z,i+l_1,j+l_2}^{k-1}\,)\right) + b_P^k\right] = \sum_{z} O_{z,i+Q,j+R}^{k-1} \tag{9}$$

$$\delta_{Zij}^{k} = \frac{\partial E}{\partial a_{Zij}^{k}} = \sum_{m=0}^{l_1-1}\sum_{n=0}^{l_2-1}\sum_{z}\frac{\partial E}{\partial a_{Z,i-m,j-n}^{k+1}}\frac{\partial a_{Z,i-m,j-n}^{k+1}}{\partial a_{Z,ij}^{k}} \tag{10}$$

$$= \sum_{m=0}^{l_1-1}\sum_{n=0}^{l_2-1}\sum_{z}\delta_{Z,i-m,j-n}^{k+1}\frac{\partial a_{Z,i-m,j-n}^{k+1}}{\partial a_{Z,ij}^{k}} \tag{11}$$

$$\frac{\partial a_{Z,i-m,j-n}^{k+1}}{\partial a_{Zij}^{k}} = \frac{\partial}{\partial a_{Zij}^{k}}\sum_{P}\sum_{Z}W_{PQR}^{k+1}f\left(a_{Zij}^{k}\right) \tag{12}$$

$$= f'\left(a_{Zij}^{k}\right)\sum_{P}W_{PQR}^{k+1} \tag{13}$$

$$Then, \delta_{Zij}^{k} = \frac{\partial E}{\partial a_{Zij}^{k}} = \sum_{m=0}^{l_1-1}\sum_{n=0}^{l_2-1}\sum_{Z}\delta_{Z,i-m,j-n}^{k+1}\sum_{P}W_{PQR}^{k+1}f'\left(a_{Zij}^{k}\right) \tag{14}$$

As the backpropagation process is a chain rule, the gradient descents of the hidden layers are influenced by the gradient descents of output or final layer. In the final layer there exists single feature set. Thus, the value of Z and P is ignored from the gradient descent calculation of final layer.

Gradient Descent of Output Layer:

From Equation 5,

$$\mathbf{E} = \frac{1}{N}\sum_{i,j=0}^{H,W}\left(X_{ij} - f\,(a_{ij}^{k})\right)^2$$

Now,

$$\delta_{11}^{k} = \frac{\partial E}{\partial a_{11}^{k}} = -\frac{2}{N}\{X_{11} - f\,(a_{11}^{k})\}f'(a_{11}^{k}) = -\frac{2}{N}\,(x_{11} - y_{11})f'(a_{11}^{k}) \tag{15}$$

$$\delta_{32}^{k} = \frac{\partial E}{\partial a_{32}^{k}} = -\frac{2}{N}\{X_{32} - f\,(a_{32}^{k})\}f'(a_{32}^{k}) = -\frac{2}{N}\,(x_{32} - y_{32})f'(a_{32}^{k}) \tag{16}$$

$$\delta_{QR}^{k} = \frac{\partial E}{\partial a_{QR}^{k}} = -\frac{2}{N}\{X_{QR} - f\,(a_{QR}^{k})\}f'(a_{QR}^{k}) = -\frac{2}{N}\,(x_{QR} - y_{QR})f'(a_{\mathrm{QR}}^{k}) \tag{17}$$

From Equation 7 we get,

$$\frac{\partial E}{\partial W_{QR}^{k}} = \sum_{i=0}^{h-l_1}\sum_{j=0}^{w-l_2}\sum_{z}\frac{\partial E}{\partial a_{Zij}^{k}}\times\frac{\partial a_{Zij}^{k}}{\partial W_{PQ}^{k}}$$

Then,

$$\frac{\partial E}{\partial W_{11}^{k}} = \sum_{i=0}^{h-l_1}\sum_{j=0}^{w-l_2}\delta_{ij}^{k}O_{i+1,j+1}^{k-1} = -\frac{2}{N}\sum_{i=0}^{h-l_1}\sum_{j=0}^{w-l_2}(X_{ij} - Y_{ij})f'(a_{ij}^{k})O_{i+1,j+1}^{k-1} \tag{18}$$

$$\frac{\partial E}{\partial W_{32}^{k}} = \sum_{i=0}^{h-l_1}\sum_{j=0}^{w-l_2}\delta_{ij}^{k}O_{i+3,j+2}^{k-1} = -\frac{2}{N}\sum_{i=0}^{h-l_1}\sum_{j=0}^{w-l_2}(X_{ij} - Y_{ij})f'(a_{ij}^{k})O_{i+3,j+2}^{k-1} \tag{19}$$

$$\frac{\partial E}{\partial W_{m'n'}^{k}} = \sum_{i=0}^{h-l_1}\sum_{j=0}^{w-l_2}\delta_{ij}^{k}O_{i+m',j+n'}^{k-1} = -\frac{2}{N}\sum_{i=0}^{h-l_1}\sum_{j=0}^{w-l_2}(X_{ij} - Y_{ij})f'(a_{ij}^{k})O_{i+m',j+n'}^{k-1} \tag{20}$$

D. LOSS LOCALIZATION AND EFFECT ON LEARNING

We divide the targeted a*b* image of each batch of the training data into multiple areas based on its color groups using k-means clustering. Then we again divide the output a*b* image of the CNN of that batch into the same number of areas based on the pixel location of the corresponding cluster. We calculate losses based on color groups instead of computing the loss on the whole image. Let $L1$, $L2$,...,Ln be the losses of $n$cluster colors. Each Li propagates on those nodes of the hinder layers in which corresponding color features are extracted. For this reason, the proposed unbiased training method can overcome the pitfalls of conventional deep learning methods for imbalanced feature problems.

Proposed Localized Loss Function:

$$\mathbf{E}\,(\mathbf{X},\ \mathbf{W}) = \frac{1}{N_1}\sum_{N_1}(X\,(\mathbf{i},\mathbf{j}) - Y\,(\mathbf{i},\mathbf{j}))^2 + \frac{1}{N_2}\sum_{N_2}(X\,(\mathbf{i},\mathbf{j}) - Y\,(\mathbf{i},\mathbf{j}))^2 + \ldots + \frac{1}{N_n}\sum_{N_n}(X\,(\mathbf{i},\mathbf{j}) - Y\,(\mathbf{i},\mathbf{j}))^2 \tag{21}$$

And the final layer representation of Localized Loss:

$$\mathbf{E} = \frac{1}{N_1}\sum_{i,j=0}^{H_1W_1}\left(X_{ij} - f\,(a_{ij}^{k})\right)^2 + \frac{1}{N_2}\sum_{i,j=0}^{H_2W_2}\left(X_{ij} - f\,(a_{ij}^{k})\right)^2 + \ldots\ldots + \frac{1}{N_n}\sum_{i,j=0}^{H_nW_n}\left(X_{ij} - f\,(a_{ij}^{k})\right)^2 \tag{22}$$

where,

N = Number of pixels of each batch = $N_1 + N_2 + \ldots\ldots + N_n$

N_i = Number of pixels in cluster i = $H_i \times W_i$

n = Numbers of Clusters

f = Activation function

Gradient descents will be calculated separately under each cluster. We assume δ^k_{11} is calculated under cluster 1, δ^k_{32} is calculated under cluster 2, and δ^k_{QR} is calculated under the last cluster n.

$$\delta^k_{11} = \frac{\partial E}{\partial a^k_{11}} = -\frac{2}{N_1}\{X_{11} - f\,(a^k_{11})\} f'(a^k_{11}) = -\frac{2}{N_1}(x_{11} - y_{11}) f'(a^k_{11}) \tag{23}$$

$$\delta^k_{32} = \frac{\partial E}{\partial a^k_{32}} = -\frac{2}{N_2}\left\{X_{32} - f\left(a^k_{32}\right) f'\left(a^k_{32}\right)\right. = -\frac{2}{N_2}(x_{32} - y_{32}) f'\left(a^k_{32}\right) \tag{24}$$

$$\delta^k_{QR} = \frac{\partial E}{\partial a^k_{QR}} = -\frac{2}{N_n}\{X_{QR} - f\,(a^k_{QR}) f'(a^k_{QR}) = -\frac{2}{N_n}(x_{QR} - y_{QR}) f'(a^k_{QR}) \tag{25}$$

Now,

$$\frac{\partial E}{\partial W^k_{11}} = \sum_{i=0}^{h-l_1}\sum_{j=0}^{w-l_2} \delta^k_{ij} O^{k-1}_{i+1,j+1} = \sum_{i=0}^{h-l_1}\sum_{j=0}^{w-l_2} -\frac{2}{N_1}(X_{ij} - Y_{ij}) f'(a^k_{ij}) O^{k-1}_{i+1,j+1} \tag{26}$$

$$\frac{\partial E}{\partial W^k_{32}} = \sum_{i=0}^{h-l_1}\sum_{j=0}^{w-l_2} \delta^k_{ij} O^{k-1}_{i+3,j+2} = \sum_{i=0}^{h-l_1}\sum_{j=0}^{w-l_2} -\frac{2}{N_2}(X_{ij} - Y_{ij}) f'(a^k_{ij}) O^{k-1}_{i+3,j+2} \tag{27}$$

$$\frac{\partial E}{\partial W^k_{m'n'}} = \sum_{i=0}^{h-l_1}\sum_{j=0}^{w-l_2} \delta^k_{ij} O^{k-1}_{i+m',j+n'} = \sum_{i=0}^{h-l_1}\sum_{j=0}^{w-l_2} -\frac{2}{N_n}(X_{ij} - Y_{ij}) f'(a^k_{ij}) O^{k-1}_{i+m',j+n'} \tag{28}$$

δ^k_{ij} is calculated under cluster t with N_t pixels.

From the Equation 18, 19, and 20 for the global loss, we see that every gradient is impacted by the total pixel values of each batch. But, from the Equation 26, 27, and 28 we see that, using the proposed localized loss function, the gradients of the major and the minor pixel are impacted by the perspective values of the cluster groups. Although the calculation of the gradient descent of hidden layers is the same as above in Equation 7 to 14, the impact of the gradient descent of the output layer (with local loss) propagates to the gradient descent of hidden layers.

IV. EXPERIMENT

A. DATASET

We used the place365 [52] validation dataset to train and verify the proposed model. Place365 validation set contains 36500 images. Place365 validation dataset consists of 365 scene categories containing 100 images per category. We took 80% for training and 20% for testing from each scene category.

B. ENVIRONMENT SET UP AND TRAIN DETAIL

Using PyTorch [53] and Python 3.7, we developed our environment. We used a GeForce GTX 1050 graphics card with 4GB of RAM and an Intel Core i5 8400 desktop computer as hardware. Adam optimizer [54] has been used to backpropagate the loss during the training. The learning rate was set to 0.001. We train and validate our proposed model using the Google Colab [55] with local GPU runtime. We could maximize batch size 32 to avoid the overflow of GPU memory. Each input was resized to 224 × 224 pixels. This configuration and setup took approximately 15 days without any interruption to complete 40 epochs (45,000 iterations) for 36500 images.

C. EVALUATION METRICS AND COMPARISON METHODS

1) EVALUATION METRICS

To assess the accuracy of the predictions, we employ both quality and quantity measurements. For quality assessment, visual perception is used to show the performance of the model. For quantity assessment, a number of evaluation markers such as Color peak signal-to-noise ratio [56] (PSNR), structural similarity [56] (SSIM), and MSE_{RGB} are chosen as indicators.

2) QUALITATIVE COMPARISON

We first visually inspected the model's color image quality.

In the case of qualitative analysis, we first evaluate our proposed network model with global and local loss modules. In Figure 3A, we have shown six different images ordered column-wise from our proposed colorization method with global loss and local loss along with the ground truth and the gray version by which the proposed model generated the color image. The local loss outputs reliable color in every region of objects of each image. Moreover, images by the local loss module are deeper and brighter compared to the ground truth.

There exists object color mismatch and desaturated color as the result of the global loss. The global loss module

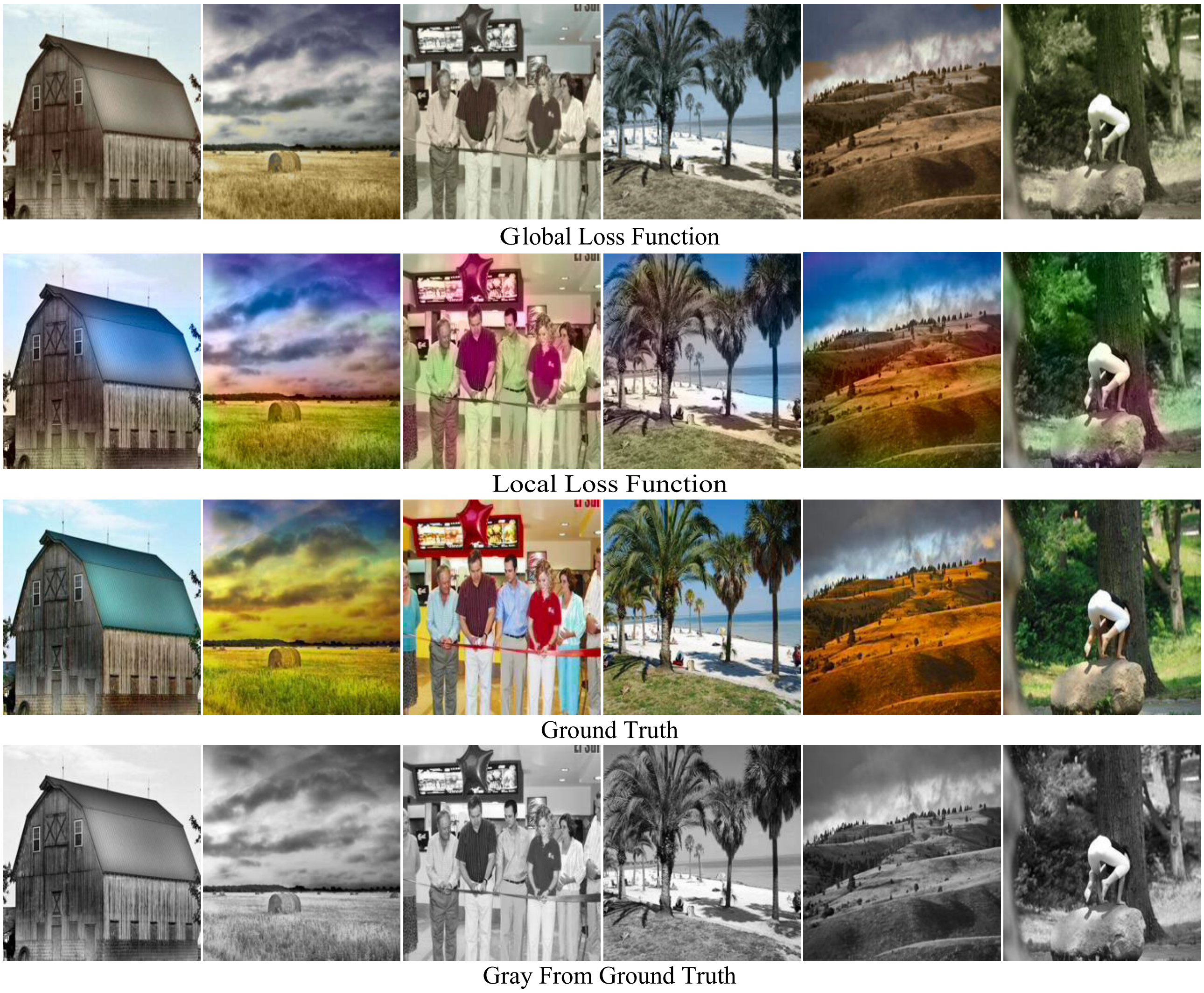


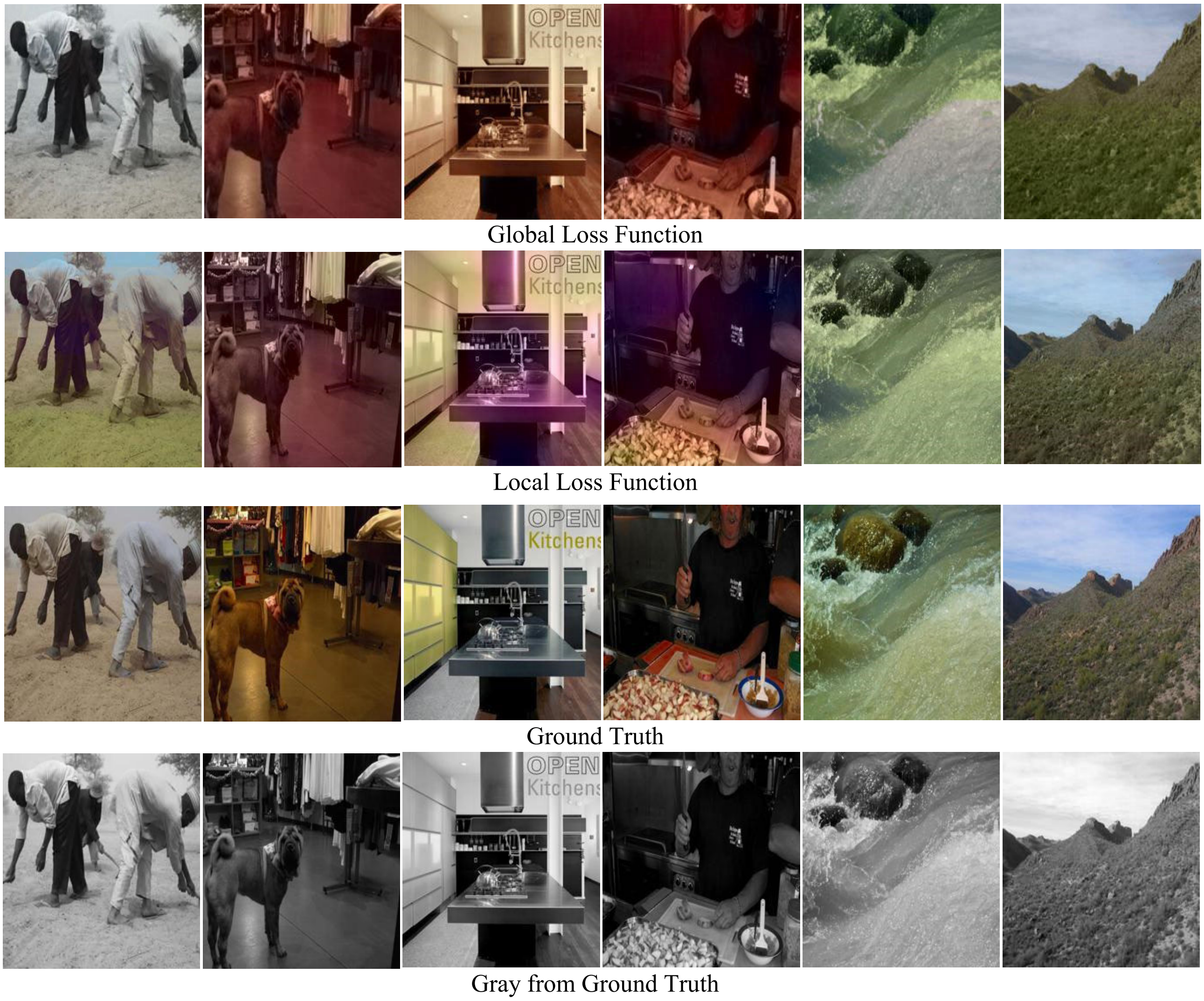
FIGURE 3. A). Some results of the local loss and global loss. The first row contains the result of global loss, the second row is the result of our proposed local loss and the third row is the ground truth and fourth row is the input gray images.

form desaturation and object color mismatch problem in the first, third, fourth, and sixth images. The proposed local loss module chooses the objects-wise suitable color and also forms saturation. In the second and fifth images, the global loss module chooses some colors correctly but remains desaturation.

In Figure 3B, we have shown another six different images order column-wise that are outputs of global loss and proposed local loss using the same network, ground truth, and gray images used as input of the models.

In the first and fifth images, the global loss fails to choose a good color in all areas. But the local loss chooses very plausible colors in all areas. In the second to fourth images, the global loss chooses the same color in all areas but the local loss chooses reliable color. In the sixth image global loss chooses the wrong color. For this why the dump of dust looks like a green hill. But the local loss chooses satisfactory color which helps to find the original image content. From Figure 3A and Figure 3B, we see that the proposed method trained with a satisfactory level using the given training data. In the case of qualitative analysis, we also compare our method with seven other methods. In Figure 4, we have displayed eight images row-wise where columns represent outputs of the proposed method and other methods, ground truth, and input gray images. For the second, fourth, sixth, and eighth images, all existing methods form desaturation and failed to peak the correct colors of the objects in the images except Gain et. al. [45] at the sixth and Su et. al. [26] at the eighth image. Our proposed method peaks object-wise saturated color and look similar to the original. For the first, third, fifth and seventh images, every model chooses some colors but the results of our proposed method are more similar to ground truth compared to any other method.

In Figure 5, we have displayed another seven different images in order row-wise for showing the qualitative



Global Loss Function

Local Loss Function

Ground Truth

Gray from Ground Truth

FIGURE 3. ***(Continued.)*** **B). Some results of the local loss and global loss. The first row contains the result of global loss, the second row is the result of our proposed local loss and the third row is the ground truth and fourth row is the input gray images.**

comparison of our method with seven other methods. In these ground truth images; colors of some objects look different colors that usually do not exist in nature. For the first three images and the seventh image, every method along with our proposed method chooses a different color from the ground truth. But in visual perception, the output of our method looks more vibrant than that of the original images. In the fourth and fifth images, the outputs of our proposed method look similar to the ground truth images. However, some methods generate over-saturated color images and others generate desaturated color images.

In the sixth image, all existing methods form a grayish effect but the outputs of our proposed method look very pleasant though it chooses a different color from the ground truth. From visual perception, we can say that our method can produce more plausible images compared to others. These results show us that the proposed method trained with a satisfactory level using the given training data. Additionally, we displayed the results of the colorization of old and contemporary photographs downloaded from the internet which are shown in Figure 7.

3) QUANTITATIVE COMPARISON

To assess the quantitative performance of various works, we employ the Peak Signal-to-Noise Ratio [56] (PSNR), Structural Similarity Index Measure (SSIM) [56], and MSE loss for the similarity metric approach. The image quality may be roughly evaluated using PSNR and typically, the greater the PSNR the better the image quality. The similarity between the reconstruction image and the original image is evaluated at the pixel level. A higher SSIM indicates greater structural similarity. The SSIM calculates the correlation between two pictures (generated and ground truth). PSNR can

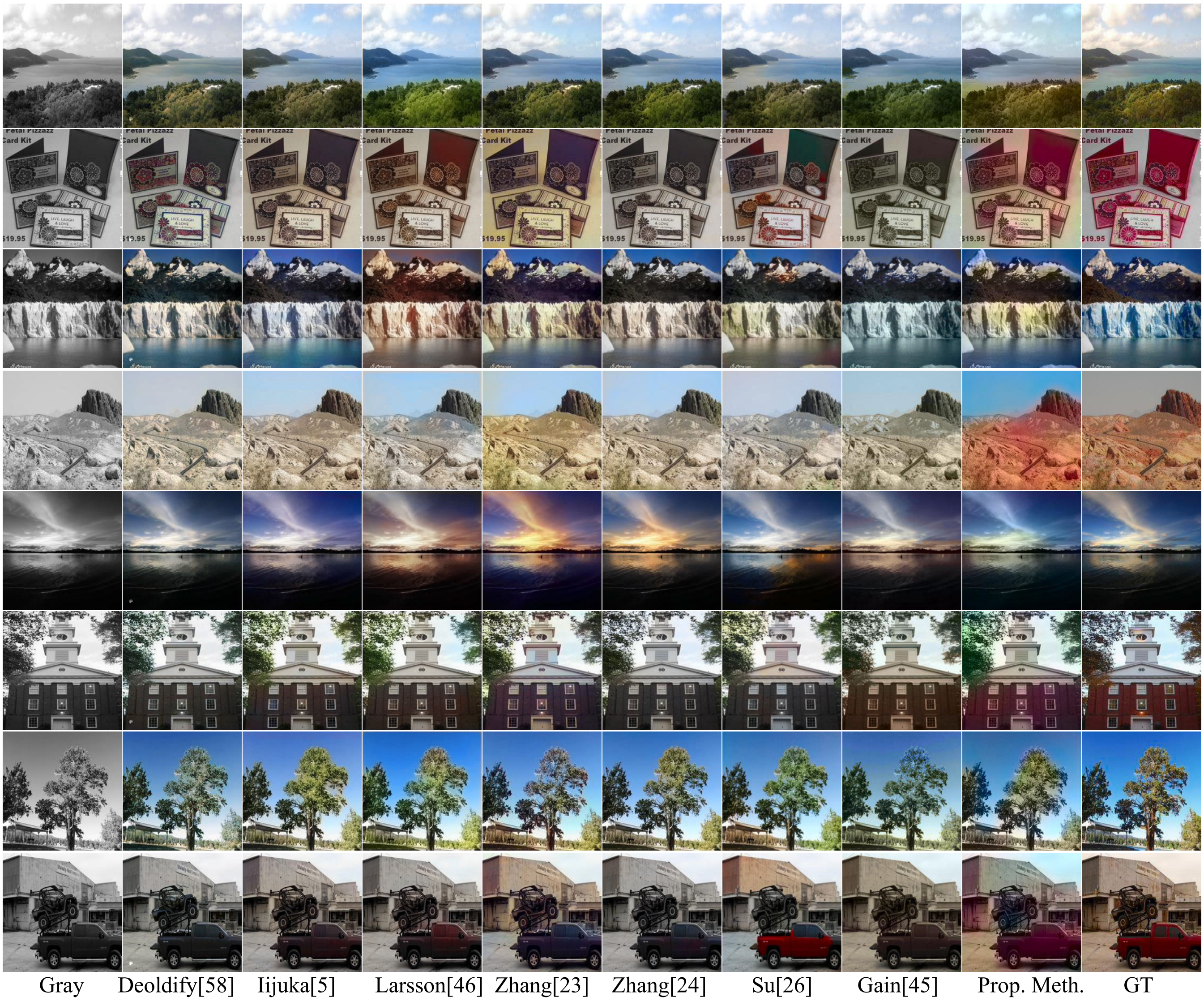


FIGURE 4. Outputs of our proposed method and some existing methods, ground truth and input gray image.

be defined as follows:

$$PSNR = 10log10\frac{255^2}{MSE} \tag{29}$$

where, $MSE = \frac{1}{N}\sum_{N}\{X(i,j) - Y(i,j)\}^2$

Here, *X(i,j)* and *Y(i,j)* denote (i, j)th pixels in both the output and ground truth RGB images. N represents the number of pixels of the sample. The SSIM can be defined as follows:

$$SSIM(X, Y) = \frac{(2\boldsymbol{\mu}_X\boldsymbol{\mu}_Y + \boldsymbol{c}_1)(2\sigma_{XY} + \boldsymbol{c}_2)}{(\mu_X^2 + \mu_Y^2 + \boldsymbol{c}_1)(\mu_X^2 + \mu_Y^2 + \boldsymbol{c}_2)} \tag{30}$$

where, μ_X and μ_Y present the average of X and Y respectively whereas σ_X and σ_Y indicate the variance of X and Y, respectively. Moreover, σ_{XY} expresses the covariance of X and Y. Here, $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$ with $k_1 = 0.01$, $k_2 = 0.03$, and $L = 255$ [57]

We have compared our results with different methods (Deoldify [58], Iizuka et. al. [5], Larsson et. al. [46], Zhang et. al. [23], Zhang et. al. [24], Su et. al. [26], Gain et. al. [45]). We have taken eight images randomly from the dataset. The ground truth, input gray images, and the outputs of the proposed methods and other existing methods are shown in Figure 6.

From Figure 6, we see that the method in Deoldify [58] chooses a desaturated color in every image. The method in Iizuka et al. [5] forms desaturation in the second, fifth and seventh images and Su et al. [26] in the second and fourth. Larsson [46] and Zhang et al. [25] form desaturation in the second, fourth, fifth, and sixth images. Zhang et al. [23] forms desaturation in the second and over-saturation in the third images. Gain et al. [45] form saturation in the seventh and eighth images. We find that there is no significant difference visually between the

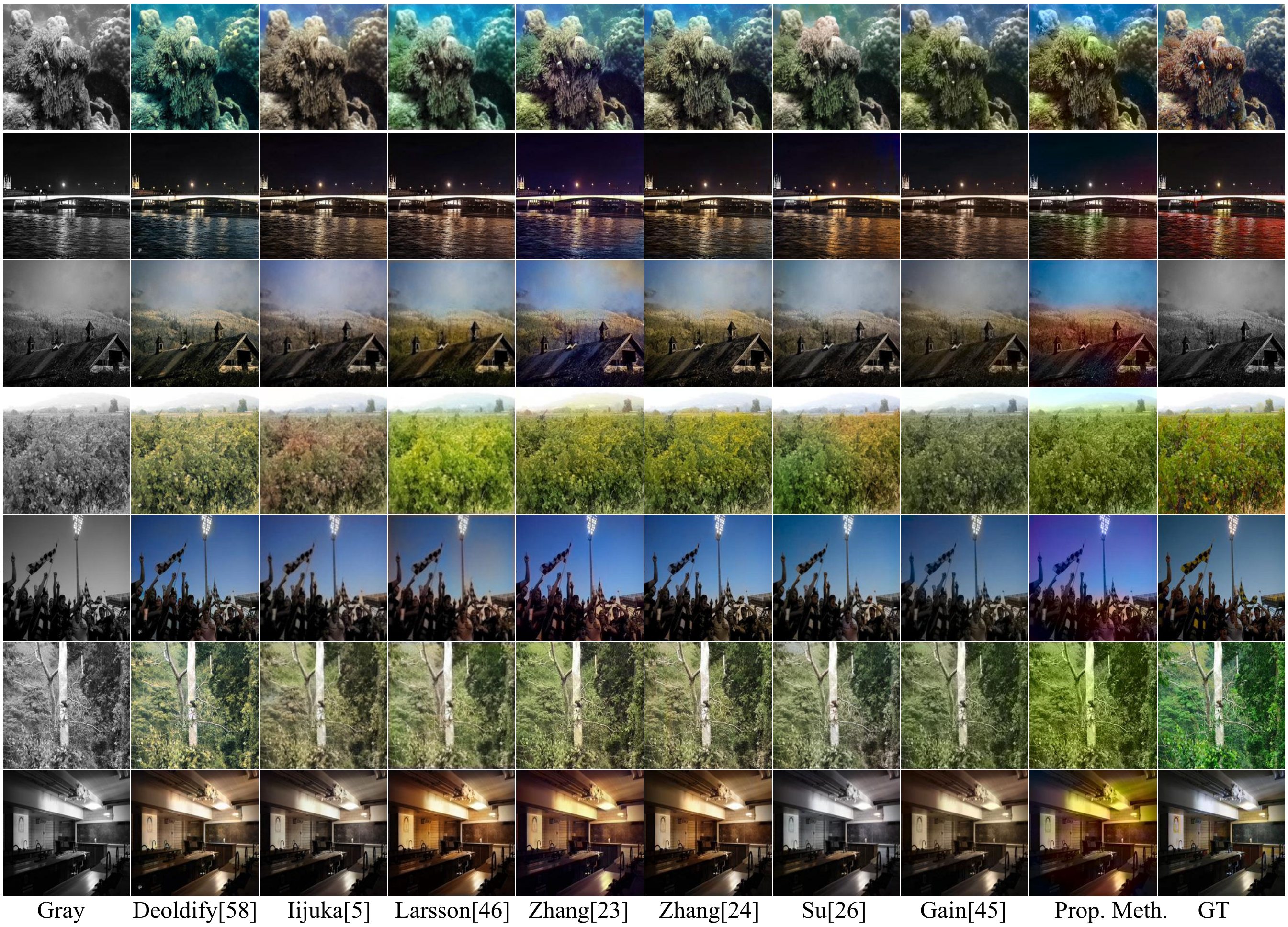


FIGURE 5. Results of our proposed method and existing methods, ground truth and input gray image.

image generated from the proposed method and the ground truth.

Table 3 shows a comparison of MSE, PSNR, and SSIM values between the global loss and the proposed local loss module for different images. From Table 3, we find that the average MSE value of our proposed localized method is smaller than the global loss method. The average PSNR and SSIM values for the proposed localized method are better than that of the global loss method for all the considered images. The average MSE value of the proposed localized method is 29% less than the global loss method. Besides, the average PSNR and SSIM value of the proposed localized method is 12.22 % and 5.29% higher correspondingly than the global loss method.

Table 4 shows a comparison of MSE, PSNR, and SSIM values among different methods for different images. From Table 4, we find that the average MSE value of our method is smaller than other methods. The average PSNR and SSIM values for the proposed method are better than that of all other methods for all the considered images. The MSE values of the proposed method are 88%, 36%, 60%, 42%, 56%, 55%, and 39%, which are less than that of the methods in Deoldify [58], Iizuka et. al. [5], Larsson et. al. [46], Zhang et. al. [23], Zhang et. al. [24], Su et. al. [26] and Gain et. al. [45], respectively.

Besides, the average PSNR values of the proposed method are 7.81%, 3.50%, 6.50%, 7.15%, 5.04%, 7.77%, and 5.69%, which are higher than that of the methods in Deoldify [58], Iizuka et. al. [5], Larsson et. al. [46], Zhang et. al. [23], Zhang et. al. [24], Su et. al. [26] and Gain et. al. [45] respectively.

In addition, the average SSIM values of the proposed methods are 1.46%, 0.32%, 1.00%, 1.00%, 0.25%, 5.05%, and 1.10% that are better than that of the methods in Deoldify [58], Iizuka et. al. [5], Larsson et. al. [46], Zhang et. al. [23], Zhang et. al. [24], Su et. al. [26], and Gain et. al. [45] respectively.

D. DISCUSSION

Colorization is a very ill-posed problem as there is no linear formula to generate color values from luminance components. Deep learning techniques show notable progress in the colorization process in recent times. In this paper, we proposed a new Local Loss (LL) model for image

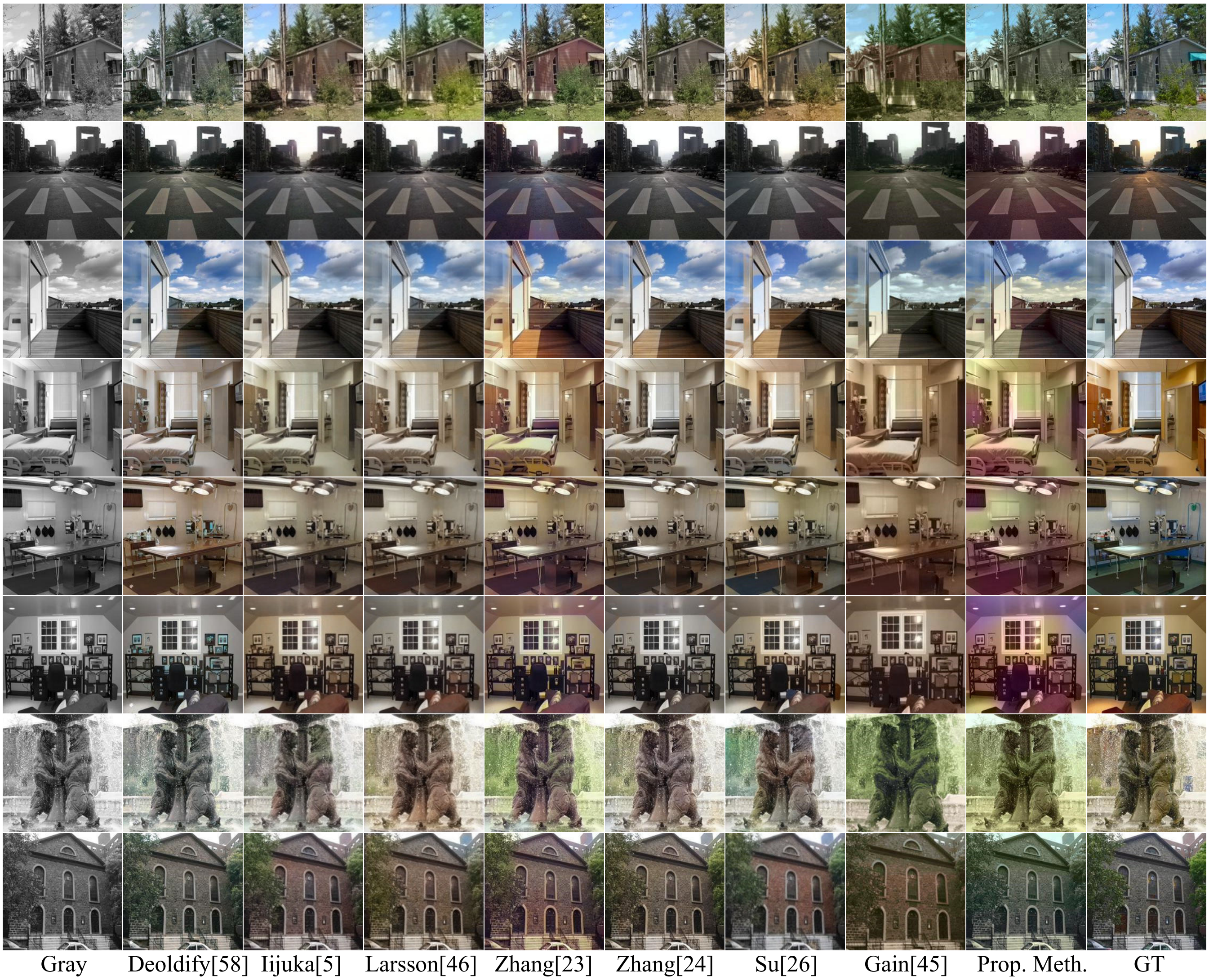


FIGURE 6. Comparison of results of our proposed method with some other methods and ground truth image.

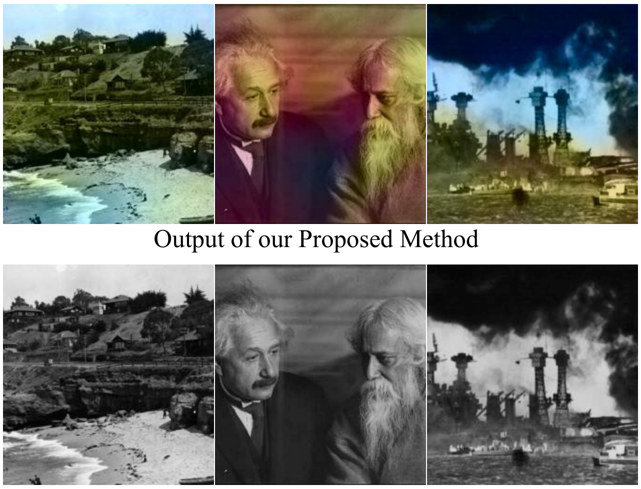
TABLE 3. Comparison of MSE, PSNR and SSIM between global and local loss methods. Here, Image No. (3A #1 to 3A #6) are corresponding to Figure 3A and image No. (3B #1 to 3B #6) are corresponding to Figure 3B.

Image No.		3A #1	3A #2	3A #3	3A #4	3A #5	3A #6	3B #1	3B #2	3B #3	3B #4	3B #5	3B #6
Global Loss	MSE ↓	361.1	**2207.8**	1239.1	345.7	**665.6**	273.7	766.07	656.9	722.6	586.1	248.9	577.7
	PSNR↑	22.55	**14.691**	17.199	22.742	**19.89**	23.75	19.288	19.65	19.54	20.45	24.17	20.51
	SSIM ↑	0.897	0.760	0.772	0.878	0.783	0.892	0.907	0.803	0.847	0.717	0.905	0.901
Local Loss	MSE ↓	**279.5**	2370.2	**970.19**	**299.7**	985.1	**218.1**	**134.17**	**363.71**	**550.5**	**365.2**	**96.33**	**54.54**
	PSNR↑	**23.66**	14.382	**18.262**	**23.362**	18.20	**24.74**	**26.85**	**22.52**	**20.72**	**22.50**	**28.29**	**30.76**
	SSIM ↑	**0.950**	**0.810**	**0.820**	**0.930**	**0.802**	**0.910**	**0.961**	**0.840**	**0.887**	**0.739**	**0.971**	**0.966**

colorization. The performance of the deep learning model largely depends on the balance state of the dataset. Unlike other image processing problems, balance data indicate the balance state at the feature (intensity) level instead of the sample level in the colorization task. When a small color area exists inside a large color area, the larger color area affects the small color area during training and the model blends small region color in its background. The number of desaturated pixels is higher than the number of saturated pixels in an image because the background areas are very larger than the main object area. For this reason, color bleeding problems and desaturated problems are found in the predicted color image

TABLE 4. Comparison of MSE, PSNR and SSIM among different methods. Here, Image No. (1 to 8) are corresponding to Figure 6.

Image No.		Deoldify *et. al.* [58]	Iijuka *et. al.* [5]	Larsson *et. al.* [48]	Zhang *et. al.* [23]	Zhang *et. al.* [24]	Su *et. al.* [26]	Gain *et. al.* [45]	Proposed Method
1	MSE ↓	350.54	293.75	392.62	256.36	180.10	459.58	234.57	**177.24**
	PSNR ↑	22.68	23.45	22.19	24.04	25.57	21.50	24.42	**25.64**
	SSIM ↑	0.943	0.946	0.936	0.945	0.954	0.889	0.943	**0.947**
2	MSE ↓	89.17	93.42	95.11	175.20	97.61	108.81	85.67	**79.20**
	PSNR ↑	28.62	28.42	28.34	25.69	28.23	27.76	28.80	**29.14**
	SSIM ↑	0.964	0.967	0.961	0.952	0.967	0.927	0.969	**0.971**
3	MSE ↓	**111.19**	123.32	118.51	398.62	169.01	260.36	157.42	138.42
	PSNR ↑	**27.66**	27.22	27.39	22.12	25.85	23.97	26.16	26.73
	SSIM ↑	0.955	0.957	**0.960**	0.936	0.953	0.912	0.941	0.946
4	MSE ↓	849.62	585.14	665.8	445.50	586.95	499.55	417.73	**381.67**
	PSNR ↑	18.83	20.45	19.89	21.64	20.44	21.14	21.92	**22.31**
	SSIM ↑	0.881	0.898	0.893	**0.904**	0.899	0.876	0.902	0.902
5	MSE ↓	391.18	**172.97**	221.26	294.77	191.50	237.12	307.62	218.53
	PSNR ↑	22.20	**25.75**	24.68	23.43	25.30	24.38	23.25	24.24
	SSIM ↑	0.923	**0.948**	0.944	0.934	0.947	0.906	0.923	0.942
6	MSE ↓	617.36	393.21	549.66	**212.64**	423.82	327.40	438.41	270.06
	PSNR ↑	20.22	22.18	20.72	**24.85**	21.85	22.98	21.71	23.81
	SSIM ↑	0.9081	0.944	0.931	0.955	0.942	0.918	0.938	**0.958**
7	MSE ↓	324.30	239.49	193.97	194.12	174.57	285.33	294.76	**151.14**
	PSNR ↑	23.02	24.37	25.25	25.25	25.71	23.57	23.43	**27.16**
	SSIM ↑	0.970	0.976	0.970	0.970	0.978	0.911	0.964	**0.989**
8	MSE ↓	93.84	86.71	166.46	164.22	92.42	148.86	151.19	**84.16**
	PSNR ↑	28.40	28.74	25.91	25.97	28.47	26.40	26.33	**28.87**
	SSIM ↑	0.9762	0.978	0.970	0.969	**0.980**	0.909	0.973	0.979


Output of our Proposed Method

Historical Gray Image

FIGURE 7. Colorization of Legacy Images.

in the existing methods. Because in the case of regression loss, the effect of minor features is affected by major features and the gradient descent of minor features disappeared during backpropagation. To solve this problem, we consider the MSE loss of each color object, where we separate different color regions using k-means clustering techniques. Then we apply the loss function separately on different regions to solve desaturation. From the experimental results, we found that our proposed method solved the above-mentioned problem effectively.

We compared our proposed method with several existing efficient methods and found that the visual effects of our proposed method outperform other methods, which are shown in Figure 4 and Figure 5. Utilizing the PSNR, SSIM, and MSE assessment measures, we also compared our suggested strategy with other methods. We found our proposed method performs well compared to other methods which are shown in Table 4, and Figure 6.

We also tested our proposed model using historical images and got plausible color visuals which are shown in Figure 7.

We have trained our model using only 28,800 images which is on average 17 times smaller than the above-mentioned compared methods. Because of hardware limitations, we fail to use more images. We hope if one can train with more training data, the performance of the model will be better than the present results of the proposed methods.

V. CONCLUSION

In this paper, we present a novel automatic image colorization model based on deep learning whose backpropagation algorithm is developed according to chromatic component-based Local Loss (LL). Existing learning-based colorization methods contain implausible color peak issues. This is because major colors in images outnumber minor colors in feature representation. The performance of the deep learning method depends on the balance state of training data. The optimization method is biased by the unbalanced feature representation, and the influence of minor colors is lost during optimization. As a result, minor colors disappeared by the model. We developed a Deep Localization Network (DL-Net) addressing this issue. The proposed method uses a chromatic component-based local loss in the backpropagation algorithm. By focusing component-based loss over local areas, local losses are produced. It increases the diverse-range color modeling and helps to remove contextual ambiguity. Besides, it can prioritize the selected semantic elements from the source image which encourages the creation of more plausible coloring. In terms of MSE, PSNR, and SSIM, our proposed method outperforms the more sophisticated existing techniques.

Due to hardware limitations, our learning model is currently trained with small-sized data. More training data may improve colorization by our method, but this also necessitates more hardware resources. We will work with more data in the future. Moreover, our proposed local loss backpropagation algorithm can be used on any deep learning or other learning models to efficiently solve feature imbalanced regression problems.

REFERENCES

[1] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu, "An adaptive edge detection based colorization algorithm and its applications," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, Hilton, Singapore, Nov. 2005, pp. 351–354.

[2] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *Proc. ACM SIGGRAPH Papers*. Rochester, NY, USA: ACM, Aug. 2004, pp. 689–694.

[3] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1120–1129, May 2006.

[4] J. An, K. K. Gagnon, Q. Shi, H. Xie, and R. Cao, "Image colorization with convolutional neural networks," in *Proc. 12th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, Manhattan, NY, USA, Oct. 2019, pp. 1–4.

[5] 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. 1–11, Jul. 2016.

[6] U. Sousa, R. Kabirzadeh, and P. Blaes, "Automatic colorization of grayscale images," Dept. Elect. Eng., Stanford Univ., Stanford, CA, USA, Tech. Rep., 2013.

[7] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," in *Proc. 29th Annu. Conf. Comput. Graph. Interact. Techn.*, Texas, SA, USA, Jul. 2002, pp. 277–280.

[8] R. K. Gupta, A. Y.-S. Chia, D. Rajan, E. S. Ng, and H. Zhiyong, "Image colorization using similar images," in *Proc. 20th ACM Int. Conf. Multimedia*, Nara, Japan, Oct. 2012, pp. 369–378.

[9] Y. Qu, T.-T. Wong, and P.-A. Heng, "Manga colorization," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 1214–1220, Jul. 2006.

[10] V. S. Devi and S. Kannimuthu, "Author profiling in code-mixed WhatsApp messages using stacked convolution networks and contextualized embedding based text augmentation," *Neural Process. Lett.*, vol. 55, pp. 1–26, Jul. 2022.

[11] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Manhattan, NY, USA, Aug. 2017, pp. 1–6.

[12] K. R. M. Fernando and C. P. Tsokos, "Dynamically weighted balanced loss: Class imbalanced learning and confidence calibration of deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 7, pp. 2940–2951, Jul. 2022, doi: 10.1109/TNNLS.2020.3047335.

[13] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *Proc. ICLR*, 2017, pp. 1–12.

[14] B. C. Wallace and I. J. Dahabreh, "Improving class probability estimates for imbalanced data," *Knowl. Inf. Syst.*, vol. 41, no. 1, pp. 33–52, Oct. 2014.

[15] R. Dahl. (2016). *Automatic Colorization*. [Online]. Available: https://tinyclouds.org/colorize

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[17] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, FL, USA, Jun. 2015, pp. 447–456.

[18] J. Hwang and Y. Zhou, "Image colorization with deep convolutional neural networks," Stanford Univ., Stanford, CA, USA, Tech. Rep. 219, 2016, pp. 1–7.

[19] F. Baldassarre, D. G. Morín, and L. Rodés-Guirao, "Deep koalarization: Image colorization using CNNs and Inception-ResNet-v2," 2017, *arXiv:1712.03400*.

[20] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-V4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.* San Francisco, CA, USA: AAAI Press, 2017, pp. 1–12.

[21] P. Qin, Z. Cheng, Y. Cui, J. Zhang, and Q. Miao, "Research on image colorization algorithm based on residual neural network," in *Proc. CCF Chin. Conf. Comput. Vis.* Midtown Manhattan, NY, USA: Springer, 2017, pp. 608–621.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[23] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proc. Eur. Conf. Comput. Vis.* Midtown Manhattan, NY, USA: Springer, 2016, pp. 649–666.

[24] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, "Real-time user-guided image colorization with learned deep priors," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–11, Aug. 2017.

[25] P. Qin, "Image colorization algorithm based on dense neural network," *Int. J. Performability Eng.*, vol. 15, no. 1, pp. 270–280, 2019.

[26] J.-W. Su, H.-K. Chu, and J.-B. Huang, "Instance-aware image colorization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 7965–7974.

[27] J. Dai, B. Jiang, C. Yang, L. Sun, and B. Zhang, "Local pyramid attention and spatial semantic modulation for automatic image colorization," in *Proc. CCF Conf. Big Data*. Midtown Manhattan, NY, USA: Springer, 2022, pp. 165–181.

[28] M. Xu and Y. Ding, "Fully automatic image colorization based on semantic segmentation technology," *PloS one*, vol. 16, no. 11, pp. 1–25, 2021.

[29] D. Wu, J. Gan, J. Zhou, J. Wang, and W. Gao, "Fine-grained semantic ethnic costume high-resolution image colorization with conditional GAN," *Int. J. Intell. Syst.*, vol. 37, no. 5, pp. 2952–2968, May 2022.

[30] M. Hesham, H. Khaled, and H. Faheem, "Image colorization using scaled-YOLOv4 detector," *Int. J. Intell. Comput. Inf. Sci.*, vol. 21, no. 3, pp. 107–118, Nov. 2021.

[31] H. Guo, Z. Guo, Z. Pan, and X. Liu, "Bilateral Res-UNet for image colorization with limited data via GANs," in *Proc. IEEE 33rd Int. Conf. Tools Artif. Intell. (ICTAI)*, Manhattan, NY, USA, Nov. 2021, pp. 729–735.

[32] L. Liu, Q. Jiang, X. Jin, J. Feng, R. Wang, H. Liao, S.-J. Lee, and S. Yao, "CASR-net: A color-aware super-resolution network for panchromatic image," *Eng. Appl. Artif. Intell.*, vol. 114, Sep. 2022, Art. no. 105084.

[33] A. Kumar, D. S. George, and L. S. Binu, "Colorization of grayscale images using convolutional neural network and Siamese network," in *Machine Intelligence and Smart Systems*. Midtown Manhattan, NY, USA: Springer, 2022, pp. 297–308.

[34] G. Özbulak, "Image colorization by capsule networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Long Beach, CA, USA, Jun. 2019, pp. 2150–2158.

[35] G. Kong, H. Tian, X. Duan, and H. Long, "Adversarial edge-aware image colorization with semantic segmentation," *IEEE Access*, vol. 9, pp. 28194–28203, 2021.

[36] Y. Wu, X. Wang, Y. Li, H. Zhang, X. Zhao, and Y. Shan, ''Towards vivid and diverse image colorization with generative color prior,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 14357–14366.

[37] T.-T. Nguyen-Quynh, S.-H. Kim, and N.-T. Do, ''Image colorization using the global scene-context style and pixel-wise semantic segmentation,'' *IEEE Access*, vol. 8, pp. 214098–214114, 2020.

[38] H. Bahng, S. Yoo, W. Cho, D. K. Park, Z. Wu, X. Ma, and J. Choo, ''Coloring with words: Guiding image colorization through text-based palette generation,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, 2018, pp. 431–447.

[39] Y. Liang, D. Lee, Y. Li, and B.-S. Shin, ''Unpaired medical image colorization using generative adversarial network,'' *Multimedia Tools Appl.*, vol. 81, no. 19, pp. 26669–26683, Aug. 2022.

[40] S. Treneska, E. Zdravevski, I. M. Pires, P. Lameski, and S. Gievska, ''GAN-based image colorization for self-supervised visual feature learning,'' *Sensors*, vol. 22, no. 4, p. 1599, Feb. 2022.

[41] M. Wu, X. Jin, Q. Jiang, S.-J. Lee, W. Liang, G. Lin, and S. Yao, ''Remote sensing image colorization using symmetrical multi-scale DCGAN in YUV color space,'' *Vis. Comput.*, vol. 37, no. 7, pp. 1707–1729, Jul. 2021.

[42] M. Sugawara, K. Uruma, S. Hangai, and T. Hamamoto, ''Local and global graph approaches to image colorization,'' *IEEE Signal Process. Lett.*, vol. 27, pp. 765–769, 2020.

[43] M. Afifi, M. A. Brubaker, and M. S. Brown, ''HistoGAN: Controlling colors of GAN-generated and real images via color histograms,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 7937–7946.

[44] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, ''Analyzing and improving the image quality of StyleGAN,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 8107–8116.

[45] M. Gain, M. A. Rahman, R. Debnath, M. M. Alnfiai, A. Sheikh, M. Masud, and A. K. Bairagi, ''An improved encoder–decoder CNN with region-based filtering for vibrant colorization,'' *Comput. Syst. Sci. Eng.*, vol. 46, no. 1, pp. 1059–1077, 2023.

[46] G. Larsson, M. Maire, and G. Shakhnarovich, ''Learning representations for automatic colorization,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Midtown Manhattan, NY, USA: Springer, 2016, pp. 577–593.

[47] A. R. Robertson, ''The CIE 1976 color-difference formulae,'' *Color Res. Appl.*, vol. 2, no. 1, pp. 7–11, Mar. 1977.

[48] Z. Wang and A. C. Bovik, ''Mean squared error: Love it or leave it? A new look at signal fidelity measures,'' *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, Jan. 2009.

[49] M. D. Zeiler and R. Fergus, ''Visualizing and understanding convolutional networks,'' in *Proc. Eur. Conf. Comput. Vis.* Midtown Manhattan, NY, USA: Springer, 2014, pp. 818–833.

[50] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, ''Densely connected convolutional networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 4700–4708.

[51] P. Baldi, ''Gradient descent learning algorithm overview: A general dynamical systems perspective,'' *IEEE Trans. Neural Netw.*, vol. 6, no. 1, pp. 182–195, Jan. 1995, doi: 10.1109/72.363438.

[52] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, ''Places: A 10 million image database for scene recognition,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.

[53] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and A. Desmaison, ''Pytorch: An imperative style, high-performance deep learning library,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[54] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' 2014, *arXiv:1412.6980*.

[55] E. Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Berkeley, CA, USA: Apress, 2019, pp. 59–64.

[56] A. Hore and D. Ziou, ''Image quality metrics: PSNR vs. SSIM,'' in *Proc. 20th Int. Conf. Pattern Recognit.*, Manhattan, NY, USA, Aug. 2010, pp. 2366–2369.

[57] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, ''Image quality assessment: From error visibility to structural similarity,'' *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[58] J. Antic. (2018). *A Deep Learning Based Project for Colorizing and Restoring Old Images (and Video!)*. [Online]. Available: https://github.com/jantic/DeOldify

MRITYUNJOY GAIN received the B.S. degree in computer science from Khulna University, Bangladesh, in 2021, where he is currently pursuing the M.S. degree in computer science. He is also a Visiting Research Student with the University of Saskatchewan, Canada. His research interests include image processing, deep learning, machine learning, explainable artificial intelligence, transformer learning, pattern recognition, and their applications.

RAMESWAR DEBNATH (Member, IEEE) received the bachelor's degree (Hons.) in computer science and engineering from Khulna University, Bangladesh, in 1997, and the M.E. degree in communication and systems and the Ph.D. degree from The University of Electro-Communications, Tokyo, in 2002 and 2005, respectively, under the Japanese Government Scholarship.

From 2008 to 2010, he was a Postdoctoral Researcher under the JSPS Fellowship with the Department of Informatics, The University of Electro-Communications; the Neuroscience Research Institute; and the National Institute of Advanced Industrial Science and Technology, Tsukuba. From 2012 to 2015, he was the Head of the Computer Science and Engineering Discipline, Khulna University, where he is currently a Professor. He has published more than 50 journals, book chapters, and peer-reviewed conference papers. He presented papers in many conferences in home and abroad. His research interests include image data analysis, deep learning, bioinformatics, support vector machine, artificial neural networks, statistical pattern recognition, and medical image processing. He was the Organizing Chair of the 16th International Conference on Computer and Information Technology (ICCIT), in 2014.