Abstract—Colorization of gray-scale images has always been a challenging task in computer vision. Recently, novel approaches have been introduced for unsupervised image translation between two domains using Generative Adversarial Networks (GAN). Since one can consider the gray-scale and colorful images as two separate domains, we propose a two-stage cycle-consistent network architecture to generate convincible images. First, an intermediate image is generated with a relatively uncomplicated objective function at the output. Next, at the second stage, the intermediate image is enhanced via a residual network structure with a more complicated objective function. Furthermore, by employing two inverse networks, a cycle-consistent architecture is formed at both stages. The proposed model is trained on the ImageNet challenge dataset, and the achieved outcomes demonstrate exceptional performance comparing with the state-of-the-art models.

Index Terms—Cycle-Consistency, Generative Adversarial Network, Image Colorization, Residual Structure

I. INTRODUCTION

THE advent of the convolutional neural networks has created a new perspective on the problem of colorization of gray-scale images. These tools have made it possible to do several tasks in the field of computer vision which were too difficult to do without intervention of humans. Colorization of images is one of the problems which has recently been concerned, and the effectiveness of convolutional neural networks in this field has been investigated.

In this paper, the goal is to train a model that can predict the color corresponding to each pixel in the image, while it is assumed that only the gray-scale version of the image is provided as input to the model.

Authors in [1]-[4] have considered the image colorization as a self-supervised problem, and they propose various convolutional network architectures with different loss functions. However, since the invention of Generative Adversarial network (GAN) architecture [19], various GAN architectures have been utilized [5]-[8] to produce realistic output images in an unsupervised approach. GAN architectures consist of two networks; the generator network which is responsible for generating output image, and the discriminator network whose job is to discriminate generated images from the real original ones. The crucial part of the structure is that the generator, also known as “the counterfeiter”, should be trained to fool the discriminator, and consequently, produce real-looking results.

On the other hand, several works have been done under the concept of image-to-image translation. The goal is constructing a two-way structure which can translate images between two domains [9]-[12]. Since it is conceivable to consider gray-scale and colorful images as two different domains, we can apply this concept into our problem.

The architecture presented in [11] has provided great quality results, especially in the case of classic image processing problems. Accordingly, we build our proposed method upon this architecture and customize it for the colorization task. Besides, our structure consists of two stages, where the second one has a residual structure and is responsible for enhancing the quality of the images generated by the first stage.

The rest of this letter is organized as follows. In Section II, the detailed explanation of our method in addition to some background theories are explained. Experiments and outcomes of our proposed method in comparison to the results in [7], [8], and [11] are presented in Section III. Finally, Section IV includes concluding remarks.

II. THE PROPOSED METHOD

A. Background

Consider x and y as images from gray-scale and colorful domains respectively, while z comes from a random distribution and is considered as the input noise. A GAN introduced in [19] consists of two neural networks, a generator \( G: z \rightarrow y \) that maps a random noise vector z to output image y and a discriminator \( D \), whose task is to discriminate generated y from real images in \( \mathcal{Y} \) domain. The generator aims to produce real-looking outputs that cannot be distinguished from “real” images while a discriminator aims at detecting the generator’s “fakes” [8].

The objective function of a GAN is defined as
$$L_{GAN} = E_{y: p_{data}(y)}[\log(D_y(y))] + E_{z: p_z(z)}[\log(1 - D_y(G(z)))]$$

where $E$ stands for expected value, and $P$ stands for probability density function. In the case of conditional GAN, the input $x$ is incorporated in the generator $G: x, z \rightarrow y$.

Accordingly, the objective function would be

$$L_{\text{GAN}} = E_{y: p_{data}(y)}[\log(D_y(y))] + E_{x, z: p_{data}(x, z), p_z(z)}[\log(1 - D_y(G(x, z)))]$$

The $D_y$ and $G$ networks are trained simultaneously in the way that $D_y$ tries to minimize the $L_{\text{GAN}}$, and $G$ would maximize it. This objective function is incorporated inside the models in [7], and [8]. On the other hand, models in [9]-[12] consist of two additional networks: inverse network $F: y, z \rightarrow x$, and another discriminator in $X$ domain $D_x$. In addition to the similar objective function to Equation (2) in the $X$ domain, cycle consistency loss is also considered as

$$L_{\text{cyc}} = E_{x: p_{data}(x)}[\| x - F(G(x)) \|] + E_{y: p_{data}(y)}[\| y - G(F(y)) \|]$$

This objective function states that when an image is translated to another domain, and subsequently, is translated to its original domain, the result has to be as close as possible to the original image. This means that the translated image in the other domain has to maintain its essential features so that it can be recovered by the inverse generator network.

### B. Overall Architecture

Inspired by the Cycle-in-Cycle GAN model in [18], we generate our final output in two stages which are both supposed to have a cycle-consistent structure. The overall architecture is shown in Fig. 1: the first generator $G_1: x \rightarrow y_1$ would generate an intermediate image through a cycle-consistent GAN structure, while $D_1$ is responsible as the discriminator of the intermediate image.

The image $y_1$ alongside the input $x$ are inputs of the second stage of our proposed model whose goal is denoising the $y_1$ image and producing the realistic image $y_2$ as alike as possible to the ground truth image $y$. Since we provide an estimation of the output to the second stage by feeding $y_1$ to it, we can incorporate more complex objective functions for $y_2$ to produce a higher quality image. The idea is that the additional information provided by the first stage would make the training process of the second stage simpler, and consequently, more complex functions could be adopted. Moreover, a residual structure is used in the second stage so that the second generator $G_2: x, y_1 \rightarrow y_2$ only focuses on the quality enhancement, and doesn’t carry the burden of regenerating the image.

Authors of [7], [8], and [10] claim that feeding noise to input of generator is not effective in conditional GAN structures because generator networks would learn to ignore the noise; hence, providing noise in different forms are proposed in the literature. Concatenating noise to the input of the first half of all layers of the generator network is proposed in [7]. Also, applying noise only in the form of dropout [15] is suggested in [8] and [10]. Similarly, we only adopt dropout in all layers of our generator networks. Furthermore, at the second stage of our model, $y_2$ can be considered as the sum of the ground-truth image and the noise which is generated by $G_1$. So in this case, components of the noise which would degrade the quality of the final image $y_2$ could be canceled, and other parts of the noise could bring about diversity in the colorization of the final image.

### C. Objective Functions

The intermediate image $y_1$ should satisfy various objective functions. The first one is the adversarial objective function. Instead of using the formulation of [19], we used the loss function proposed in [20], known as Least Square GAN which is claimed to stabilize the training process. In [21] numerous GAN formulations are investigated, and it is shown that Least Square GAN is capable of producing high-quality images, and also partly improves the instability of the original GAN formulation. Therefore, in this formulation, the discriminator $D_1$ would minimize the objective function

$$L_{\text{dis}}^{\text{GAN1}} = E_{x: p_{data}(x)}[(D_1(y) - 1)^2] + E_{x: p_{data}(x)}[D_1(G_1(x))^2]$$

Conversely, the generator $G_1$ would minimize

$$L_{\text{gen}}^{\text{GAN1}} = E_{x: p_{data}(x)}[(D_1(G_1(x)) - 1)^2]$$

Cycle-consistency is preserved through inverse generator network $F_1$ and objective function of

$$L_{\text{cyc}} = E_{x: p_{data}(x)}[\|x - F_1(G_1(x))\|]$$

Using traditional identity loss between the input and output images is employed by [8], and [18]. Similarly, we utilize L1 distance as an objective function to encourage the generator networks not only to fool the discriminator networks, but also produce the output closer to ground truth image:

$$L_{\text{idt}} = E_{x: p_{data}(x, y)}[\| y - G_1(x) \|]$$

The overall objective function for the $G_1$ and $F_1$ is the weighted sum of all previously mentioned functions:

$$L_{\text{cyc1}} = \lambda_1 L_{\text{dis}}^{\text{GAN1}} + \lambda_2 L_{\text{cyc1}} + \lambda_3 L_{\text{idt}}$$

where $\lambda_1, \lambda_2, \lambda_3$ are constant parameters.

Similar objective functions are employed in the second stage of our model. The discriminator $D_2$ would minimize

$$L_{\text{dis}}^{\text{GAN2}} = E_{y: p_{data}(y)}[(D_2(y) - 1)^2] + E_{x: p_{data}(x)}[D_2(G_2(x, G_1(x)))^2]$$

The first three components of the objective function of $G_2$ and $F_2$ are similar to the objective function of $G_1$ and $F_1$, proved as follows:
Fig. 1. The overall architecture of our proposed method, where $x$ is the input gray-scale image, $y$ is the ground truth image, $y_1$ is the intermediate generated image, and $y_2$ is the final output image. $G_1$, $G_2$, $D_1$, and $D_2$ are generators and discriminators respectively. $F_1$ and $F_2$ are inverse generators which produce $x'$ and $x''$, the input images of cycle-consistency objective functions.

\[
L_{\text{GAN}}^{\text{gen}} = E_{x} : p_{\text{data}}(x) [((D_2(G_2(x,G_1(x)))) - 1)^2] \\
L_{\text{cyc}} = E_{x} : p_{\text{data}}(x) [\left\| (y - F_2(G_2(x,G_1(x)))) \right\|_2] \\
L_{\text{idt}} = E_{x} : p_{\text{data}}(x,y) [\left\| (y - G_2(x,G_1(x))) \right\|_2] \\
\text{where } \nabla_h \text{ and } \nabla_v \text{ are horizontal and vertical gradient operators respectively.}
\]

Furthermore, some more complex objective functions are adopted to increase the quality of the output image. The first one is a total variation of $y_2$ which would suppress the noise level in the image and improves the smoothness of the output.

\[
L_{\text{TV}} = E_{x} : p_{\text{data}}(x) [\left\| \nabla_h G_2(x,G_1(x)) \right\|_2] + E_{x} : p_{\text{data}}(x) [\left\| \nabla_v G_2(x,G_1(x)) \right\|_2] \\
\text{where } \nabla_h \text{ and } \nabla_v \text{ are horizontal and vertical gradient operators respectively.}
\]

The structural similarity index (SSIM) originally introduced in [22] to measure the similarity between two images. The multi-scale SSIM (MS-SSIM) [23] is a version of SSIM which is more flexible in variations of the image resolution or aspect angle. We incorporate MS-SSIM loss function to preserve luminance, contrast, and structural information of ground truth image in the produced image [23].

Finally, we adopt Wiener loss in our objective function introduced in [24]. The authors of [24] argue that using this loss function would reduce the color cast phenomenon in colorization task which is an unwanted tint of a particular color in the image. Consequently, it would make the output image closer to the ground truth image.

Therefore, the aggregate objective function for $G_2$ and $F_2$ is

\[
L_{\text{gen}}^{\text{gen}} = \gamma_1 L_{\text{GAN}}^{\text{gen}} + \gamma_2 L_{\text{cyc}} + \gamma_3 L_{\text{idt}} + \gamma_4 L_{\text{TV}} + \gamma_5 L_{\text{MS-SSIM}} + \gamma_6 L_{\text{Wiener}}
\]

where $\gamma_1$, $\gamma_2$, $\gamma_3$, $\gamma_4$, $\gamma_5$, $\gamma_6$ are constant parameters, and $L_{\text{MS-SSIM}}$ and $L_{\text{Wiener}}$ are objective functions corresponding to MS-SSIM and Wiener loss. The detail descriptions of these two objective functions are delivered in [23] and [24].

D. Network Architecture

Several network architectures are proposed in the literature. The encoder-decoder architecture which is a series of downsampling layers followed by upsampling layers is utilized in [1], and [9]. This structure suffers from the gradient disappearance problem, and also, downsampling layers might lose the spatial information exist in the image. The gradient disappearance problem is lightened in U-Net structure [13] by providing skip connection between layers of encoders and decoders. The U-Net structure is adopted in [8], and [10], but these models still suffer from losing spatial information. On the contrary, in order to preserve spatial information, [7] proposed fixed size layers and continuous concatenation of the input image with internal layers of the network. However, this model still seems to have issues with gradient flow.

Similar to [11], [12] and [18], we adopt residual structure networks proposed in [14]. The residual block structure enables the network to have several layers while preserving spatial information and back-propagating gradient without difficulties. This structure has shown great capability in the field of computer vision. In addition to using several residual blocks in our model, we incorporate the idea in the global structure of the second stage of our model so that the whole generator network $G_2: x \rightarrow y_2$ could be considered as a residual structure.

Here, we introduce our notation for describing our networks architectures. Let $Ck Sm BN R n$ denote a $k \times k$ Convolutional BatchNormalization-ReLU layer with stride $m$ and $n$ output filter. $Ck Sm BN LR n$ denotes the same structure with the leaky ReLU active function [16]. Notice that in this notation, $BN$ stands for Batch Normalization layer [17], and absence of
BN in the notation signifies its absence in the actual layer. \( R_n \) denotes a residual block, described in [14], with two 3x3 convolutional filters with \( n \) output filters at each layer. \( F_n \) denotes a fully connected layer with \( n \) output.

The network of \( G_1; x \rightarrow y_1 \), consists of:

\[
C7-S1-BN-R-64, C3-S2-BN-R-128, C3-S2-BN-R-256, R256, R256, R256, R256, R256, C3-S1/2-BN-R-128, C3-S1/2-BN-R-64, C7-S1-R-3
\]

The network of \( G_2; x, y_1 \rightarrow y_2 \) consists of:

\[
C7-S1-BN-LR-64, C3-S2-BN-LR-128, C3-S2-BN-LR-256, R256, R256, R256, R256, R256, R256, R256, C3-S1/2-LR-128, C3-S1/2-LR-64, C7-S1-R-3
\]

The networks of \( F_1; y_1 \rightarrow x' \) and \( F_2; y_2 \rightarrow x'' \) consist of:

\[
C7-S1-BN-R-64, C3-S2-BN-R-128, C3-S2-BN-R-256, R256, R256, R256, C3-S1/2-BN-R-128, C3-S1/2-BN-R-64, C7-S1-R-1
\]

The networks of \( D_1 \) and \( D_2 \) consist of:

\[
C3-S2-LR-32, C3-S2-BN-LR-64, C3-S2-BN-LR-128, C3-S2-BN-LR-256, C3-S2-BN-LR-512, F1
\]

III. EXPERIMENTAL RESULTS

We have trained our model on ImageNet object detection challenge dataset [25]. We randomly selected a subset from the dataset without any constraint on the category of the images so that the network would not be biased toward specific color tone.

In order to compare our results with the state-of-art methods, we also implemented methods provided in [7], [8] and [11]. Since the model in [11] is a generic image-to-image translation network, we modified its objective function and added L1 identity loss function to it in order to customize the objective function for the colorization task.

We fixed the input image size for all models to 200x200 pixels, and accordingly, all images in the dataset are cropped in the center at this size. The Adam optimizer [26] is used for the training process, and all models are trained using TensorFlow [27] and Keras [28] software packages accelerated on 1x Tesla K80 GPU Hardware.

The training process of our proposed method is organized as follows. First, we pre-train the generator and the discriminator of the first stage of our model. Then, the principal part of the training begins, where all networks are being trained simultaneously. Eventually, we pause the training process of all the networks except the generator of the second stage \( G_2 \), which is being fine-tuned at the final section of the training.

Notice that the generator \( G_2 \) should not be pre-trained with ground truth images at its input because the network would be biased to ignore the gray-scale image, which is the essential input to the network.

Colorization results of the implemented methods on the validation image set are presented in Fig. 2. One can observe that the J. Y. Zhu et al. [11] method outperforms the Y. Cao et al. [7] and P. Isola et al. [8] methods in producing vivid, realistic images. Furthermore, the results of our proposed method demonstrate that our model less suffers from the undesired artifacts related to J. Y. Zhu et al. [11] approach, while it is still capable of producing realistic images with vivid colors.

In order to evaluate our method quantitatively, we exploited PSNR, SSIM, and MS-SSIM, which are prominent metrics in the field of image quality assessment. We evaluated these three metrics on our validation image set, and the results are reported in Table I. Examining the outcomes reveals that our proposed method outperforms other colorization techniques in the literature so far, in terms of the above mentioned metrics.

IV. CONCLUSION

In this letter, we proposed a novel two-stage model based on the image-to-image translation architectures and incorporating a residual structure at the second stage to enhance the quality of the produced images. Colorization results on ImageNet challenge dataset demonstrate the superiority of our proposed model to other state-of-art approaches in creating realistic images close to ground truth ones.

For future work, we intend to investigate the effect of training the discriminator of the second stage with different datasets. Particular datasets may inject a general color theme, such as cold or warm tones, to the output images.

![Colorization results of different methods. (a) gray-scale image; (b) ground truth image; (c) Y. Cao et al. [7]; (d) P. Isola et al. [8]; (e) J. Y. Zhu et al. [11] with additional identity loss function; (f) our proposed method.](image-url)
REFERENCES


