Survey Report on Image Colorization

Namit Kumar	Abhinav Mishra	Ritik Agrawal	Saurabh Bazari	G Sharath Kumar
160101046	160101005	160101055	160101061	150101023

Abstract

Image colorization is an interesting topic in computer vision which takes in input an image without color and attempts to output a coloring scheme. Given input image can be different types, this paper survey mentions approaches taken for different types of input like grayscale, NIR and line sketch images to colorize into RGB or Lab format images. There are several approaches to solve these problems. Major problem trained and predict output based on CNN or GAN based model. Different types of GANs are used like conditional GAN, cycle GAN, asymmetric cycle GAN. We discuss the implementations and approaches of some paper. The limitations of the paper have also been observed based on their input or training condition.

Introduction

Colors make images more vivid and attractive to humans, it is also more easy to understand the details of the objects. There may be diverse colorful images possible for a grayscale image because so many colors have the same grayscale value, like the color of the car, the color of the shirt, etc. A human may intuitively guess the colors given a monochrome picture because we know what is in the picture and we have prior knowledge on how their colors should be. However, models lack the capability of guessing the appropriate color depending upon the context of the image provided, this particular field of filling color to a grayscale image is known as image colorization.

Image Colorization is ongoing research in the field of computer vision and is defined as the process of outputting a colorful image from a given input image which may be grayscale images, line sketch images, NIR images, etc. The primary goal of the problem is to produce a realistic 3-channel colorful image from the input image. The solution to the problem statement is ambiguous in nature as multiple colorization schemes are possible for a single grey-level image.

There have been different approaches to tackle the different scenarios related to the problem, some of which will be explored in a later section. The use of an encoder-decoder (VAE) architecture using CNNs to extract features and map these features to colorized image pixels has been largely applied to solve the problem of solving grayscale as well as NIR images.

GANs are a deep neural net architecture that allows a network to mimic internal structure as other data. Image generation is one of the major applications of GAN. GANs consists of two components: Generator and Discriminator. These two models are trained simultaneously. Discriminator’s task is to discriminate i.e differentiate between a real and a fake image generated by the Generator whereas Generator aims to generate an image that can be bypassed through the discriminator. This is an example of a min-max algorithm between these two.

Advantages of using Near Infrared(NIR):

The NIR imagery has an advantage over Visible spectrum imagery which suffers from lighting conditions and object surface's color. The NIR imagery can achieve clearer details than what can be achieved by using Visible Spectrum. The NIR spectrum range is just outside the human visibility range. The difference in NIR intensities also depends upon the absorption and reflection of dyes and not only on the particular color of the material.

Image colorization is useful in many scenarios like in the field of anime production, it is used to color the line sketch images. Another application of image colorization is to recover color in old photographs originally black and white in nature. Image colorization techniques are widely applied to predict colors in low light photographs. The method of colorizing images based on caption provided is used to fill and renovate old comics and movie frames. The photographs of outer space taken from telescopes are often colorized to generate a corresponding visually enhanced image. Thus we can conclude, the applications and use of image colorization are widespread which makes it an interesting research topic.

Example of image colorization from a grayscale image

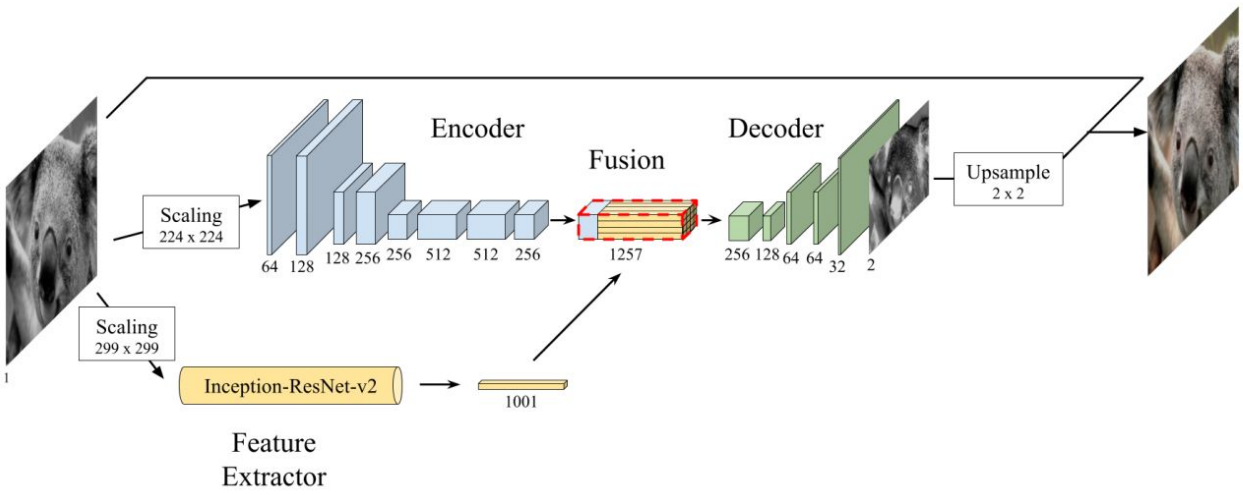
Literature Survey

Image Colorization using CNNs and Inception-Resnet-v2:

In this approach[1], we use the Convolutional Neural Network(CNN) model which was trained from scratch with a pre-trained Inception-ResNet-v2 model for high-level feature extraction. For input images, we chose CIE L*a*b* color space to separate luminance from color properties. The luminance can later be combined with the resultant components to produce the final colored image.

Architecture:

As input, we only have a grayscale image(containing only luminance component) and our model predicts two components: a*b* and by combining it with input we obtain the output containing all three components.
The model is structured into four major parts. Firstly, the encoding component obtains mid-level features. Secondly, a feature extraction component is used to obtain high-level features from the input. Thirdly, these features are merged by the fusion component to give the input to the decoder component which finally processes these features to predict the output. Below is the detailed explanation of each component:



Overview of the Feature-Extraction Model Architecture

Encoder:

The Encoder takes input grayscale image as H(height)* W(width) and outputs an H/8 * W/8* 512 features() representation. Padding is used to maintain the layer's input size and a kernel size of 3*3 for each convolutional layer. Downsampling is applied at first, third and fifth layers with stride 2 to reduce the number of computations required.

Feature Extractor:

This component takes an input image of 299*299 dimensions and outputs a 1001*1*1 embedding. This component identifies high-level features like "vegetation", "person" etc. The pre-trained Inception model is used to extract an image embedding.

Fusion:

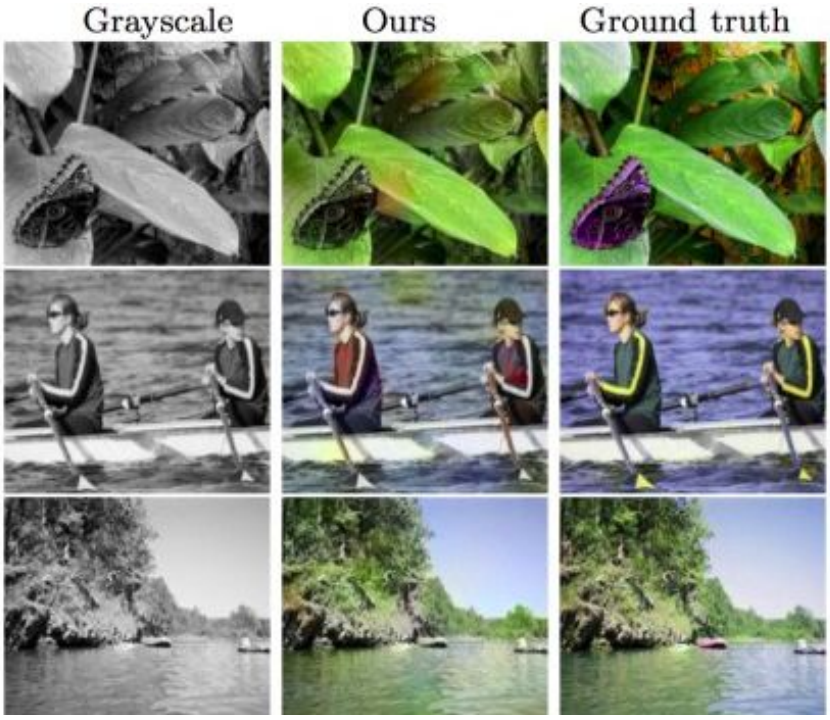
This component as the name suggests merges the features extracted from the above components. To convey the feature information to every region of the image(output of the encoder) feature vector(output of the Feature Extractor) is replicated H*W/(8^2) times. This generates the output of dimension H/8*W/8*256 by applying 256 convolutional kernels of size 1*1.

Decoder:

Finally, this component takes input from Fusion component and generates an output of dimension H*W*2 by applying Upsampling at first, third and fifth layer.

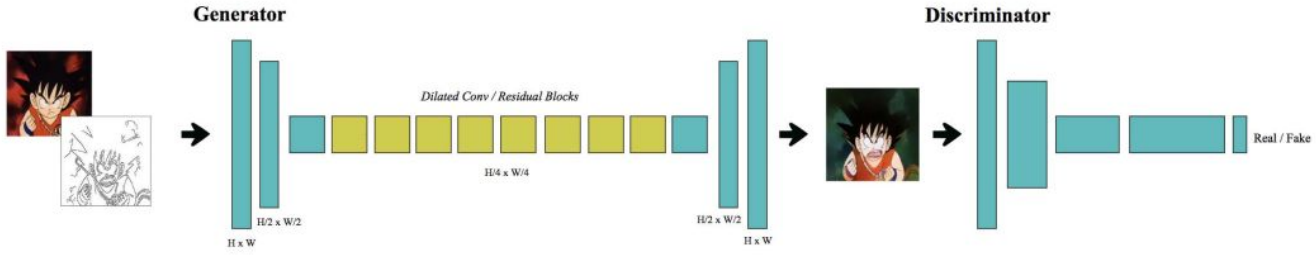
The loss function used Mean Square Error between the estimated output and targeted output pixels.

As given results explain that the last grayscale image colored better than the first two. In first example model did not the color butterfly. In the second example model colored shirt differently because it depends on how our model is trained and in the last example model gives a photo-realistic image.



Automatic Temporally Coherent Video Colorization:

This model proposed in this paper[2] takes input in the form of synchronized grayscale images and converts them to colorized images. This can reduce costs for anime production studio significantly while speeding up the animation process. Using a U-net based generator and patch-based discriminator, a high dimensional input is mapped to a high dimensional output. But it is limited because of Downsampling in encoding and Upsampling in decoding of an input image. This is because the input data may already be sparse and Downsampling on such data will result in data loss. The generator gives the color prediction by taking a greyscale or line art image as input conditioned on the previous color frame. Various losses are combined as a joint loss to train our model. The losses are Adversarial Loss, Style Loss, Content Loss, L1 Loss. The Adversarial Loss includes a previous predicted image to encourage temporal consistency. The perceptual similarity between predicted and ground truth color frames is maintained by Content Loss while Texture similarities between predicted and ground truth color frames are maintained by Style Loss.



Architecture of model

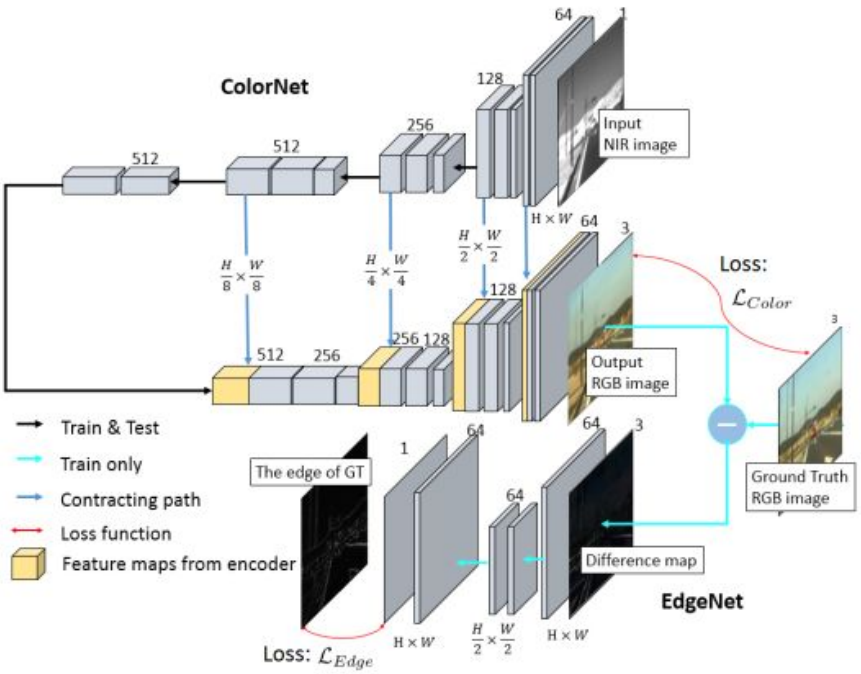
Inputs for the discriminator are Current and previous input, generated image and ground truth image. Initially i.e, in the absence of the previous frame so the blank image is passed as an input to the generator. According to Bernoulli distribution, either the blank image or the corresponding previous color frame is sampled.
To train the model we take a dataset of colorful images. This dataset is converted into two sets: grayscale and line art images. This model is trained on both of the sets. We use Canny edge detection with a Gaussian filter of standard deviation 1 for mimicking line art. The learning rate of the generator is set higher than the discriminator to provide an opportunity for the generator to learn a mapping before the discriminator.
The Style Loss and Content Loss gives us the advantage to minimize the flicker effect and checkerboard effect.

(a) Greyscale (b) Baseline (c) Ours (a) Line art (b) Baseline (c) Ours

By manipulating the loss functions in the baseline, so our model gives better results than the baseline model.

Infrared Image Colourization Using an S-Shape Network:

This paper[3] tackles the problem of colorizing Near-Infrared Images (NIR) without any reference image by proposing an “S-shaped network” which is a combination of two encoder-decoder architecture trained in an end to end fashion to increase performance. The first encoder-decoder architecture ColorNet extracts the context and feature from the image and transfer it into an RGB representation. The second encoder-decoder architecture also called the assistant network or the EdgeNet enhances the edges to produce better output, further detailed explanations about the architecture and working are given below.



ColorNet:

The ColorNet is responsible for taking an NIR image as input and output a corresponding RGB representation. The encoder of this network consists of 5 convolutional blocks with each block having kernel size 3X3 followed by a Batch Normalization Layer and ReLu as an activation function. After each block, a max-pooling factor of 2 is applied to perform downsampling. The encoder finally outputs a 512X7X7 dimensional feature representation of the NIR image. The decoder consists of 4 convolutional blocks with each block having the same kernel size, activation function and Batch Normalization Layer as the encoder. The last layer, however, is followed by a 1X1 convolution with a tanh() activation layer. The decoder finally outputs a 3 channel RGB image. The L2 loss function is used to train the ColorNet network.

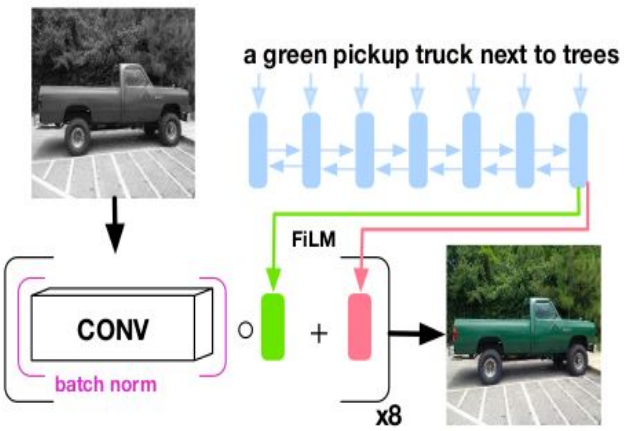
EdgeNet:

The EdgeNet improves the clearness of edges and relearns the color of the region in the ground truth taking the output of the ColorNet as input and outputs a much clearer RGB representation of the original NIR image. The encoder of this network consists of 2 layers with each layer using a convolution of stride 2 to perform downsizing. The decoder of this network consists of 2 layers with each layer performing transposed convolution of stride 2 for up-sampling. The EdgeNet network also uses the L2 loss function.

The overall end to end training is performed to better performance by defining the overall loss function as the sum of the losses from the two networks.

Learning To Color From Language:

This paper[4] conditions the process of automatic colorization of grayscale images on language, allowing users to manipulate the output colorized images by feeding in different captions. 2 different architectures, FILM and CONCAT have been proposed for language-conditioned colorization. Three channels can be used to create a colorized image. The first channel being the luminance (L) channel, and 2 other channels a and b representing green to red and blue to yellow respectively. Input to the model will be the L channel(from the greyscale image) and a caption to the image using which we have to predict the a and b channels to colorize the image.



Model:

The base-line network architecture is FCNN which has 8 convolution blocks each containing many convolutional layers. FCNN does not use the caption to colorize the image. We improve on this model to include the caption as an input. The input text is encoded and integrated with every convolutional block of FCNN. This helps the model to color the specific features of the image based on the caption. This model is called CONCAT. By including the text encoding, the number of parameters of the model has increased. Another similar model proposed is feature-wise linear modulation (FILM) which has lesser parameters compared to CONCAT.

Ground-truth image | FCNN output | FILM conditioned on "a school bus driving on the road"

Based on experiments performed it is found that FILM and CONCAT give more or less similar results.
A major advantage of including the caption for image colorization is that some objects always have a specific color, for example, a school bus is always yellow. This helps the model to colorize the image more accurately. It can be seen below that FCNN not having the caption is unable to color the school bus.

The results for the 3 models on different experiments can be seen in the below table. 3 human experiments performed are to check plausibility, quality, and measure of how changing the caption change the output.

	ab Accuracy		Human Experiments		
Model	acc@1	acc@5	plaus.	qual.	manip.
FCNN	15.4	45.8	20.4	32.6	N/A
CONCAT	17.9	50.3	39.0	**34.1**	77.4
FILM	**23.7**	**60.5**	**40.6**	32.1	**81.2**

NIR to RGB Domain Translation Using Asymmetric Cycle Generative Adversarial Networks:

This paper[5] aims at colorizing NIR(Near Infrared) Images. Some differences between NIR and grayscale images have been mentioned like the luminance of grayscale and RGB images are identical whereas since the light source is different for NIR, we get different luminance from RGB. GANs are used to get realistic image

colorization. A variation of GAN is a cycle GAN which is used when we have unregistered data. Instead of training a single pair of generators and discriminator, cycle GAN trains 2 pairs. The generators are inverse of each other whereas the 2 discriminators discriminate between the 2 different types of images. One of the generators aims to convert image A to image B whereas the inverse convert image B to image A. To get realistic colorization and deal with unregistered data, a cycle GAN based approach is proposed. Since an RGB image has more information than a NIR image, RGB to NIR conversion is easier compared to NIR to RGB. Therefore an asymmetric cycle GAN to deal with the 2-way conversion. UNet and ResNet are combined in the generator to increase the depth and FPN(Feature Pyramid Network) is used in discriminator to capture spatial context information.

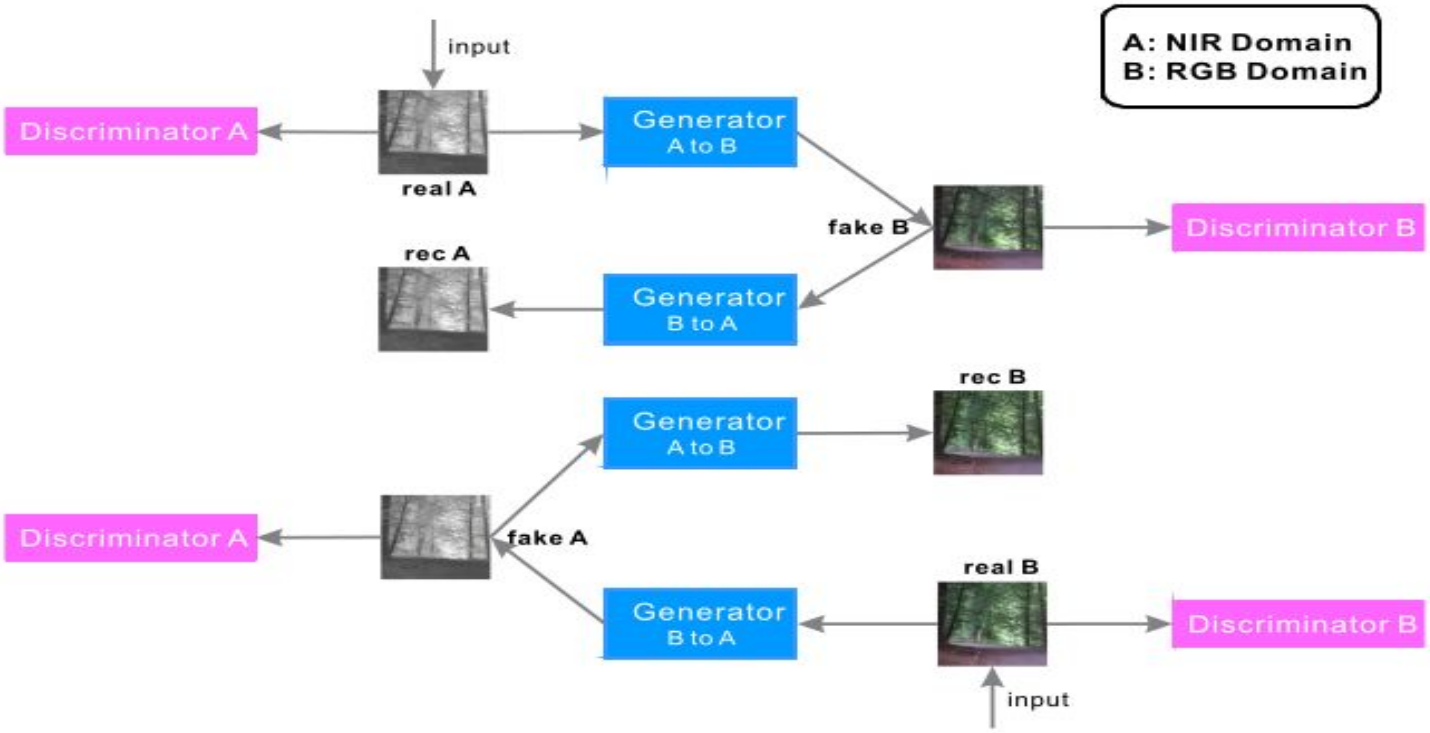


FIGURE 1. NIR to RGB domain translation using asymmetric cycle GAN, redrawn from [14]. Single directional model often calculates L_1 norm *L*(*fake B*, *real B*) as training loss. When *real A* is not registered with *real B*, the training loss is confused by unregistration but the cyclic loss *L*(*real A*, *F*(*G*(*real A*))) is unaffected by unregistration.

Model:

- Asymmetric Generator - UNet modules are used in both the generators as it helps to preserve low level and high-resolution features. For RGB to NIR conversion, only UNet is used but for NIR to RGB conversion, ResNet blocks are added to capture high-level features.
- Asymmetric Discriminator - Again since RGB to NIR conversion is easy, only CNN is enough for the discriminator. But for discriminator for NIR to RGB conversion FPN is used which is effective to identify objects separately and hence good for NIR Colorization.

The following loss function is used for training the model -

$$Total\_loss = \min_G \max_D \{L_{GAN}(G_{AB}, D_B, A, B) + \lambda_1 L_{GAN}(G_{BA}, D_A, A, B)$$

$$+ \lambda_2 L_{rec}(G_{AB}, G_{BA}, A, B)\},$$

A - NIR Image, B - RGB Image
The 3 types of the loss function that can be seen correspond to Loss for NIR to RGB conversion, RGB to NIR conversion and unregistered data scenario, respectively. The contribution of each of the losses are based on the coefficient parameters. Batch Normalization is used since these models are difficult to train.

The proposed method gave better results as compared to other implementations of GAN on experimentation. Models were tested on different categories of images like urban, old buildings, etc. The results can be seen in the below table -
AE - AutoEncoder (Smaller is better)
SSIM - Structural Loss

	AE		SSIM	
	Urban	*Old-Building*	*Urban*	*Old-Building*
Conditional GAN [31]	5.77	5.96	0.84	0.86
Stacked Conditional GAN [33]	5.04	4.78	0.90	0.91
UNIT [34]	16.02	14.65	0.52	0.52
Cycle GAN [14]	8.41	8.15	0.81	0.83
Proposed Method	5.05	5.30	0.90	0.89

Near-Infrared Imagery Colorization:

In this paper[6], a Stack Conditioned GAN(SC-GAN) has been proposed for NIR image colorization. A variant of GAN architecture has been used which uses multiple loss functions over a conditional probabilistic model. The generator network has been modified to use feature hierarchical representation. Gaussian noise has been fused with the input to ensure more diversity.

Architecture:

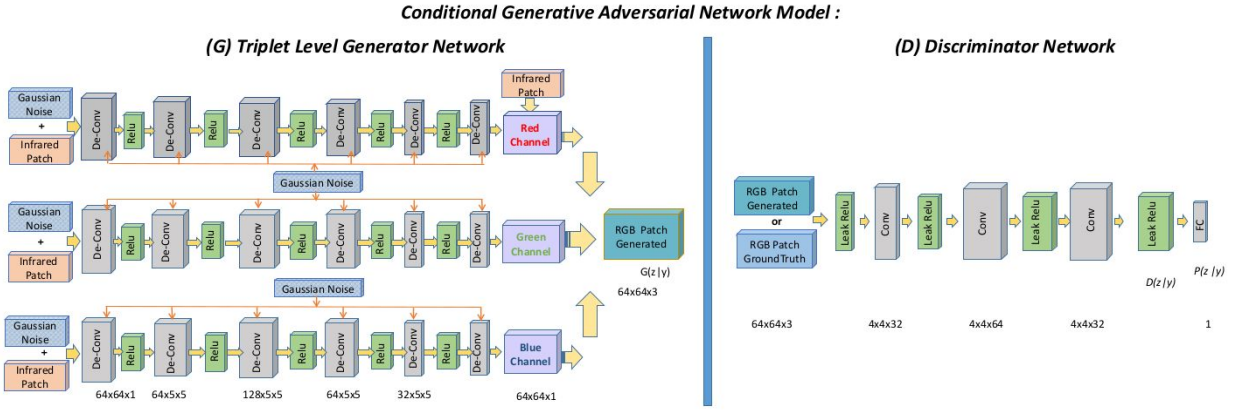


Fig. 1. Illustration of the proposed triplet GAN architecture used for NIR image colorization.

A feature hierarchy has been added to each layer during the learning process encouraging the representation manifold of the generator to align with the discriminative network's bottom. As can be seen from Fig. 1 Gaussian noise has been included in every layer of the triplet architecture in the generator model for the generalization to optimize the learning process. SC-GAN was used for the following reasons:

- To condition the learning on NIR imagery and Gaussian noise from the source.
- It reaches the convergence faster as compared to other architectures.
- Sampling becomes simple and efficient.
- High-level features from the generator are optimized.

So, in a nutshell, the generative model G is trained from a NIR image plus the Gaussian noise-producing an RGB image and the discriminator model D is trained to assign the correct label to the generated image based on the ground truth images.
Various loss functions have been combined to produce the final loss function(**L *final***):

- **Adversarial Loss**:

  It is designed to minimize the cross-entropy improving the texture loss

  $$\mathcal{L}_{Adversarial} = -\sum_{i} logD(G_w(I_{z|y}), (I_{x|y}),$$

  Where D and G*w* are the discriminator and generator. X|y is the real while z|y is generated.

- **Intensity Loss(MSE):**

  $$\mathcal{L}_{Intensity} = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(RGBe_{i,j} - RGBg_{i,j})^2,$$

  This loss measures the intensity difference between the pixels without considering the texture and content comparisons. RGBe(i, j) is the estimated RGB representation whereas RGBg(i, j) is the ground-truth RGB image.

- **Structural Loss(SSIM):**

  $$\mathcal{L}_{SSIM} = \frac{1}{NM}\sum_{p=1}^{P} 1 - SSIM(p),$$

  SSIM(p) is defined for a pixel which is the Structural Similarity Index centered in pixel *p* of the patch(P).

  Finally, **Final Loss** is represented as follows:

  $$\mathcal{L}_{final} = 0.65\mathcal{L}_{Adversarial} + 0.2\mathcal{L}_{Intensity} + 0.15\mathcal{L}_{SSIM}.$$

Limitations

In [1], Nevertheless, the performance in coloring small details are still to be improved. As we only used a reduced subset of ImageNet, only a small portion of the spectrum of possible subjects is represented, therefore, the performance on unseen images highly depends on their specific contents. To overcome this issue, our network should be trained over a larger training dataset. In [2], This model also suffers a limitation due to the incorrect colorization of an inside frame as it will corrupt the next frame. The model faces problems in case of the addition of new characters. To solve this we have to retrain our model with colorized frames adjusted by artists for new characters. Sometimes information such as dots or lines in scenes is missed by Canny edge detection. In [4], The model gets biased against unnatural colorization like a yellow sky. Also, a smaller object is problematic due to the color leak. Limitation of [4] could be removed by using a better edge detecting model of [3] so that color leak is lesser in smaller objects.

Problem Statement

With an increase in work on colorizing black and white color, line sketches and NIR image because of applications in various fields including anime generation, generation of colorful photographs from old ones, space image, etc. the topic of image colorization has become an interesting topic of research in the field of computer vision. Our problem statement is to generate an image with realistic color by training it on a given set of images with corresponding ground truth color or a reference condition. The model is not expected to always output the ground truth but the output image is expected to be realistic enough so that any human finds it difficult to differentiate between our output image and the ground truth which is real.

References

1. Deep Koalarization: Image Colorization using CNNs and Inception-Resnet-v2, Federico Baldassarre, Diego Gonz´alez Mor´ın, Lucas Rod´es-Guirao, KTH Royal Institute of Technology. https://arxiv.org/pdf/1712.03400v1.pdf

2. Automatic Temporally Coherent Video Colorization, Harrish Thasarathan, Kamyar Nazeri, Mehran Ebrahimi, Imaging Lab, Faculty of Science University of Ontario Institute of Technology, Canada. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8781608

3. INFRARED IMAGE COLORIZATION USING A S-SHAPE NETWORK, Ziyue Dong, Sei-Ichiro Kamata, Toby P.Breckon, Graduate School of Information, Production, and Systems, Waseda University. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8451230

4. Learning to Color from Language, Varun Manjunatha Mohit Iyyer Jordan Boyd-Graber Larry Davis University of Maryland: Computer Science, Language Science, iSchool, UMIACS. https://arxiv.org/pdf/1804.06026v1.pdf

5. NIR to RGB Domain Translation Using Asymmetric Cycle Generative Adversarial Networks, TIAN SUN, CHEOLKON JUNG, (Member, IEEE), QINGTAO FU, AND QIHUI HAN, School of Electronic Engineering, Xidian University, Xi'an 710071, China.
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8790680

6. Near InfraRed Imagery Colorization Patricia L. Suarez, Angel D. Sappa, Boris X. Vintimilla and Riad I. Hammoud, Escuela Superior Politécnica del Litoral, ESPOL, Guayaquil, Ecuador, Computer Vision Center, Campus UAB, Bellaterra, Barcelona, Spain, BAE Systems FAST Labs, Burlington, MA 01803, USA
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8451413