Image Colorization

Imanklayan Sarkar
MT20010

Soumam Banerjee
MT20043

Shradha Agarwal
MT20123

January 5, 2022

1 Abstract

Given a grayscale image , the proposed model computes a colorful image that is plausible in nature. Traditional image colorization techniques based on Deep CNN methods generally tend to capture the semantics due to which issues like semantic confusion occur thus leading to color bleeding. Considering all such problems we have proposed a multi-path GAN based colorization network guided by edge and saliency maps. The proposed architecture not only reduces bleeding but also generates considerable outputs which are at par with the pre-existing GAN models based on saliency given limited computational resources. We have used pre-trained Resnet-18 architecture to generate image feature maps from the input grayscale images,which is stacked with one of the blocks of the encoder. This leverages us to work on very less data as compared to other traditional models. Besides the original image , Saliency maps are also provided as a target to be learned by the network to reduce bleeding. To improve the performance of the generator , we have used a Patch GAN based discriminator for guided colorization. Our proposed model is trained on 5,000 images of Flickr dataset. Our results show that the model can generate reasonable colorized images.

keywords - *Resnet, Saliency map , edge map, Patch GAN* discriminator

2 Introduction

Image colorization is a task where colors are added to a given gray-scale image to make it more beautiful and more aesthetic to the human eye. Image colorization is a sophisticated task as the shades of the color may vary from image to image. Colors may also be determined by the presence of an object or the environment of the image. Previously the images were mainly colored by experts. Since an image may contain numerous color shades the task of manual colorization was sophisticated as well as time-consuming. with the advancement of technology, other methods of colorization like scribble-based coloring came up. In the scribble-based method, colors were given to adjacent pixels based on some low-level similarity features. This method could give some satisfactory result, but only if some expert guidance were given, i.e it contained human intervention and was not fully automatic. With the advancement in the field of machine learning, example-based colorization came up where colors were transferred from an example image to the target image. the downside of this method was that it was pretty hard to find a suitable reference image to produce a satisfactory result. So the above methods either needed human intervention or it was hard to find a reference.

3 Related Work

Earlier methods of image colorization involved non parametric methods where user gives a gray image and some reference color image which was used to generate color outputs.
Zang et al. [2] proposed a deep CNN architecture to predict the probability distribution of each pixel for 313 quantised AB values in LAB space. They have used multinomial cross entropy with class rebalancing term to handle multimodal uncertainty and colour class rebalancing problem. A score of 32.3 is achieved in Perceptual realism test.
Cao et al. [3] proposed conditional GAN with fully convolutional generator. They have used multi-layer noise and condition information for diversity enhancement and to maintain realism. A score of 80 is achieved on the Turing test depicting their colours were convincing.
Manjunatha et.al [4] proposed the colorization task and manipulating it by feeding different captions. They used feature wise affine transformation on the output of each convolutional layer. The weights were conditioned on the language features.
Nazeri et al [5] used DCGAN based on UNet [1] architecture with alternating cost function to predict the a,b channels.One sided label smoothing is done to prevent the network to produce strictly 0 and 1 output. MSE error of 5.1 is achieved on CIFAR-10 dataset.
Jheng et al [6] extracted features for each object using MaskRCNN which are fused with the features of the image. Smooth L1 loss function is used. PSNR score achieved is 27.562.
Zhao et al. [7] proposed SCGAN to jointly predict colorization and saliency map. The generator comprised of Global feature network, attention prediction network and main colorization network. Two PatchGAN discriminators are used. They worked on the problem of colour bleeding and semantic confusion. PSNR score achieved is 23.80.

4 Methodology

Data Pre-processing: Grayscale images are extracted from the training samples in RGB space and are resized to 64 X 64. Also, we have used Canny edge detection to create the edge maps of the images. Saliency maps are generated for the RGB training samples. Moreover, all the images are normalised to the range of -1 and 1. Finally, the data is shuffled and divided into batches of size 64. The grayscale image and the corresponding edge map are stacked together which is inputted to the generator.

Input: The input to the generator is a two-channel input of size 64 X 64, with grayscale image on one channel and the edge map of the image on the other.
Output: The output of the generator is a four-channel output of size 64 X 64, with RGB image on the first three channels and the saliency map on the last channel.
Architecture of the generator: It consists of the colorization network and the pre-trained ResNet-18 model. The first part of the generator has seven down-sampling and seven up-sampling layers. The corresponding layers of the encoder and the decoder layers are connected via skip connections. The activation function used is Leaky ReLU for all the layers and Tanh for the output layer. Also, feature extraction is done from the pre-trained ResNet-18 model. All the layers till the last block are considered and the network weights are frozen. The output from this model is 512 X 7 X 7 which is resized to 512 X 2 X2 for the purpose of concatenating it with the sixth down-sampling layer of the generator. Architecture of the discriminator: It is similar to that of PatchGAN discriminator. There are six convolutional-2D layers with LeakyReLU as the activation function. For the output layer, sigmoid activation is used. The output of the discriminator is of size of 1 X 14 X 14. Each value in this output is the probability of the pixel being real or fake.
Discriminator Loss: The loss of the discriminator is the difference between discriminator loss on the fake RGB image; i.e., the image generated by the generator; and the discriminator loss on the real RGB image.

$$\text{Loss\_Discriminator} = E[D(G(x_{gray})_{RGB})] - E[D(x_{RGB})]$$

Here, x is the input image, G is the generator, D is the discriminator and E is the expectation.

Generator Loss: The loss of the generator consists of three types of losses, namely, L1 loss for the RGB image, L1 loss for the weighted saliency with RGB image and the GAN loss. L1 loss for the RGB image is calculated between the real RGB image and the corresponding fake RGB image generated by the generator. The generated/ fake saliency map is multiplied with the generated/ fake RGB image to calculate the weighted saliency with RGB image. For calculating the GAN loss, the RGB image output of the generator is stacked (channel-wise) with the input grayscale image and passed through the discriminator. The negative of the output of the discriminator is then averaged to form the GAN loss. The weighted sum of these three loss results in the loss of the generator. The weights on which the model is run are 1, 0.05 and 0.5 for L1 loss for the RGB image, L1 loss for the weighted saliency with RGB image and the GAN loss respectively.

$$LossGenerator = L1\_loss + \lambda_G * GAN\_Loss + \lambda_{sal} * saliency\_loss$$

Here, λ_G, λ_{sal}, GAN_Loss are the weights for generator loss and saliency loss; and the GAN loss respectively.

Xavier initialisation is utilised for weight initialisation. Adam optimiser is used with learning rates of the generator and the discriminator as 1 X 10^{-4} and 1 X 10^{-5} respectively.

5 Dataset, Baselines, and Results

Flickr30k images dataset has been used for training our model. Due to limited GPU resource of colab we could train upon only 5000 images where as our original baseline models have been implemented on 1.3 lakhs images . Even though we trained on less data we validated it on further 5000 images of Flickr30k dataset. Besides, we also created a test dataset of our own to verify if our model has overfitted on the image categories of the training dataset. We found equally plausible colorized images across various categories of our own created test dataset.

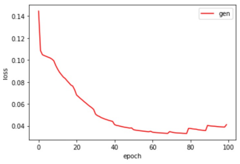

Figure 1: Training loss

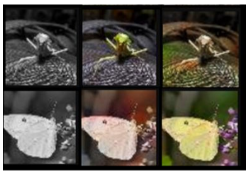

Figure 2: Gray(L), Fake(M),Real(R)

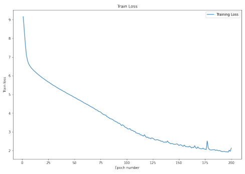

Figure 3: Training loss

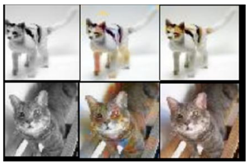

Figure 4: Gray(L), Fake(M), Real(R)

5.1 Baseline 1: reference [7]

Architecture Generator: It is similar to "Main Colorization Network".Seven down-sampling layers, followed by seven up-sampling layers of Conv2D with Leaky ReLU (tanh for the last layer) is used. Skip connections are used to help faster convergence. The loss function is the difference between L1 loss and mean of fake image * $\lambda(0.05)$.
Architecture Discriminator: It is similar to "PatchGAN discriminator". Six Conv2D layers with Leaky ReLU (sigmoid for the last layer) is used. The input is either the concatenation of gray image and fake BGR image or true BGR image. The output is one channel with size 30*30. The loss function is the difference between the mean of true scalar and fake scalar.
Implementation details: Dataset: tiny ImageNet 200; Generator lr (η_G) : 2e-4; Discriminator lr(η_D): 1e-3 Optimisers: Adam; Epochs: 100
Observations – Considering Colab's RAM, GPU,disk space constraints our model could learn colorization and could generate plausible colored outputs.(Figure 2). At the end of 100^{th} epoch, the loss was 0.04. (Figure 1).

5.2 Baseline 2: reference [2]

The input to the model is the normalised gray scale image and the output is the probability distribution for each pixel in the AB space spread over 313 quantised values. Soft encoding

is done to form the training output data.
Architecture:There are five blocks consisting two or three convolutional layers with ReLU activation function followed by batch normalisation layer. Up sampling is done after the fourth block. The output layer uses softmax activation function over 313 quantised number of channels.
Implementation details: Dataset: tiny ImageNet 200 dataset; Loss function: multinomial cross entropy loss; optimiser: NAG; learning rate: 0.01; epochs: 200.
Observations: The training loss decreased with epochs and at the end of 200 epochs, the loss was 2.2. From figure 3, it can be observed that the model could learn to predict the appropriate colours within small number of epochs.

As can be seen above, Fig 2. and Fig 4. are the results of our respective baselines of colorful image colorization by Zhang et al. and SCGAN model by Zhao et al. In Fig2 it can be easily observed that color bleeding occurs, as there is semantic confusion in color when we use Deep CNN based approaches. Alhough, it has been able to colorize the images upto an extent but its not comparable to current state of the art. Moreover the architecture used in Colorful image colorization (Baseline 2) was very heavy and needed humongous amount of training data as it is not aided with any other pre-trained network which makes it infeasible to work in limited computational constraints like ours. In Fig 4, even though SCGAN model seems to work comparatively better than our baseline 2 but it tends to produce results on very limited category of images . In our case, baseline 1 was trained on tiny image net data that consists of 100 categories(50k images) but due to our limited computational resources we could train on only 8 categories of images (4000 images), due to which our model failed badly across varying categories other than the particular categories it has been trained upon. For Fig 1. even though our training loss seemed to decrease but it is purely over-fitting and hence failed on totally unseen dataset, where as for SCGAN model the loss of generator decreased surely but it only worked on limited categories.Thus, observing the potential of SCGAN model result we decided to build our modifications on baseline 1 to improve it further. .

6 Analysis

6.1 Result

In comparison to our above 2 baseline models our model is able to generate better plausible colorful images across varying categories. Out of all categories, we have cherry picked few of the best colorful images and few of the failure cases. Noticing such promising results on test data set of Flickr we created our own dataset to check its performance on a dataset having a little different data distribution based on categories, but we found positive results along with few failure cases. Successful results generally varied across categories ranging from human skin, skies, grass, dogs, water bodies whereas failure cases were mainly seen across categories having multiple plausible color possibilities like house color,cars, jersey of

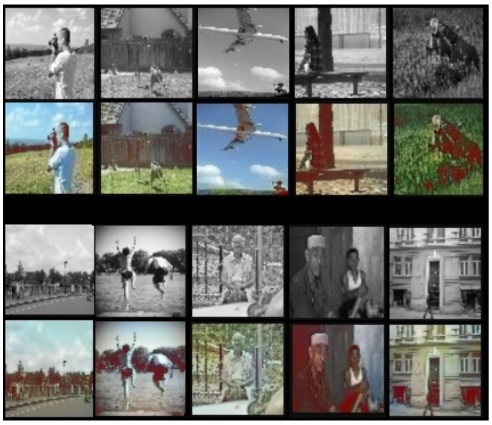

Figure 5: Flickr test set

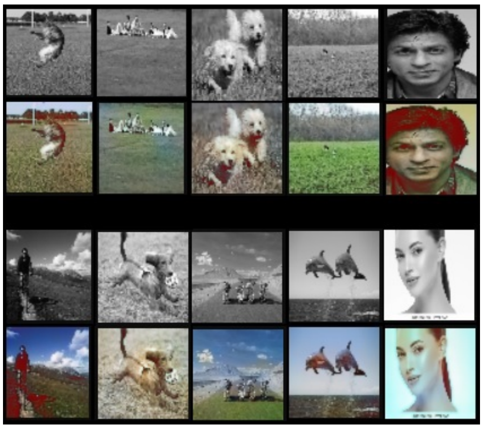

Figure 6: Random Test Data

team players and vintage photos.
Also, the random test data is evaluated on the PSNR metric. The number of samples considered for the random test data is fifty-four. For these samples, we achieved a PSNR score of 15.63. The PSNR score obtained in [7] is 23.80. The PSNR score obtained by our approach cannot be compared with the PSNR score obtained in the [7] because the dataset and the number of samples considered is very different.

6.2 Error

There is substantial amount of error present which is evident considering our limited RAM and GPU constraints. We find our model failed to produce substantial colors on images having multiple possibilities of color. It also fails on old black and white images and images having multiple color variance in a particular image like in case of peacock and rainbow. Model was trained on 12 GB RAM and Tesla K80 GPU provided by Google Colab. As Resnet is a very deep model we didn't fine tune the loss of the pre-trained model and used it only to generate image features, which is not the case in general, for multi-path networks. Also, our training dataset comprised of only 5000 images where as SCGAN model is trained on 1.3 lakhs images which is one of the primary reason we couldn't generate such comparable colorized images across all categories. Moreover, due to limited usage time of colab's GPU we could train only on 20 epochs at a time. Since the model is GAN based which is very difficult to set the perfect hyper-parameters, it was highly challenging for us to work on a very few combinations of hyper-parameters.

6.3 Visualisation

Given above in Fig. 7 and Fig. 8, represents individual losses for patch GAN discriminator and Edge cum saliency aided multi-patch generator. It can be observed that our overall generator loss has decreased with epochs and started to stagnate after 50 epochs due to less data.

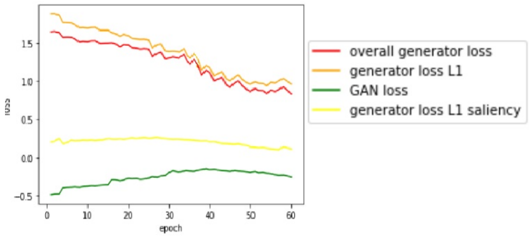

Figure 7: Generator loss

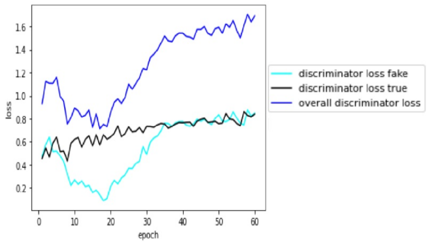

Figure 8: Discriminator loss

As we know there is a tug-of-war coming to discriminator loss and our generator tries to keep it in equilibrium . Even though our overall discriminator loss has not been successfully able to achieve the equilibrium but is fluctuates around the value 0, which can arise due to 2 reason , firstly our discriminator labels both the real and fake images as fake and secondly the ideal condition i.e. it marks true image as true and fake image as also true considering our generator is able to generate images from distribution that is identical to our True target distribution from the latent input space.

7 Individual Member Contribution

Shradha Agarwal: Baseline-1 implementation, Baseline-2 implementation, Final implementation, Architecture of the novel approach
Soumam Banerjee:Baseline-1 implementation, baseline-2 implementation, final implementation, idea of making GUI for the project, loss functions of the novel approach
Imankalyan Sarkar: Baseline-1 implementation, baseline-2 implementation, final implementation, creation of random test data, literature survey, find appropriate datasets

8 Conclusion

In this report ,we proposed a multipath GAN based image colrization architecture guided by saliency and edge maps. It is able to generate perceaptually plausible images especially for landscape images containing sky, waterbodies, grounds, dogs, human-skin, and few more categories. We also suffered few failure cases as mentioned above considering our limited computational constraints. We were also able to generate saliency maps as a byproduct from the given edge and grayscale input images automatically .Credit also goes to the pre-trained resnet-architecture which helped the model to learn colorful features for the given input gray image, thus helping us to achieve our target with very limited dataset. The addition of edge based input along with grayscale image has been inferred from User-Guided Deep Anime Line Art Colorization with Conditional Adversarial Networks paper, which helped us to considerably improve our result than our baseline SCGAN model across various categories of objects.

References

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: `1505.04597 [cs.CV]`.

[2] Richard Zhang, Phillip Isola, and Alexei A. Efros. *Colorful Image Colorization*. 2016. arXiv: `1603.08511 [cs.CV]`.

[3] Yun Cao et al. *Unsupervised Diverse Colorization via Generative Adversarial Networks*. 2017. arXiv: `1702.06674 [cs.CV]`.

[4] Varun Manjunatha et al. *Learning to Color from Language*. 2018. arXiv: `1804.06026 [cs.CV]`.

[5] Kamyar Nazeri, Eric Ng, and Mehran Ebrahimi. “Image Colorization Using Generative Adversarial Networks”. In: *Lecture Notes in Computer Science* (2018), pp. 85–94. ISSN: 1611-3349.

[6] Jheng-Wei Su, Hung-Kuo Chu, and Jia-Bin Huang. *Instance-aware Image Colorization*. 2020. arXiv: `2005.10825 [cs.CV]`.

[7] Yuzhi Zhao et al. “SCGAN: Saliency Map-guided Colorization with Generative Adversarial Network”. In: *IEEE Transactions on Circuits and Systems for Video Technology* (2020), pp. 1–1. ISSN: 1558-2205.