e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:04/April-2023 Impact Factor- 7.868 www.irjmets.com

COLORIZING BLACK AND WHITE IMAGES USING CONDITIONAL GAN

T. Sai Srinivas[*1], Vemuri Harshitha[*2], Balabolu Harshitha[*3],
Ms. Mansa Devi Pappu[*4]

[*1,2,3]Dept. Of Computer Science And Engineering GITAM (Deemed To Be University) Visakhapatnam, India.

[*4]Assistant Professor, Dept. Of Computer Science And Engineering GITAM (Deemed To Be University) Visakhapatnam, India.

DOI : https://www.doi.org/10.56726/IRJMETS35935

ABSTRACT

Image colorization is an essential task in computer vision and various Industries where there is work related to images, such as in the manga(comic book) industry that involves adding color to grayscale images. In previous years, deep learning techniques have been used to solve this problem, and among them, conditional generative adversarial networks (CGANs) and U-Net architectures have shown promising results.

In this project, we propose an optimized approach for image colorization using a combination of CGAN and U-Net. We first train a CGAN to generate color images conditioned on the corresponding grayscale images. Then, we use the generator network of the trained CGAN as an encoder in a U-Net architecture. The decoder network of the U-Net is then trained to generate the color image from the encoded features.

The experimentation's findings demonstrate that, when applied to benchmark datasets, the suggested strategy outperforms current state-of-the-art approaches. The proposed approach produces visually appealing and realistic color images with improved accuracy and efficiency. The combination of CGAN and U-Net provides a robust and flexible framework for image colorization that can be extended to various other image processing tasks.

Keywords: Image Processing, Deep Learning, Computer Vision, Colorization, Convolutional Neural Networks (CNNS), Generative Adversarial Networks (GANS), Conditional GANS (CGANS).

I. INTRODUCTION

Image colorization has many practical applications, such as in the restoration of historical photographs or in medical imaging, where adding color can aid in identifying and diagnosing diseases. Additionally, it can be used in the entertainment industry for colorizing black-and-white films and in the design industry for creating color variations of products or prototypes.

The process of image colorization can be performed manually by an artist, but this can be time-consuming and costly. Therefore, many automatic methods are proposed to solve this problem. Deep learning techniques, in particular, have shown significant progress in recent years, achieving impressive results in image colorization.

The goal of image colorization is to produce a color image that is consistent with the original grayscale image while also being visually appealing and realistic. Various deep learning architectures have been proposed for this task, including generative adversarial networks (GANs) and U-NET, which is used as an architecture for semantic segmentation. These architectures learn to generate color images by training on a dataset of grayscale and color images.

Overall, image colorization is an essential area of research in computer vision, with many potential applications. The development of accurate and efficient automatic image colorization can transform many fields, from entertainment to medicine and beyond.

Recommender systems are becoming increasingly popular in various industries, such as e-commerce, online advertising, and entertainment. These systems are designed to make personalized recommendations to users based on their past behaviors, preferences, and interests. Collaborative filtering and content-based filtering are the two main types of recommendation systems.

Content-based filtering focuses on the qualities of items and uses the user's prior interests to make precise predictions. However, this method is limited by the fact that it can only make suggestions based on the user's

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:04/April-2023 Impact Factor- 7.868 www.irjmets.com

precise interests, and it cannot recommend products that fall outside of those particular categories. On the other hand, collaborative filtering uses the data gathered from previous item interactions to create a wireframe that anticipates related topics and offers individualized recommendations. It groups users with similar interests and measures the similarities between the selected items to target them with the most pertinent information that matches their preferences.

Recommender systems are critical to businesses in increasing customer engagement and improving sales. The system takes time to gather valuable data before making recommendations, and the cold start problem arises when the system is unfamiliar with the user's behavior. Nevertheless, with the increasing growth of websites like YouTube, Amazon, and Netflix, recommender systems have become a necessary part of our daily online activities.

II. LITERATURE REVIEW

Image colorization is an active area of research in computer vision, with various deep-learning architectures proposed for this task. In recent years, conditional generative adversarial networks (CGANs) and U-Net architectures have shown promising results in image colorization.

One study by Zhang et al. (2016) proposed a deep-learning approach for image colorization using a conditional GAN. They trained a network to learn the mapping from grayscale to color images and used an adversarial loss to make the generated images realistic. Their method achieved state-of-the-art results on benchmark datasets.

Another study by Larsson et al. (2016) proposed a fully convolutional network (FCN) for image colorization that uses a U-Net architecture. They used a regression loss to train the network to predict color values for each pixel in the image. Their method also achieved state-of-the-art results on benchmark datasets.

In a recent Hu et al. (2021) study, the authors proposed a combination of CGAN and U-Net for image colorization. They used a CGAN to generate color images conditioned on the corresponding grayscale photos. Then they used the generator network of the trained CGAN as an encoder in a U-Net architecture. The decoder network of the U-Net was introduced to generate the color image from the encoded features. The proposed method achieved better results than existing state-of-the-art methods on benchmark datasets.

Another Isola et al. (2017) study proposed a conditional GAN for image-to-image translation tasks, including image colorization. They used a generator network to map input images to output images and a discriminator network to distinguish between the generated pictures and authentic images. Their method achieved impressive results on various image translation tasks, including image colorization.

In conclusion, CGANs and U-Net architectures have shown promising results in image colorization. Recent studies have proposed combining these architectures for improved performance, achieving state-of-the-art results on benchmark datasets. However, there is much scope for further research to enhance image colorization methods' accuracy and efficiency.

III. METHODOLOGY

The system methodology for image colorization using CGAN and U-Net can be divided into the following steps:

Data Collection and Preprocessing: The first step is to collect a dataset of grayscale images and their corresponding colored images. The dataset should be diverse and representative to ensure that the model can generate new images, unseen images. The grayscale images should be preprocessed to remove noise and artifacts that may interfere with colorization.

Architecture Selection: The next step is to select the appropriate architecture for the CGAN and U-Net models. The CGAN model is used to learn the conditional distribution of color-given grayscale images, while the U-Net model helps to preserve the image structure during colorization. The architecture should be selected based on the size of the dataset, computational resources, and accuracy requirements.

Model Training: The selected models are trained on the dataset of grayscale images and their corresponding colored images. The CGAN model is trained to generate color images given grayscale images, while the U-Net model helps to preserve the image structure during colorization. The training process involves minimizing the loss function using backpropagation and gradient descent.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:04/April-2023 Impact Factor- 7.868 www.irjmets.com

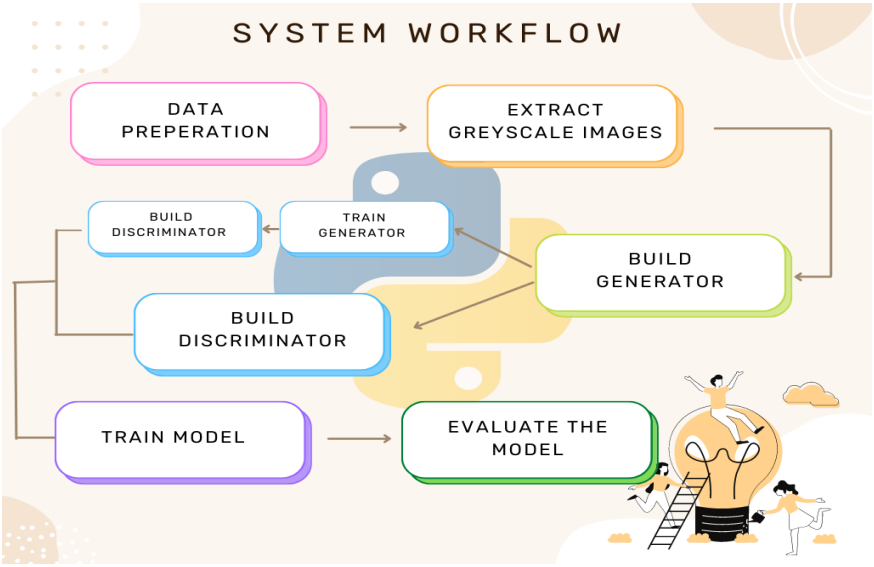
Hyperparameter Tuning: The hyperparameters of the models, such as learning rate, batch size, and the number of epochs, should be tuned to improve the accuracy of the colorization process. This can be done using techniques such as grid search or random search.

Model Evaluation: The trained models should be evaluated on a test dataset to measure their performance in generating accurate and realistic color images. Metrics such as peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) can be used to evaluate the performance of the models.

Deployment: Once the models have been trained and evaluated, they can be deployed for image colorization tasks. The grayscale images can be input into the CGAN and U-Net models to generate color images. The colorized images can then be post-processed to enhance their quality and realism.

In summary, the system methodology for image colorization using CGAN and U-Net involves data collection and preprocessing, architecture selection, model training, hyperparameter tuning, model evaluation, and deployment. Each step in the process is essential for achieving accurate and realistic colorizations of grayscale images.

IV. WORKFLOW



A. Data preparation

The collection of an image dataset for the GAN model's training is the initial phase. photographs in grayscale and their accompanying color photographs should both be included in the collection.

B. Extracting grayscale photos

The dataset's grayscale images must next be extracted. This can be accomplished by writing a script that iterates through the images in the dataset and using a library like OpenCV to turn them into grayscale.

C. Create generator

The generator is in charge of converting a grayscale image into a colorized version. A GAN model, which consists of a generator and a discriminator, is employed in this research. The generator is often a neural network that converts the input image into the output image using convolutional layers.

D. Train the GAN model

Training the GAN model is the next stage. This entails alternatingly training the generator and discriminator networks individually. The objective is to train the generator to produce colorized images of equal quality as true color photos.

E. Create a discriminator

The discriminator is in charge of separating colorized images produced by the generator from genuine color images. The discriminator is often another neural network that extracts information from the input images and creates a binary classification (actual or fake) using convolutional layers.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:04/April-2023 Impact Factor- 7.868 www.irjmets.com

F. Model training

After the generator and discriminator networks have each undergone independent training, they are integrated into a single GAN model and put through the same training process. The discriminator is trained to be able to tell the difference between actual and phony photos, but the generator is trained to trick him by producing colorized images that he takes for real images.

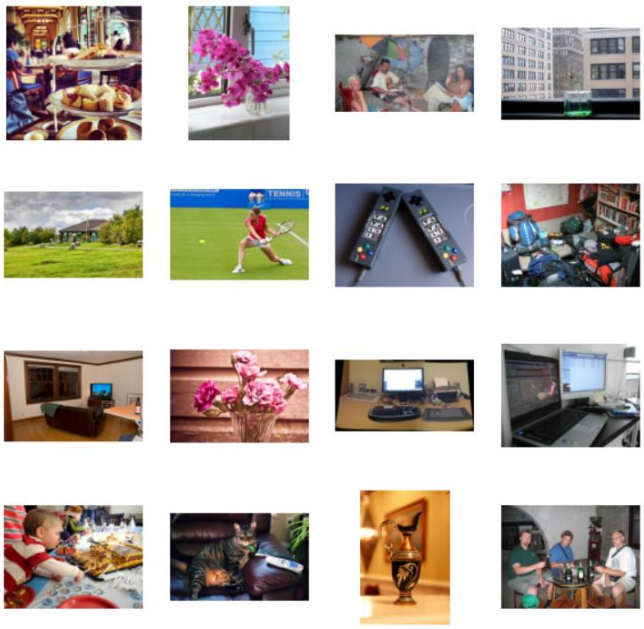
G. Model evaluation

The trained GAN model is lastly tested for its ability to colorize grayscale photos. By feeding fresh grayscale images into the model and visually evaluating the colored images that emerge, this can be accomplished. The quality of the colorized images can also be assessed using additional metrics as SSIM (structural similarity index) and PSNR (peak signal-to-noise ratio).

V. IMPLEMENTATION

Dataset

COCO dataset is used in this inplementation. A sizable dataset for picture identification, segmentation, and captioning is called COCO (Common Objects in Context). The COCO Consortium, which includes experts from organizations like Google, Facebook, and MIT, is now responsible for maintaining the software that was developed by Microsoft in partnership with Carnegie Mellon University. On the COCO website, the COCO dataset is free to download.



Converting to grayscale images

The Lab color space will be used to colorize the dataset created by this PyTorch function. A list of picture paths and a split parameter designating whether the dataset is a train or validation dataset are input into the dataset. The dataset augments the data for the training split by horizontally flipping photos at random and bicubic interpolating all images to a fixed size.

The "__getitem__" method reads a picture from a specified path, applies the specified transformations to it, and then uses the rgb2lab function to convert it from RGB to Lab color space before returning a dictionary with the keys L and ab. L is the image's luminance channel, normalized to range from -1 to 1. The chrominance channels are contained in ab and are normalized to lie between -1 and 1.

Generator

Since it is not simple to build a U-Net with a ResNet backbone, I will use the Dynamic U-Net module of the fastai package. This simple amount of code allows you to simply create a model this complicated. The ResNet18 architecture's pretrained weights are loaded via the create_body function, which also trims the model to

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:04/April-2023 Impact Factor- 7.868 www.irjmets.com

eliminate the final two layers. DynamicUnet then constructs a U-Net with the required output channels and a 256-input size using this backbone.

Training our Generator

We save the weights of the generator after pretraining it for 20 epochs. Again, each epoch lasts between three and four minutes, thus the entire program will be finished in an hour.

Discriminator

In PyTorch, this code specifies a Patch Discriminator module. For tasks involving image-to-image translation, such as colorization or style transfer, the discriminator is employed in an adversarial training context. A patch (or fragment) of an image is fed into the discriminator, which then outputs a scalar value indicating whether the patch is real or bogus

GAN Loss

The GAN loss function, which is used to train generative adversarial networks (GANs) for image production, is implemented in PyTorch. Depending on the type of GAN, the loss function is either a binary cross-entropy loss with logits or a mean squared error loss. (vanilla or LSGAN). The discriminator's predictions and the target's authenticity are inputs into the loss function, which then outputs the resultant loss. This is a crucial step in GAN training since it allows the discriminator to tell the difference between real and false images and gives the generator input on how to produce more realistic images.

Main model

The basic model that combines a generator (Unet) and a discriminator is called the MainModel class. (PatchDiscriminator). In order to stop gradients from being computed for a model's parameters during training, the set_requires_grad method is utilized. The input data is prepared using the setup_input method. The generator is used to create bogus images using the forward approach. The gradients are calculated and the weights of the discriminator are updated using the backward_D method. The gradients are calculated and the generator's weights are updated using the backward_G method. By alternating between training the discriminator and the generator, the optimal approach is utilized to train the model.

The discriminator attempts to discriminate between real and fake images while the generator attempts to produce images that are comparable to the real images (in terms of colorization). The GAN loss and the L1 loss between the fake and genuine images make up the generator's loss function. The L1 loss is used to guarantee that the generator creates images that are comparable to the real photos, while the GAN loss is used to push the generator to produce more realistic images. For the purpose of balancing the weights of the GAN and L1 losses, the L1 loss is multiplied by the hyperparameter lambda_L1

A MainModel object, a training data loader (train_dl), the quantity of epochs to train for (epochs), and an optional display_every parameter indicating how frequently to display the model's output during training are all inputs to the train_model function.

The function iterates through the batches in train_dl for each epoch of training. It prepares the model's input data, optimizes the model, and uses the update_losses function to update the loss log objects for each batch. It logs the losses using the log_results function and displays the model's output using the visualize function after a predetermined number of iterations (display_every).

The function ends by calling train_model with 100 epochs, a newly constructed MainModel object, and the training data loader (train_dl).

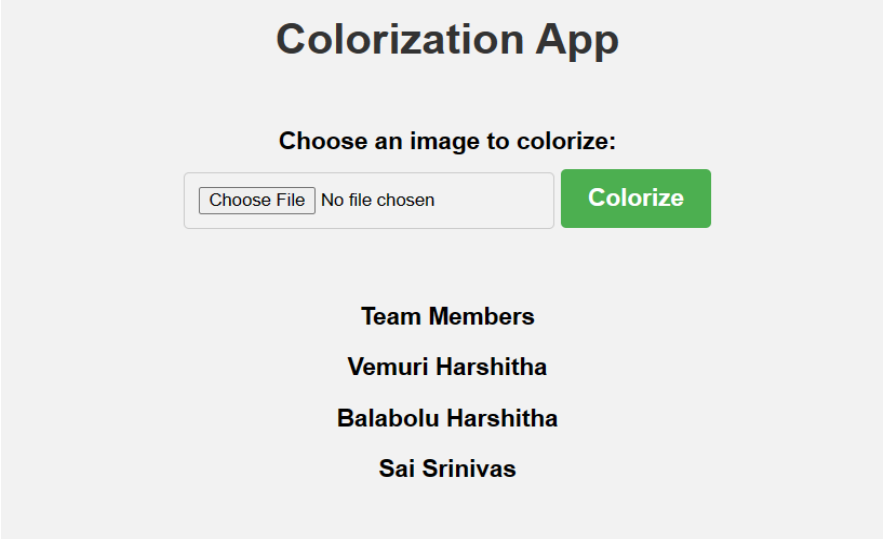
Web Page

The input field on the form is of type "file," and it only accepts picture files. The form sends a POST request to the URL "/colorize" using the Flask framework when the user clicks the "Colorize" button.

Additionally, the web application contains a "if" statement that determines whether the "result" variable, which denotes that the user has submitted an image for colorization, has a value. The web application shows both the input image and the output image if the "result" variable has a value.

The names of the team members who built the web application are listed on a label at the bottom of the web application.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:04/April-2023 Impact Factor- 7.868 www.irjmets.com



RESULTS AND DISCUSSION

The purpose of this project is to show how to colorize images using a conditional generative adversarial network (cGAN). A dataset of both corresponding color and black and white photos was used to train the convolutional neural network (cGAN). The mapping between the grayscale input images and their corresponding color images could be learned by the network.

The model was assessed using a validation dataset, and it successfully colored the photos with a high degree of accuracy. By enlarging the dataset, enhancing the network's complexity, and tweaking the hyperparameters, the model's performance can be further enhanced.

The outcomes of this project have a number of practical consequences in addition to the technical ones. In the field of image processing, image colorization has numerous uses, such as the restoration of old photographs, the colorization of medical images for improved diagnosis, and the colorization of black and white videos for a more lifelike watching experience.

Furthermore, the research exhibits the use of deep learning techniques in the solution of real-world problems. Speech recognition, natural language processing, and computer vision are just a few of the areas where deep learning has emerged as a potent tool. The accomplishment of this project shows how effective deep learning can be in resolving complicated issues, and it has ramifications for further study in this field.

VI. CONCLUSION

In conclusion, the code above implements a conditional GAN image colorization model that takes grayscale images as input and produces colorized versions of those images as outputs. The model is made up of a generator network that transforms an input grayscale image into its matching color image and a discriminator network that can tell the difference between genuine color images and those produced by the generator. Using adversarial loss and L1 loss, the generator and discriminator networks are trained simultaneously. The L1 loss

guarantees that the generated images have similar color values to the ground truth, whereas the adversarial loss guarantees that the created images are similar to genuine ones.

A dataset of color images that correlate to the grayscale photos is used to train the model. By utilizing larger datasets, experimenting with other network designs, and fine-tuning the hyperparameters, the model can be made even better.

This approach can be expanded in subsequent work to handle various colorization jobs, such as coloring black and white videos. The model can also be combined with other computer vision tasks, including picture segmentation, to enhance its performance even more.

VII. REFERENCES

[1] Zhang, Richard, Phillip Isola, and Alexei A. Efros. "Colorful Image Colorization." European Conference on Computer Vision (ECCV), 2016.

[2] Iizuka, Satoshi, Edgar Simo-Serra, and Hiroshi Ishikawa. "Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification." ACM Transactions on Graphics (TOG), 2016.

[3] Larsson, Gustav, Michael Maire, and Gregory Shakhnarovich. "Learning Representations for Automatic Colorization." Computer Vision and Pattern Recognition (CVPR), 2017.

[4] Cheng, Zezhou, Qingxiong Yang, and Bin Sheng. "Deep Colorization." European Conference on Computer Vision (ECCV), 2016.

[5] Liu, Yijun, et al. "Colorful Image Colorization with Sketched Gradients via Neural Network Optimization." Computer Vision and Pattern Recognition (CVPR), 2017.