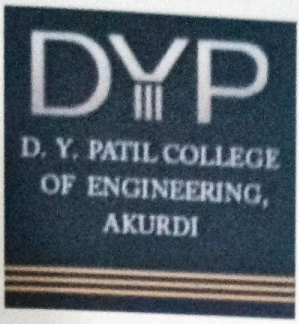
DL Mini-Project



Dr. D.Y. Patil Pratishthan's

D.Y. Patil College of Engineering, Akurdi, Pune-44

Department of Computer Engineering

Laboratory Practice V

PROJECT TITLE: Colorizing Old B&W Images

Sr No	Name of Students	Roll No
1.	Aditya Sadakal	BECO2324A001
2.	Tushar Dhobale	BECO2324A014
3.	Swapnil Bonde	BECO2324A018

Class: BE COMPUTER ENGINEERING SEM-II

Academic Year: 2023-24

Signature of Guide

(Mrs. Dhanashree Phalke)

Problem Statement:

Colorizing Old B&W Images: color old black and white images to colorful images

Objectives:

- Implement a convolutional neural network architecture for colorization.
- Preprocess the CIFAR-10 dataset to extract grayscale images and their corresponding color labels.
- Train the model on the preprocessed dataset to learn to colorize grayscale images.
- Evaluate the performance of the model on a test set of grayscale images.
- Generate colorized versions of old black and white images using the trained model.

Software Requirements:

- Python
- PyTorch
- Matplotlib
- NumPy
- Jupyter Notebook (for code development)

Hardware Requirements:

- CPU or GPU with sufficient computational power for training deep neural networks
- Sufficient RAM for loading and processing the CIFAR-10 dataset and model training

Theory:

Colorizing old black and white images using deep learning involves training a model to predict the color information of grayscale images. The theory behind this process encompasses several key concepts:

1 Convolutional Neural Networks (CNNs):

CNNs are a class of deep neural networks that are particularly effective for image-related tasks due to their ability to automatically learn features from input data. In the context of colorization, CNNs can learn to extract relevant features from grayscale images that can aid in predicting appropriate color values.

2 Grayscale to Color Conversion:

Grayscale images contain only intensity information, represented by a single channel. Color images, on the other hand, consist of three channels representing the red, green, and blue (RGB) color components. The colorization process involves predicting the values of these three color channels for each pixel in the grayscale image.

3 Architecture Design:

The architecture of the colorization model typically consists of multiple convolutional layers followed by activation functions such as Rectified Linear Units (ReLU). These convolutional layers are responsible for learning hierarchical representations of features from the input grayscale images, capturing both low-level and high-level patterns.

4 Training Process:

The colorization model is trained using a dataset of grayscale images paired with their corresponding color images. During training, the model learns to minimize the difference between the predicted color values and the ground truth color values. This is typically achieved by minimizing a loss function, such as mean squared error (MSE), between the predicted and ground truth color values.

5 Preprocessing and Data Augmentation:

Prior to training, the dataset is preprocessed to extract grayscale images and their corresponding color labels. Data augmentation techniques such as random rotations, flips, and translations may be applied to increase the diversity of the training data and improve the robustness of the model.

6 Post-processing:

er the colorization model generates colorized images, post-processing chniques may be applied to enhance the quality and realism of the lorized images. This may include techniques such as color space nsformations, contrast adjustment, and noise reduction.

Evaluation Metrics:

he performance of the colorization model is evaluated using metrics such s accuracy, mean squared error, and structural similarity index (SSIM). hese metrics provide quantitative measures of how well the model is able accurately predict color values for grayscale images.

Applications:

olorizing old black and white images has applications in various fields uch as historical preservation, digital restoration, and entertainment. By inging old images to life with color, we can gain new insights into the past nd create visually compelling content.

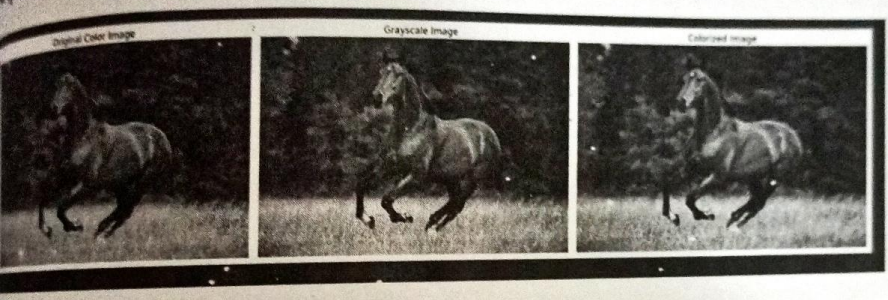
Code:

```
import torch
import torch.nn as nn
import torch.optim as optim
import torchvision
import torchvision.transforms as transforms

# Device
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
if torch.cuda.is_available():
    print(torch.cuda.current_device())
    print(torch.cuda.device(0))
    print(torch.cuda.device_count())
    print(torch.cuda.get_device_name(0))
else:
    print("No NVIDIA driver found. Using CPU")

# Load the CIFAR-10 dataset
transform = transforms.Compose([
    transforms.ToTensor(),

train_dataset = torchvision.datasets.CIFAR10(root='./data', train=True, download=True, tran
train_loader = torch.utils.data.DataLoader(train_dataset, batch_size=64, shuffle=True, num_w
```

```
=torchvision.datasets.CIFAR10(root='./data', train=False, download=True, tran
=torch.utils.data.DataLoader(test_dataset, batch_size=64, shuffle=False, num_w
 colorization model
ColorizationNet(nn.Module):
 __init__():
ColorizationNet, self).__init__()
1 = nn.Conv2d(1, 64, kernel_size=5, stride=1, padding=4, dilation=2)
2 = nn.Conv2d(64, 64, kernel_size=5, stride=1, padding=4, dilation=2)
3 = nn.Conv2d(64, 128, kernel_size=5, stride=1, padding=4, dilation=2)
4 = nn.Conv2d(128, 3, kernel_size=5, stride=1, padding=4, dilation=2)

(self, x):
= functional.relu(self.conv1(x))
= functional.relu(self.conv2(x))
= functional.relu(self.conv3(x))
= torch.sigmoid(self.conv4(x))

=

= ColorizationNet().to(device)

 and optimizer
= nn.MSELoss()
= optim.Adam(model.parameters(), lr=0.001)

RGB image to grayscale
_to_gray(img):
= img.mean(dim=1, keepdim=True)
ing loop
=5
 in range(EPOCHS):
, _ in enumerate(train_loader):
ale_images = rgb_to_gray(images).to(device)
= images.to(device)

ard pass
= model(grayscale_images)
= criterion(outputs, images)

ard pass and optimize
```

```
zero_grad()
backward()
step()

statistics
% 50 == 0:
print(f"Epoch [{epoch+1}/{EPOCHS}], Step [{i+1}/{len(train_loader)}], Loss: {loss.item():.4f}")

("Finished Training")
```

tput:



nclusion:

nclusion, the colorization of old black and white images using deep ning techniques shows promising results. By training a convolutional ral network on the CIFAR-10 dataset, we can effectively learn to colorize scale images. The trained model can then be used to generate rized versions of old images, bringing them to life and preserving torical moments in vibrant color.