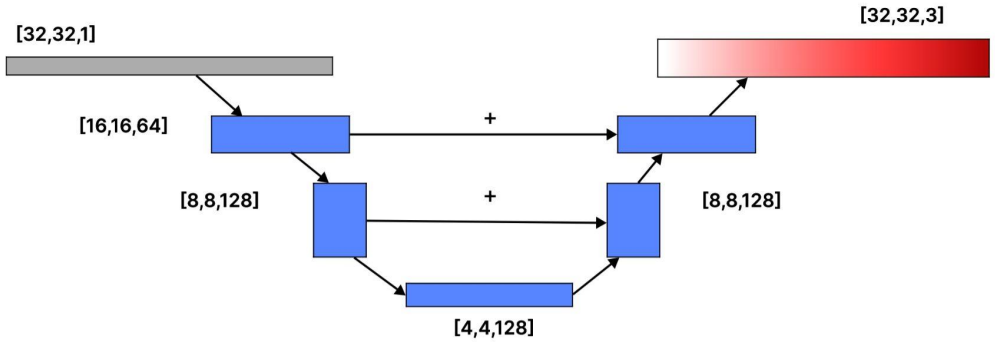
Introduction

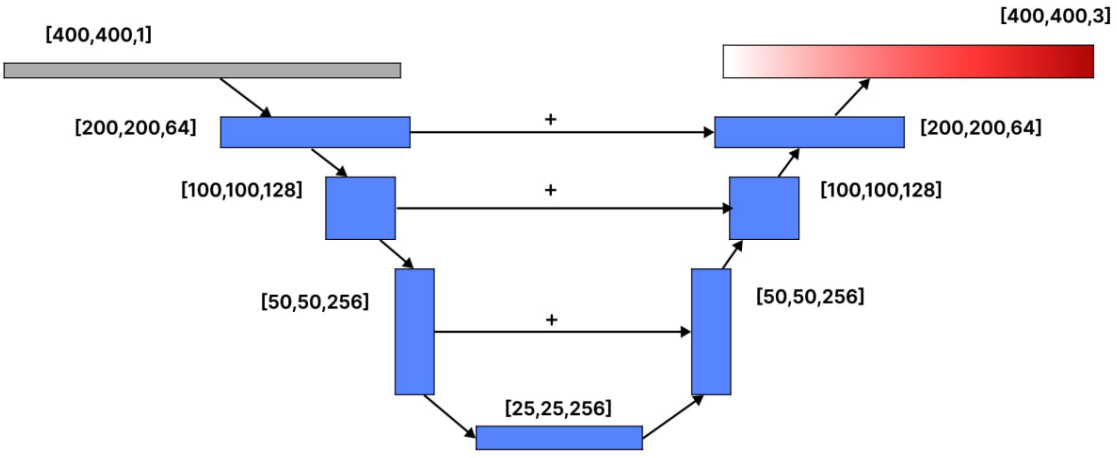
This study is dedicated to the development of a system capable of automatically colorizing black and white images using Generative Adversarial Networks (GANs). GANs are a type of deep learning model that comprises two components: a generator and a discriminator. The generator is tasked with generating realistic colorized images, while the discriminator attempts to differentiate between real and generated images. Through an adversarial training process, the generator learns to produce high-quality colorized images that are indistinguishable from real color images.

The generator is designed with an encoder-decoder architecture, while the discriminator consists of one encoder plus fully-connected layers, which gives one-class output. Skip-connection is applied in the generator, specifically in the U-Net architecture.

The architecture of generator for (32x32) images:



The architecture of generator for (400x400) images:



The Generative Adversarial Network (GAN) architecture comprises each block of a Convolutional Transpose (Conv) layer, followed by Batch Normalization and a Leaky Rectified Linear Unit (ReLU). The final activation function for the generator is the hyperbolic tangent (tanh), while for the discriminator, it is the sigmoid function. The input images are normalized prior to being fed into the network. The model underwent optimization utilizing the Adam optimizer.

The dataset used for this project is CIFAR10 with some modifications and the Image Colorization Dataset from Kaggle. (https://www.kaggle.com/datasets/aayush9753/image-colorization-dataset/)

The following instructions are provided for the use of the notebooks:

- `PML_Project.ipynb` is used to preprocess images to 4 filters for training the image.
- `PML_Project_GAN.ipynb` is used for model training on the CIFAR10 dataset. To train on 4 filters, replace the variable i with the number of the filter in the following variables:

```
X_train_images_dir = f'dataset/train/filter{i}'
X_test_dir = f'dataset/test/filter{i}_test'
date = f'1126.filter{i}'
```

- `PML_Project_GAN_Large.ipynb` is used for model training on the large dataset.
- `PML_Project_Test` is used to colorize the image. Here, the image should be placed in the 'dataset/val' folder for 32*32 images and ‘data/val’ for 400*400 images. For large images you also need to write ‘True’ in args_dict[‘large’]. The result will be in the 'img/Test/GAN__100L1_bs32_Adam_lr0.0001' folder.

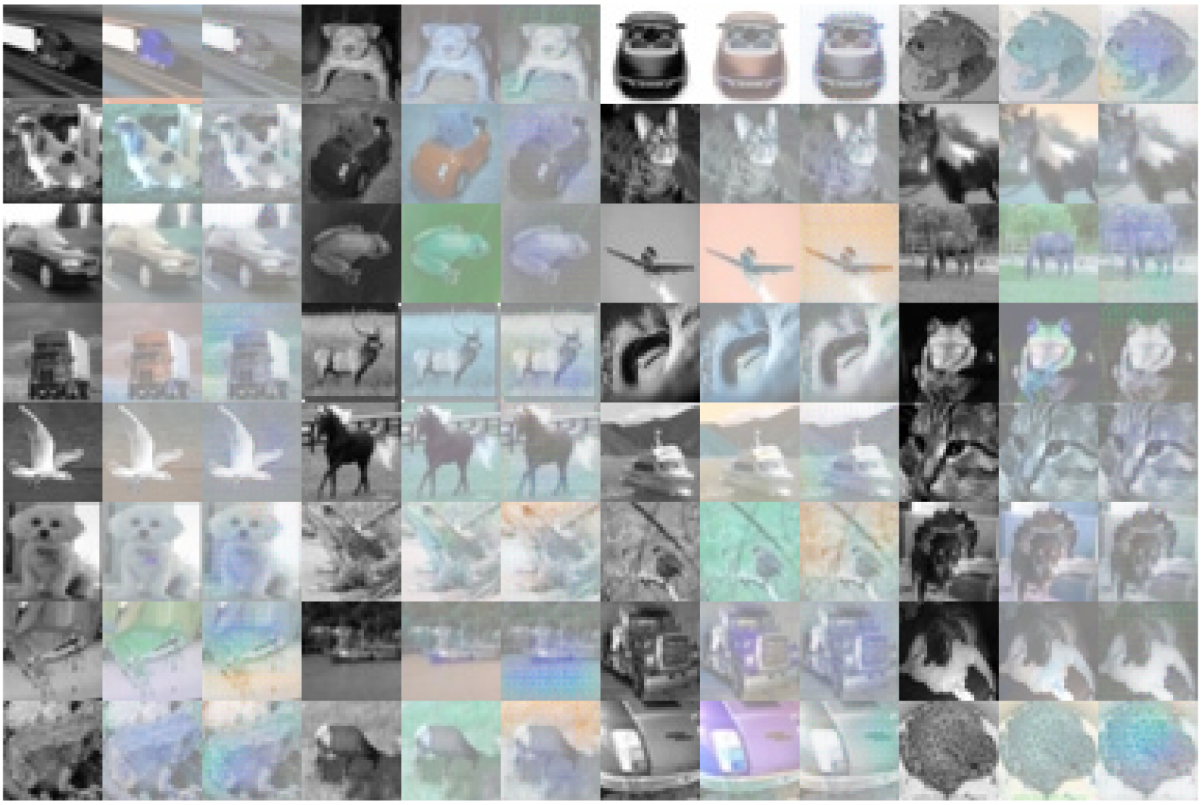
Results

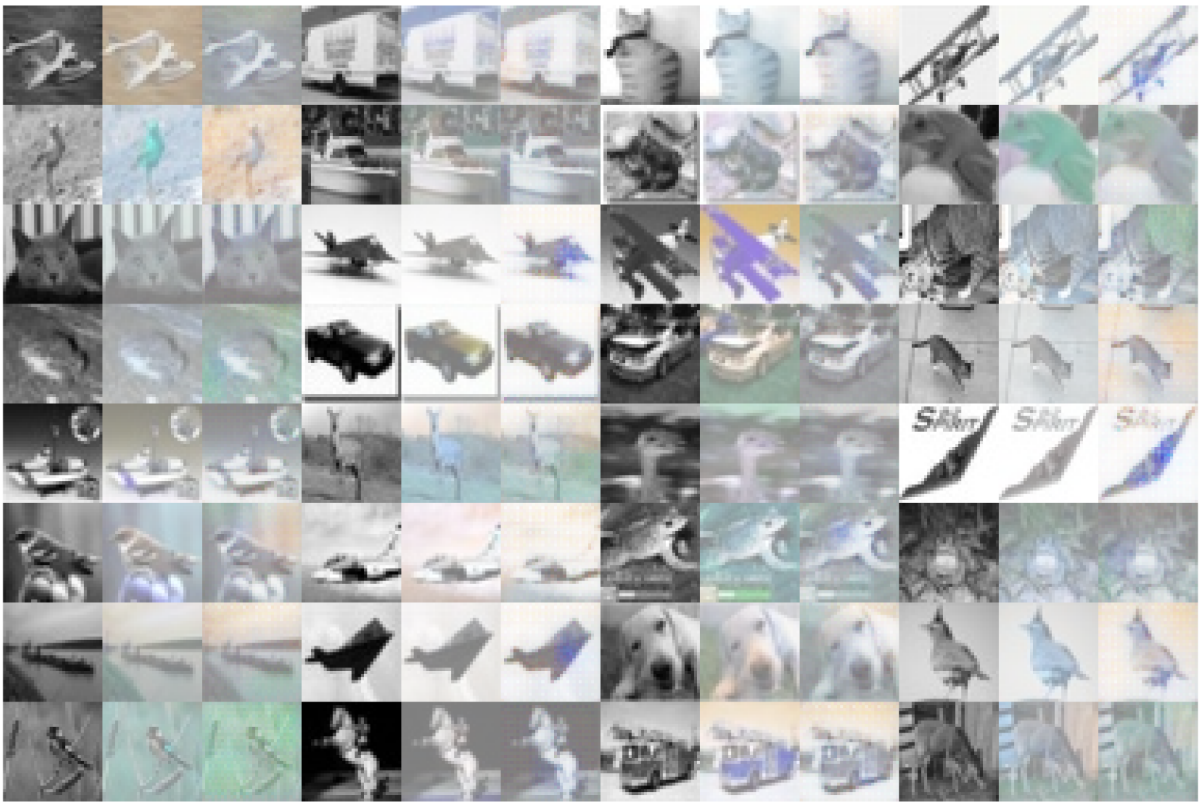
This section presents the results of our project, specifically focusing on the comparison of generated images with their original counterparts. The images are arranged in a horizontal configuration, consisting of three columns. The first column represents the grayscale image, which serves as the input. The second

column displays the raw image, which serves as the ground truth. The third column presents the generated image, which serves as the output. Ideally, the generated image (third column) should closely resemble the raw image (second column). These output images were generated from the testing set.

Rewrite

Filter1:





Large dataset, resized to 32*32