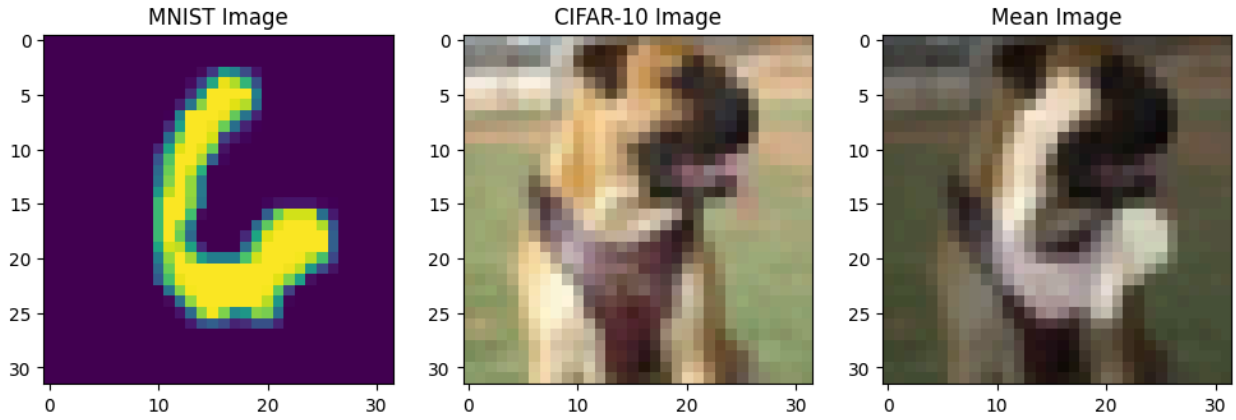
Deep Learning Course

Assignment 7

Merge Dataset



Objective

In this assignment, you will work with Autoencoders and Variational Autoencoders to create a model that can reconstruct two original images from a merged image. The merged image is the mean of one image from the CIFAR-10 dataset and one from the MNIST dataset. You will use PyTorch for this task.

Tasks

1. Dataset Creation

Create a merged dataset where each image is the mean of one image from CIFAR-10 and one from MNIST.

2. Model Design

Design a model using Autoencoders or Variational Autoencoders to reconstruct the two original images from the merged image.

3. Training and Evaluation

Train your model on the merged dataset and evaluate its performance in reconstructing the original images.

Detailed Steps

1. Dataset Preparation

- Download the CIFAR-10 and MNIST datasets.
- Preprocess the datasets to ensure they have compatible shapes.
- Create the merged dataset by taking the mean of corresponding images from CIFAR-10 and MNIST.

2. Model Design

- Design an Autoencoder or Variational Autoencoder model using PyTorch.
- The encoder should take the merged image as input and learn to encode it into a latent space.
- The decoder should reconstruct the two original images from the latent representation.

3. Training the Model

- Train the model using the merged dataset.
- Use appropriate loss functions to ensure the reconstructed images are as close as possible to the original CIFAR-10 and MNIST images.

4. Evaluation

- Evaluate the performance of your model using suitable metrics.
- Visualize the reconstructed images and compare them with the original ones.

Code for Dataset Preparation

Below is a sample code to create the merged dataset from CIFAR-10 and MNIST using PyTorch:

```
class MergedDataset(Dataset):
    def __init__(self, dataset1, dataset2):
        self.dataset1 = dataset1
        self.dataset2 = dataset2
        self.resize = transforms.Resize((32, 32))

    def __len__(self):
        return min(len(self.dataset1), len(self.dataset2))

    def __getitem__(self, idx):
        image1, _ = self.dataset1[idx]
        image2, _ = self.dataset2[idx]

        # Resize image1 to match the size of image2
        image1 = self.resize(image1)

        # If image1 is grayscale, duplicate the channel dimension to match
image2
        if image1.shape[0] == 1:
            image1 = torch.cat([image1] * 3, dim=0)

        # Compute the mean of the two images
```

```
        mean_image = (image1 + image2) / 2

        return mean_image
```

Additional Guidelines

- Ensure your model architecture is well-defined and documented.
- Use suitable optimization techniques and hyperparameters.
- Provide a detailed report including your model design, training process, and evaluation results.
- Include visualizations of some reconstructed images compared to the original ones.

Submission

Submit the following:

1. Your code for creating the merged dataset.
2. The PyTorch model code for the Autoencoder or Variational Autoencoder.
3. A report detailing your approach, results, and any challenges faced.
4. Visualizations of reconstructed images and the original images.