CSET419 – Introduction to Generative AI

Lab – 2

Objective

To implement and train a basic GAN model to generate new synthetic images and evaluate the quality of generated samples over training epochs.

Experiment 2: Train a Basic GAN Model for Image Generation

A university's digital archive server has partially crashed, and many old scanned images are missing. Before recovering the archive, the team wants to simulate synthetic images so that their AI pipeline can be tested end-to-end. Your task is to train a basic Generative Adversarial Network (GAN) that can generate realistic-looking synthetic images similar to a given dataset.

Dataset

Use one dataset only (choose any ONE):

- MNIST (handwritten digits)
- Fashion-MNIST (clothing items)

Dataset must be loaded using either TensorFlow/Keras or PyTorch/Torchvision dataset utilities.

Input Parameters

Your program must take the following inputs from the user (using input() or clearly defined configuration variables):

- dataset_choice: 'mnist' or 'fashion'
- epochs: positive integer (recommended 30–100)
- batch_size: positive integer (recommended 64 or 128)
- noise_dim: positive integer (recommended 50 or 100)
- learning_rate: positive float (e.g., 0.0002)

- save_interval: positive integer (save samples every k epochs, e.g., 5)

Task

You must:

1. Design a Generator network that converts random noise vectors into images.
2. Design a Discriminator network that classifies images as real or fake.
3. Train the GAN for the given number of epochs using alternating training of Discriminator and Generator.
4. Generate and save synthetic images periodically during training.
5. Predict the labels of generated images using a pre-trained classifier model (transfer learning) and report label distribution.

Instructions

- Images must be normalized correctly (e.g., scaled to [0,1] or [-1,1]).
- Generator output shape must match the dataset image shape.
- Training must include loss computation for both Generator and Discriminator.
- Save images as a grid (at least 5×5 = 25 samples per saved image).

Expected Output

Your program must generate the following outputs:

Output 1: Training Logs

Print epoch-wise logs in the format:

Epoch 10/50 | D_loss: 0.53 | D_acc: 78.12% | G_loss: 1.24

Output 2: Generated Samples Folder

Create a folder named: generated_samples/

Save generated samples at each interval with filenames: epoch_05.png, epoch_10.png, ...

Output 3: Final Generated Images

At the end of training, generate and save 100 synthetic images in the folder: final_generated_images/

Output 4: Labels of Generated Images

Using a pre-trained classifier, predict labels for the 100 generated images.