

# Deep Convolutional Generative Adversarial Network

## Problem Statement

This project demonstrates generating images of handwritten digits resembling the MNIST dataset using a Deep Convolutional Generative Adversarial Network (DCGAN). It also integrates SQLite to store and manage data, showcasing SQL usage in machine learning workflows.

## Overview

Generative Adversarial Networks (GANs) involve training two models simultaneously:

- **Generator:** Creates images that mimic real ones.
- **Discriminator:** Distinguishes real images from fake ones.

The adversarial setup refines both models to improve image generation and classification.

## Architecture

- **Generator:** Uses random noise as input, upsampling with Conv2DTranspose layers, and applies LeakyReLU activations (output layer uses tanh).
- **Discriminator:** A CNN that classifies images as real or generated.

## Training Loop

1. The generator creates an image from random noise.
2. The discriminator classifies real and generated images.
3. Loss is calculated for both models.
4. Gradients update the generator and discriminator.

## Dataset and SQLite Integration

The MNIST dataset is used for training. SQLite is employed to store images and labels, demonstrating efficient data storage and retrieval in machine learning workflows.

## Key Features:

- Image generation with DCGAN.
- Practical SQL integration for data management.
- Hands-on application of adversarial training techniques.

This project showcases the synergy of machine learning and SQL for innovative data-driven solutions.