**Title: Generating Synthetic Images Using GAN on CIFAR-10 Dataset**

**1. Introduction**

This report documents the implementation of a Generative Adversarial Network (GAN) trained on the CIFAR-10 dataset to generate synthetic images. It covers dataset preparation, GAN architecture, training procedures, challenges, and evaluation metrics.

**2. Dataset Preparation**

The CIFAR-10 dataset consists of 60,000 color images (32x32 pixels) across 10 classes. The dataset was preprocessed using the following steps:

* Normalization: Pixel values were scaled to [-1, 1] using Min-Max normalization.
* Batching: The dataset was loaded in batches for efficient training.
* Augmentation (if applicable): Data augmentation techniques such as horizontal flipping were considered to improve generalization.

**3. GAN Architecture**

The GAN consists of a **Generator** and a **Discriminator**:

* **Generator:**
  + Uses Transposed Convolution (Conv2DTranspose) to upsample from a latent space (random noise input).
  + Utilizes Batch Normalization and ReLU activation.
  + Outputs a 32x32x3 image using a Tanh activation function.
* **Discriminator:**
  + Uses Convolutional layers (Conv2D) to extract image features.
  + Leverages LeakyReLU activation and Dropout for stabilization.
  + Outputs a probability score using the Sigmoid activation function.

**4. Training Procedure & Hyperparameters**

The GAN model was trained using the following:

* **Loss Function:** Binary Cross-Entropy (BCE)
* **Optimizer:** Adam (learning rate = 0.0002, beta1 = 0.5)
* **Batch Size:** 128
* **Epochs:** 50+ (monitored for convergence)
* **Latent Vector Dimension:** 100
* **Weight Initialization:** Xavier/Glorot initialization for stability

Training was carried out in alternating steps:

1. **Train the Discriminator** on real and generated images.
2. **Train the Generator** to produce realistic images that fool the Discriminator.

**5. Challenges and Mitigation Strategies**

**Mode Collapse:**

* Used feature matching and mini-batch discrimination to ensure diverse outputs.
* Introduced noise perturbations during training to prevent overfitting.

**Vanishing Gradients:**

* Applied LeakyReLU in the Discriminator.
* Used Batch Normalization to stabilize updates.

**Training Instability:**

* Used Adam optimizer with tuned hyperparameters (beta1 = 0.5).
* Implemented label smoothing to prevent overconfident updates.

**6. Evaluation Metrics**

**Quantitative:**

* **Fréchet Inception Distance (FID):** Measures similarity between generated and real images.
* **Inception Score (IS):** Evaluates diversity and realism of images.

**Qualitative:**

* Visual inspection of generated images to assess realism and diversity.
* Side-by-side comparison with real CIFAR-10 images.

**7. Results & Conclusion**

* The GAN successfully generated visually realistic images resembling CIFAR-10 classes.
* Lower FID scores indicated improvements in quality.
* Training stability improved with the use of tuned hyperparameters and architectural modifications.
* Future improvements could include using Wasserstein GAN (WGAN) to enhance training robustness.

**References:**

* Goodfellow et al., "Generative Adversarial Networks" (2014).
* Heusel et al., "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium" (2017).