Report on Image Generation with VAE and GANs

1. Introduction:

The purpose of this assignment was to explore the use of generative models for signature image generation, focusing on two key approaches: Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs). Both models were trained using a dataset of segmented signature images, and the process involved data preprocessing, augmentation, model training, and image generation. The goal was to understand the architectures and their ability to generate realistic signature images.

2. Methodology:

Dataset:

The dataset used for this task consists of segmented signature images stored in the folder `segmented_signatures`. Each image was grayscale, and all images were resized to a fixed shape of 64x64 pixels to ensure uniformity in training.

Preprocessing:

The preprocessing phase involved:
- Loading the images from the dataset using `cv2` for reading grayscale images.
- Resizing each image to 64x64 pixels.
- Normalizing pixel values to the range [0, 1] for better model performance.
- Reshaping the images to have the format (batch_size, 1, 64, 64) where 1 is the number of channels (since they are grayscale images).

Data Augmentation:

To further enhance the diversity of the data and improve generalization during model training, data augmentation techniques such as random rotations, flipping, and slight translations were applied to the images.

VAE Architecture:

The Variational Autoencoder (VAE) model architecture consisted of an encoder that compressed the input image into a latent representation and a decoder that reconstructed the image from this latent space. The VAE trained on the dataset to learn a probabilistic representation of the images. Key components of the architecture included:
- Encoder: Two convolutional layers followed by a fully connected layer.
- Latent space: Representation of the image in a lower-dimensional space.
- Decoder: Symmetrical to the encoder, used to reconstruct the image from the latent vector.

- KL Divergence Loss: Encouraged the latent distribution to be close to a standard normal distribution.
- Reconstruction Loss: Captured the difference between the generated and real images (using mean squared error).

VAE Training:

The VAE was trained for 20 epochs with a batch size of 32, using the Adam optimizer. The loss function combined the KL divergence and reconstruction loss.

GAN Architecture:

The Generative Adversarial Network (GAN) comprised two networks:
- Generator: A neural network that learns to generate synthetic images by taking in a random noise vector and producing an image in the same shape as the real data.
- Discriminator: A binary classifier that differentiates between real images and fake (generated) images.

GAN Training:

The GAN was trained using the following steps:
- Random noise vectors were fed to the generator to produce synthetic images.
- These generated images, along with real images from the dataset, were fed to the discriminator.
- The discriminator predicted whether the input was real or fake, and both networks were updated based on the feedback.

3. Results:

Training VAE:

The training loss for the VAE, combining KL divergence and reconstruction loss, decreased consistently across the 20 epochs. After training, the VAE was capable of generating signature images with recognizable features.
- Training Loss: The loss decreased steadily, indicating that the VAE was learning to generate more accurate images over time.
- Generated Images: The images generated by the VAE displayed smooth and realistic structures, although some small artifacts persisted.

Training GAN:

The GAN showed competitive improvement as the generator learned to produce increasingly realistic images while the discriminator honed its ability to distinguish between real and fake images. Over 100 epochs, the generator was able to produce clear signature-like images.
- Training Loss: GAN training is typically unstable, and we observed fluctuations in both the generator and discriminator losses. Over time, however, the generated images showed significant improvements.

- Generated Images: By the end of training, the generator was capable of producing images that resembled the real signature dataset closely, with fewer artifacts compared to the VAE.

4. Discussion:

Sentence Coherence:

For VAE:
- The VAE's ability to generate images relied on learning meaningful latent representations. As training progressed, the coherence and quality of the images improved, with the generated signatures showing increasingly clear strokes and smoother outlines.
- The generated images initially contained noise, but by epoch 20, the model had significantly improved in creating legible images.

For GAN:
- The GAN initially produced noisy images, but as the generator received feedback from the discriminator, the quality of generated images improved considerably.

Challenges Encountered:

- **VAE**: Balancing the reconstruction and KL divergence losses to ensure meaningful latent space representations was a challenge.
- **GAN**: GAN training posed more challenges due to instability, requiring careful hyperparameter tuning and frequent adjustments to the learning rate.
- **Hardware Constraints**: The training process, particularly for GANs, required extensive computation. Using TPU in the Kaggle environment was essential to expedite the process.

5. Conclusion:

This assignment demonstrated the effectiveness of VAEs and GANs in generating signature images. The VAE model was able to capture a lower-dimensional latent space representation, while the GAN model produced highly realistic images through adversarial training.
- **VAE**: Showed consistent improvement in generating images, but the quality was slightly limited by the latent space dimensionality.
- **GAN**: Produced highly realistic signature images, with more fine details and clearer structures.

Overall, both models provided valuable insights into generative models, and GANs, in particular, showed superior performance for image generation tasks. This experiment highlighted the potential of these models for various applications, such as forgery detection and synthetic data generation.