GenCV00 Assessment Report

1 Project Objective

The objective of this project is to use a pre-trained VAE to generate ten diverse yet consistent facial images from a single given input image.

2 Background on Variational Autoencoders (VAEs)

A Variational Autoencoder is a deep generative model that learns to compress and then reconstruct data. It consists of two components connected by a probabilistic latent space:

- **The Encoder:** This neural network takes an input image and maps it to a compressed, low-dimensional representation. Unlike a standard autoencoder which maps to a single point, the VAE's encoder maps the input to a **probability distribution** in the latent space. This distribution is Gaussian, which is defined by two vectors: a mean (μ) and a log-variance ($\log_v ar$). This output defines a region in the latent space where the encoded image is likely to exist.
- **The Decoder:** This network takes a point sampled from the latent space and attempts to reconstruct the original input image from it.

The generative power of the VAE comes from the **Reparameterization Trick**, that makes the model trainable via backpropagation, as the original random sampling is not differentiable. To generate a new sample, we first sample a random noise vector (ε) from a standard normal distribution (mean=0, variance=1). This random noise is then scaled by the learned standard deviation (derived from $\log_v ar$) and shifted by the learned mean (μ) to produce the final latent vector $\mathbf{z}$:

$$\mathbf{z} = \mu + \varepsilon \cdot \sigma \quad (\text{where } \sigma = \exp(0.5 \cdot \log_v ar))$$

3 Implementation and Methodology

3.1 Environment and Tools

The project was implemented in a Google Colab environment. The tools are:

- **Libraries:** Python, PyTorch for model definition and tensor operations, and Matplotlib for data visualization.
- **Generative Model Libraries:** The Hugging Face diffusers library was used for its robust and easy-to-use implementation of the Stable Diffusion pipeline.
- **Pre-trained Models:**
  - **VAE:** A Variational Autoencoder model pre-trained on the CelebA dataset (suitable for the task) was sourced from an open-source GitHub repository (moshesipper/vae-torch-celeba).
  - **Stable Diffusion:** The runwayml/stable-diffusion-v1-5 checkpoint was used for the state-of-the-art comparison.

3.2 Stage 1: Baseline Image Generation with VAE

This step is for setting up the VAE to generate a reconstruction from the provided input image.

1. **Model Loading:** The pre-trained VAE model weights (.pth file) were loaded into the model architecture defined in the source repository. Due to the model being saved with an older version of PyTorch, a safe_globals context was used during loading to prevent unpickling errors.
2. **Image Preprocessing:** The input image was transformed using the prescribed celeb_transform, which resizes the image to the model's expected input dimensions, performs a center crop, and converts it to a PyTorch tensor.
3. **Encoding:** The preprocessed image tensor was passed through the VAE's encoder. This outputs the two vectors defining the image's latent distribution: the mean (μ) and the log-variance ($\log_v ar$).
4. **Decoding:** A single latent vector $\mathbf{z}$ was sampled from this distribution using the reparameterization trick. This vector was then passed through the VAE's decoder to produce the final reconstructed image tensor.

3.3 Stage 2: Experimenting for Diversity Control

The primary goal was to generate *diverse* images. The baseline process can interduce variation, but we wanted a method for controlling this diversity.

1. **Problem:** The amount of variation in the output image is proportional to the standard deviation of the sampled latent vectors. This is determined by the $\log_v ar$

learned by the model, which prioritizes reconstruction accuracy, leading to low variance and low diversity.

2. **Solution:** To control diversity, a scalar diversity_factor is used. This factor directly multiplies the standard deviation (σ) during the reparameterization step, effectively "amplifying" the effect of the random noise ε.

3. **Modified Formula:** The reparameterization formula was adjusted as follows:

$$\mathbf{z} = \mu + (\varepsilon \cdot \sigma \cdot \text{diversity\_factor})$$

4. **Experiments:** To find the best diversity_factor. A loop was created to iterate through a range of factors from 1.0 to 10.0. For each factor, a grid of 10 new images was generated and visually inspected. So we can find the best trade-off between creative diversity and thematic coherence with the original input.

3.4 Stage 3: State-of-the-Art Comparative Analysis

A primary limitation of standard VAEs is generating blurry, low-fidelity images, **resulting from pixel-wise reconstruction loss**. To show this weakness against current standards, a comparative analysis with a state-of-the-art model is done. Using Stable Diffusion to have a clear benchmark for what is possible in generating images.

1. **Methodology:** The Stable Diffusion Image-to-Image pipeline was used. The same input image was given to the pipeline along with a descriptive text prompt ("a high-resolution photograph of a person's face...").

2. **Parameter Control:** The strength parameter was used to control how much the output should adhere to the input image, while the generator's seed was varied to produce 10 unique outputs, analogous to the VAE's sampling process.

4 Experimental Results and Analysis

4.1 The Impact of the Diversity Factor

- **Low Diversity Factors (1.0 - 3.0):** At low factors, the generated images were highly coherent and thematically consistent with the input image. However, they didn't give significant diversity, with variations being limited to small changes.

- **Medium Diversity Factors (4.0 - 7.0):** This range provided a good balance. The images startes to show more meaningful variations in features such as head pose, facial expression and facial structure, while still clearly belonging to the same "domain" of faces.

- **High Diversity Factors (8.0 and above):** At high factors, the model produced highly diverse images. However, this came at the cost of reduced quality and coherence. The generated faces started to exhibit unrealistic features, and a loss of thematic consistency with the original input.

Conclusion of Experiment: Based on this visual analysis, a diversity_factor of **7.0** was identified as providing the optimal balance between creative diversity and image quality for this assignment.

4.2 Final VAE-Generated Images (Deliverable)

Figure 1: Original base image

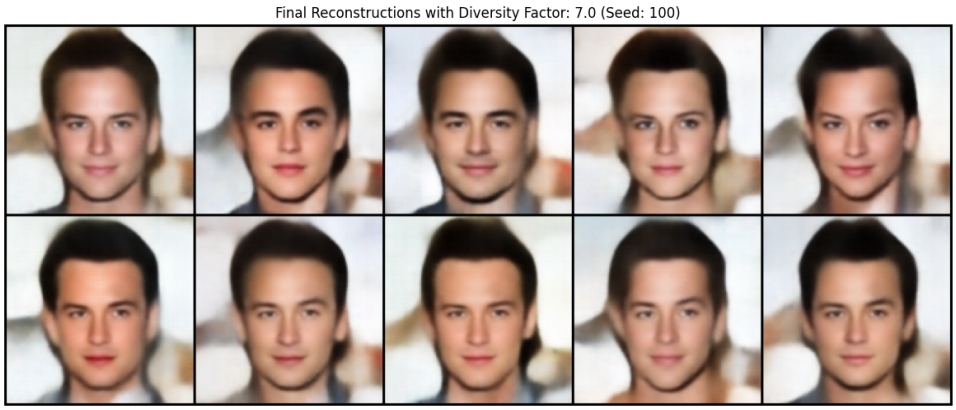

Figure 2: Final VAE-Generated Images with diversity_factor of **7.0**

You can see the rest of diversity_factors outputs in the notebook

4.3 Analysis of VAE Limitations: Image Blurriness

Root Cause Analysis: This blurriness is a direct result of the VAE's optimization objective. The model is trained using a **pixel-wise reconstruction loss**, such as Mean Squared Error (MSE) or Binary Cross-Entropy (BCE). This loss function penalizes the model based on the average difference between the pixels of the reconstructed image and

the original image. When the model is uncertain about a high-frequency detail (like the texture of skin or a sharp edge of a shadow), its mathematically "safest" strategy to minimize the average error is to predict the mean of all plausible pixel values. This averaging process inherently smooths out sharp details, resulting in a blurry output.

4.4 Comparative Analysis with Stable Diffusion

To visually demonstrate this limitation and contextualize the VAE's performance, a comparative generation was performed using the Stable Diffusion v1.5 model.

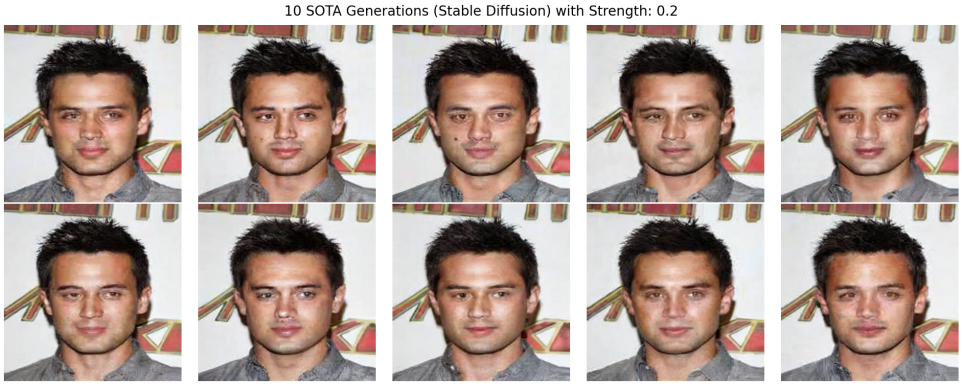

Figure 3: Stable Diffusion images

Analysis: The images generated by Stable Diffusion are photorealistic, sharp, and contain intricate, high-frequency details that are entirely absent in the VAE outputs. This is because modern architectures like Diffusion Models and GANs do not use simple pixel-wise losses. Instead, they are trained with more complex objectives (such as adversarial losses or noise prediction) that compel them to produce statistically realistic and sharp textures.

5 Conclusion

We used pre-trained Variational Autoencoder (VAE) to generate ten diverse facial images from a single input. then an experimental approach was used to control the output diversity by scaling the latent standard deviation. Found inherented limitation of producing blurry images, a direct result of its pixel-wise reconstruction loss. To contextualize this, a comparative analysis was performed with a state-of-the-art Diffusion Model (Stable Diffusion), highlighting modern solutions to this problem.