Project1 - EE798R

Kartik Soni–210496

October 24

The Imagic method involves three main stages:

1. **Text Embedding Optimization**: Optimize the text embeddings to better match the input image.
2. **Model Fine-tuning**: Fine-tune the diffusion model (UNet) to reconstruct the input image when conditioned on the optimized embeddings.
3. **Text Embedding Interpolation and Image Generation**: Interpolate between the optimized and target text embeddings to generate the edited image using the Stable Diffusion Image-to-Image Pipeline.

In preparation, the target text prompt, for example, "A bird sitting on hand" is tokenized using the tokenizer. The corresponding initial text embeddings $\mathbf{e}_{\text{tgt}}$ are obtained using the text encoder.

To find an optimized text embedding $\mathbf{e}_{\text{opt}}$ that, when used in the diffusion model, reconstructs the input image as closely as possible, we follow these steps:

1. **Initialization**: Clone the target embeddings $\mathbf{e}_{\text{tgt}}$ to create a trainable embedding $\mathbf{e}_{\text{opt}}$.
2. **Latent Encoding**: Encode the input image into the latent space using the VAE encoder to obtain latent representations $\mathbf{z}$.
3. **Noise Addition**: Add random noise ϵ to the latent representations at random timesteps t to simulate the diffusion process.
4. **Noise Prediction**: Use the UNet model to predict the added noise $\epsilon_\theta(\mathbf{z}_t, t, \mathbf{e}_{\text{opt}})$.
5. **Loss Computation**: Calculate the mean squared error (MSE) between the predicted noise and the actual noise.
6. **Optimization**: Update $\mathbf{e}_{\text{opt}}$ by minimizing the loss function over several iterations.

The mathematical formulation for this process is:

$$\mathcal{L}_{\text{emb}} = E_{t,\epsilon}\left[\|\epsilon - \epsilon_\theta(\mathbf{z}_t, t, \mathbf{e}_{\text{opt}})\|_2^2\right]$$

Next, we fine-tune the UNet model so that it accurately reconstructs the input image when conditioned on the optimized embeddings $\mathbf{e}_{\text{opt}}$. This process involves freezing the parameters of the VAE and text encoder to prevent them from being updated during fine-tuning. Noise is added to the latent representations, and the UNet model predicts the noise, updating its parameters θ. The loss is computed, and the UNet model parameters are updated to minimize the loss over several iterations.

The mathematical formulation for model fine-tuning is:

$$\mathcal{L}_{\text{model}} = E_{t,\epsilon}\left[\|\epsilon - \epsilon_\theta(\mathbf{z}_t, t, \mathbf{e}_{\text{opt}})\|_2^2\right]$$

For text embedding interpolation and image generation, an interpolated embedding $\mathbf{e}_{\text{interp}}$ is computed to balance between the original image content and the desired edit specified by the target text prompt.

$$\mathbf{e}_{\text{interp}} = \eta \cdot \mathbf{e}_{\text{tgt}} + (1 - \eta) \cdot \mathbf{e}_{\text{opt}}$$

Where $\eta \in [0, 1]$ is an interpolation parameter controlling the influence of the target text.

To use the interpolated embeddings within the Stable Diffusion Image-to-Image Pipeline, we override the pipeline's internal method for encoding prompts to return $\mathbf{e}_{\text{interp}}$ when the target text prompt is used. After image generation, we restore the pipeline's original prompt encoding method.

The final image is expected to reflect the desired modifications specified by the target text prompt while retaining significant features and details from the original image.

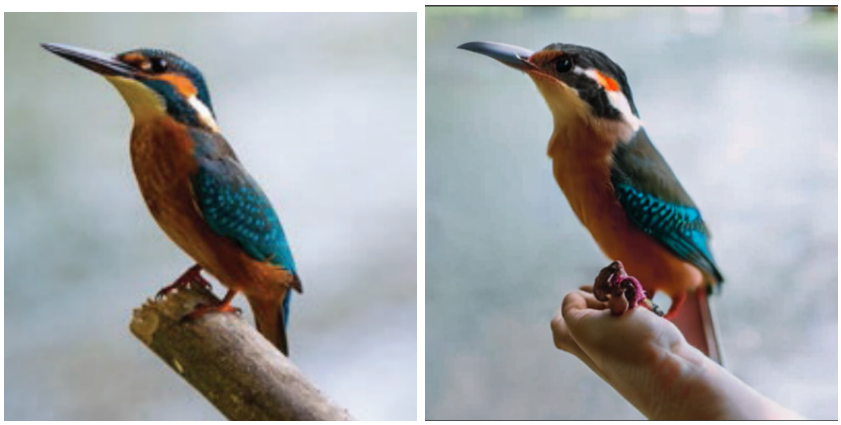

Figure 1: Comparison between the original and edited images.