Brightfield to Red and Green Fluorescence Image Translation using Conditional Diffusion

By Mayank Kunwar

Data Preparation:

- Load triplets (BF, Red, Green)
- Convert all images to 256x256, normalize to [-1,1]
- For each sample:
  a) BF = condition image
  b) Fluorescence (red or green) = target
  c) Condition vector = [1,0] (red) or [0,1] (green)

Input construction:

For every training step build a 6-channel input:

- Noisy fluorescence image (1 channel)
- BF RGB (3 channel)
- Condition one-hot map (2 channel)

Noise addition (forward diffusion):

- Sample a random timestep t
- Add DDPM noise to target fluorescence $x_t = \sqrt{\alpha_t} x_0 + \sqrt{(1-\alpha_t)}\epsilon$
- Goal: Model learns top predict ϵ from x_t

UNet forward pass (epsilon prediction):

- The 6-channel input is passed through the UNet
- UNet predicts the added noise $\hat{\epsilon} = \epsilon_\theta(x_t, t, BF, c)$
- Reconstruct a clean fluorescence estimate $\hat{x}_0 = \frac{x_t - \sqrt{(1-\alpha_t)}\hat{\epsilon}}{\sqrt{\alpha_t}}$

Loss computation:

Two losses guide the diffusion model:

- L1-noise loss: $L_{denoise} = \left||\hat{\epsilon} - \epsilon|\right|_1$
- Perceptual loss (VGG16): $L_{perc} = \left||VGG(\hat{x}_0) - VGG(x_0)|\right|_1$

$$Total\ loss = L_{denoise} + 0.01 * L_{perc}$$

Optimization:

- AdamW optimizer
- Update UNet parameters
- Update EMA weights for smooth and stable inference $\theta_{EMA} = 0.995\theta_{EMA} + 0.005\theta$

Inference (reverse diffusion):

- Start from pure random noise x_T
- Run DDPM reverse steps $x_{t-1} = DDPM\ step(x_t, \epsilon)$

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}}\hat{\epsilon}_t\right) + \sigma_t z, \qquad z \sim \mathcal{N}(0, I)$$

- Each step progressively denoises using the UNet