

# CUSTOMIZED IMAGE GENERATION

## (Style Transfer via Stable Diffusion + LoRA Fine-Tuning)

Môn học: Các Kỹ Thuật Học Sâu và Ứng Dụng – CS431.Q12

Giảng viên: Nguyễn Vinh Tiệp & Chế Quang Huy

### 1. Thông tin nhóm

#### Thành viên:

1. Phan Đức Thành Phát – 23521149
2. Nguyễn Minh Quốc – 23521304
3. Nguyễn Khang Hy – 2352662

### 2. Tổng quan và lý do chọn đề tài

- Các mô hình sinh ảnh như Stable Diffusion đã mở ra khả năng tạo ra hình ảnh mới dựa trên điều kiện tùy chỉnh.
- Đề tài nhóm tập trung vào hướng “Customized Image Generation” – sinh ảnh mới theo phong cách mong muốn, cụ thể là bài toán Style Transfer.
- Hệ thống cho phép người dùng tải ảnh gốc (content) và chọn phong cách (style class), kết hợp fine-tuning mô hình Stable Diffusion bằng kỹ thuật LoRA (Low-Rank Adaptation) để học các phong cách khác nhau.
- Cách tiếp cận này giúp mô hình có thể tùy biến sinh ảnh theo phong cách cụ thể mà không cần sử dụng văn bản (prompt), phù hợp với hướng “customized image generation” hiện đại.

### 3. Phát biểu bài toán

- Mục tiêu: Fine-tune mô hình Stable Diffusion để sinh ảnh theo phong cách cụ thể (style class) dựa trên ảnh gốc (content image).
- Mô hình học phân phối có điều kiện  $p(x | \text{style})$ , cho phép tạo ra ảnh mới giữ bố cục content nhưng mang đặc trưng của style.

#### Input:

- Content\_Image: ảnh gốc giữ bố cục và nội dung chính.
- Style\_Class hoặc Style\_Image: lựa chọn phong cách từ thư viện có sẵn hoặc upload ảnh phong cách.

- Tùy chọn: *style\_strength*, *mask* vùng áp style.

### Output:

- Ảnh mới giữ bố cục content và mang phong cách tương ứng.

## 4. Hướng tiếp cận

- Sử dụng mô hình Stable Diffusion v1.5 làm nền tảng.
- Fine-tuning bằng kỹ thuật LoRA (Low-Rank Adaptation) để thêm phong cách mới mà không cần huấn luyện lại toàn bộ mô hình. Nhóm chỉ fine-tune phần UNet trong Stable Diffusion bằng LoRA để học phong cách từ tập ảnh WikiArt, thay vì huấn luyện lại toàn bộ mô hình.  
*(UNet là phần chịu trách nhiệm sinh ảnh từ không gian latent, nên việc tinh chỉnh tại đây giúp mô hình học nhanh đặc trưng phong cách mà vẫn giữ được chất lượng ảnh gốc.)*
- So sánh LoRA với các kỹ thuật khác như DreamBooth, Textual Inversion:
  - + LoRA: nhẹ, dễ huấn luyện, thêm nhanh phong cách mới.
  - + DreamBooth: học style hoặc subject cụ thể nhưng tốn tài nguyên hơn.
  - + Textual Inversion: chỉ học embedding đơn giản cho style.

## 5. Pipeline chi tiết

1. Chuẩn bị dữ liệu:
  - Content: COCO 2017 (ảnh thật, bố cục tự nhiên).
  - Style: WikiArt (3–5 phong cách, 50–100 ảnh/style).
2. Fine-tune LoRA:
  - Fine-tune các attention layer trong UNet.
  - Mỗi phong cách ~5k–8k bước, batch 2–4, learning rate 1e-4.
  - Sử dụng AdamW, scheduler cosine, GPU T4/A100.
3. Inference:
  - Encode Content\_Image → latent vector.
  - Áp dụng LoRA checkpoint (style) vào UNet.
  - Decode → ảnh mới mang phong cách đã học.

## 6. Cấu hình huấn luyện LoRA (dự tính)

Thành phần	Cấu hình
Base model	runwayml/stable-diffusion-v1-5
Fine-tune target	UNet (attention layers)
Rank	4
Learning rate	1e-4
Batch size	2–4
Steps	5,000–8,000
Optimizer	AdamW
Scheduler	Cosine
Training time	2–3 giờ/style (Colab T4/A100)

## 7. Hàm mất mát (Loss Functions)

Không dùng prompt, nên bỏ CLIP text loss.

Tổng loss kết hợp ba thành phần:

$$L_{\text{total}} = \alpha \cdot L_2 + \beta \cdot \text{LPIPS} + \gamma \cdot \text{StyleLoss}$$

- L2 loss: tái tạo chi tiết ảnh.
  - LPIPS: duy trì độ tự nhiên theo cảm nhận người nhìn.
  - Style loss (Gram matrix): giữ họa tiết, màu sắc của style.
- Giá trị  $\alpha$ ,  $\beta$ ,  $\gamma$  được tinh chỉnh qua thực nghiệm.

## 8. Độ đo đánh giá

- FID (Fréchet Inception Distance): đo độ “thật” của ảnh sinh ra.
- LPIPS (Learned Perceptual Image Patch Similarity): đo cảm nhận giữa ảnh output và style.
- SSIM (Structural Similarity Index): đo độ giữ cấu trúc content.
- Runtime, GPU memory, kích thước model để đánh giá tính thực tế.

## 9. Demo & Triển khai

Giao diện Gradio:

1. Upload ảnh content.
2. Chọn style class (đã fine-tune LoRA sẵn).
3. Chỉnh style\_strength, mask nếu cần.
4. Xem ảnh kết quả và tải về.

Triển khai demo: Hugging Face Spaces hoặc Colab.

## 10. Kế hoạch & phân công

- Nguyễn Khang Hy: EDA, đánh giá, tổng hợp kết quả và báo cáo.

- Nguyễn Minh Quốc: Fine-tuning LoRA, tối ưu pipeline huấn luyện.
- Phan Đức Thành Phát: Tích hợp mô hình và xây dựng giao diện demo.

## **11. Kết quả kỳ vọng**

- Fine-tune thành công 3–5 phong cách (VD: Monet, Ukiyo-e, Pop Art, Sketch, Minimalism).
- Ảnh sinh ra đạt FID < 60, LPIPS thấp, SSIM cao.
- Demo chạy ổn định, thời gian suy luận < 5s/ảnh.