# Report White Balance

Report day: 24/03/2025

Name: Nguyễn Thanh Hòa

Team: 2 Team leader: Hòa

Team member: Nguyễn Thanh Hòa, Nguyễn Đình Vũ Hải, Nguyễn Quý Toàn

✅ **Task title 1: InstructPix2Pix for White Balance**

# ▼ 1. Purpose

Triển khai mô hình **InstructPix2Pix** đã được convert sang định dạng **ONNX**, nhằm phục vụ cho tác vụ điều chỉnh cân bằng trắng (white balance) trong ảnh.

# ▼ 2. Action

- ✅ Convert mô hình InstructPix2Pix sang ONNX, tách riêng các thành phần: `vae_encoder`, `unet`, `vae_decoder`.
- ✅ Giữ nguyên `CLIPTokenizer` và `CLIPTextModel` ở PyTorch.
- ✅ Xây dựng server FastAPI, thực hiện:
  - Encode prompt bằng `CLIPTokenizer` và `CLIPTextModel`.
  - Encode ảnh input qua `vae_encoder`
  - Diffusion loop sử dụng `unet`
  - Decode latent thành ảnh bằng `vae_decoder`
- ✅ Kết nối và gửi request thành công đến Triton Inference Server
- Quy trình:
- Triển khai **FastAPI** để xây dựng API nhận ảnh đầu vào và prompt từ người dùng.
- Sử dụng mô hình **StableDiffusionInstructPix2PixPipeline** để xử lý ảnh.
- Kết nối với **Triton Inference Server** để thực hiện các bước tính toán chính:
  - **encode_vae_latents**: Mã hóa ảnh thành latent representation.
  - **run_unet**: Dự đoán noise và điều chỉnh ảnh dựa trên prompt.
  - **decode_vae_latents**: Chuyển latent representation thành ảnh đầu ra.

- Áp dụng **Classifier-Free Guidance** để cải thiện chất lượng ảnh bằng cách kết hợp các embedding của prompt và negative prompt.
- Lưu ảnh đầu vào, ảnh trung gian và ảnh đầu ra vào thư mục tương ứng (`input/`, `intermediate/`, `output/`).
- Tối ưu bộ nhớ GPU bằng cách gọi `torch.cuda.empty_cache()` giữa các bước xử lý.

## ▼ 3. Result

- ✅ Hoàn thành API inference sử dụng **FastAPI** kết nối với **Triton Inference Server**.
- ✅API hoạt động ổn định, xử lý ảnh nhanh và đảm bảo kết quả cân bằng sáng chất lượng cao.
- ✅ Lưu trữ kết quả inference để kiểm tra và debug nếu cần thiết.
- ✅ Kết quả cân bằng trắng đúng với hướng dẫn prompt:
  - "make the image brighter and correct the white balance" →Làm sáng + cân bằng trắng.
  - "correct lighting and make colors more natural" → Giống auto tone Adobe.⇒tối ưu hơn.
- ✅ API trả về JSON gồm: prompt, thời gian xử lý, ảnh input (base64), ảnh output (base64).

## ▼ 4. Upcomming

☐ Triển khai tính năng **batch inference** để tăng tốc khi xử lý nhiều ảnh.

☐ Tối ưu lại pipeline để giảm thời gian loop diffusion (tuning scheduler).

☐ Đóng gói và triển khai hệ thống.

☐ Đánh giá độ chính xác cân bằng trắng bằng công cụ đo lường khách quan.

✅ **Task title 2: Quantize (fp16, int8, int4)**

## ▼ 1. Purpose

Thực hiện ba **Quantize**: fp16, int8 và int4.

Giảm kích thước mô hình và cải thiện hiệu suất

Đảm bảo khả năng tương thích với FastAPI và Docker.

## ▼ 2. Action

Chuyển đổi các thành phần của mô hình (text_encoder, unet, vae) sang định dạng ONNX.

- Gặp vấn đề với model Unet khi chuyển sang định dạng Onnx

Áp dụng **Quantize:**

- Áp dụng **Quantize** fp16 cho mô hình ONNX.
- Thử nghiệm **Quantize** int8 bằng ONNX Runtime.
- Nghiên cứu khả năng áp dụng **Quantize** int4.

## ▼ 3. Result

Gặp một số mất mát nhỏ về độ chính xác,cần tinh chỉnh thêm

Cần kiểm thử thêm để đánh giá

## ▼ 4. Upcomming

Cân bằng giữa độ chính xác và hiệu suất.

Đo lường sự khác biệt về hiệu suất giữa các mức