BÁO CÁO LAB 2
Tinh chỉnh mô hình DeepSeek-OCR trên Dữ liệu tiếng Việt

Bàng Mỹ Linh
MSSV: 23122009
Khoa Công Nghệ Thông Tin
Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM
23122009@student.hcmus.edu.vn

Abstract

Nhận dạng chữ viết tay tiếng Việt là một bài toán thách thức do sự phức tạp của hệ thống dấu thanh và sự đa dạng trong phong cách viết. Đồ án này trình bày quá trình tinh chỉnh mô hình DeepSeek-OCR, một mô hình thị giác-ngôn ngữ tiên tiến, trên tập dữ liệu UIT-HWDB. Kỹ thuật LoRA (Low-Rank Adaptation) được áp dụng kết hợp với thư viện Unsloth để tối ưu hóa quá trình huấn luyện trên tài nguyên tính toán giới hạn. Kết quả thực nghiệm cho thấy mô hình sau khi tinh chỉnh đạt được sự cải thiện vượt bậc: Tỷ lệ lỗi ký tự (CER) giảm từ **810.95%** (mô hình gốc) xuống còn **32.41%** trên tập test. Báo cáo này phân tích chi tiết quy trình thực hiện, cấu hình siêu tham số, và các mẫu lỗi phổ biến nhằm đóng góp vào các nghiên cứu tương lai về OCR tiếng Việt.

1 Giới thiệu và Nền tảng

1.1 OCR và những thách thức với dữ liệu tiếng Việt

Nhận dạng ký tự quang học (OCR) cho chữ viết tay vẫn là một lĩnh vực nghiên cứu mở. Đối với tiếng Việt, bài toán này càng trở nên khó khăn hơn do các đặc điểm ngôn ngữ học:

- **Hệ thống dấu phức tạp**: Tiếng Việt có 6 thanh điệu và nhiều nguyên âm có dấu phụ (ă, â, ê, ô, ơ, ư). Việc nhận dạng sai hoặc bỏ sót các dấu nhỏ này dẫn đến sai lệch hoàn toàn về ngữ nghĩa.
- **Đa dạng phong cách viết**: Chữ viết tay tự do có độ biến thiên lớn về độ nghiêng, kích thước và cách nối nét.
- **Thiếu hụt dữ liệu**: So với tiếng Anh hay tiếng Trung, các bộ dữ liệu gán nhãn quy mô lớn cho chữ viết tay tiếng Việt còn hạn chế.

1.2 Mô hình DeepSeek-OCR

DeepSeek-OCR (Wei et al., 2025) là một mô hình thị giác-ngôn ngữ (Vision-Language Model) được thiết kế với triết lý "nén ngữ cảnh quang học" (Contexts Optical Compression), nhằm mục đích xử lý văn bản dài thông qua biểu diễn hình ảnh nén hiệu quả. Kiến trúc mô hình bao gồm hai thành phần chính được tối ưu hóa đồng bộ:

- **Vision Encoder (DeepEncoder)**: Đây là bộ mã hóa thị giác được thiết kế riêng với khoảng **380 triệu tham số**, giải quyết thách thức về độ phân giải cao và chi phí tính toán. DeepEncoder có cấu trúc lai ghép nối tiếp gồm hai module:
  1. **Module Perceptio**: Sử dụng kiến trúc **SAM-base** (80M tham số) với cơ chế window attention để trích xuất các đặc trưng chi tiết cục bộ từ ảnh độ phân giải cao.
  2. **Module Knowledge**: Sử dụng **CLIP-large** (300M tham số) với global attention để nắm bắt ngữ nghĩa tổng quát.

  Điểm đặc biệt là giữa hai module này có một **Bộ nén Convolutional 16x**. Bộ nén này giảm số lượng token thị giác xuống 16 lần trước khi đưa vào module CLIP, giúp mô hình xử lý được ảnh kích thước lớn (ví dụ: 1024x1024) chỉ với số lượng token rất nhỏ (256 token), đảm bảo hiệu quả bộ nhớ tối ưu.
- **Language Decoder**: Sử dụng kiến trúc **DeepSeek3B-MoE-A570M**, một mô hình ngôn ngữ lớn dạng Mixture-of-Experts. Mặc dù có tổng cộng 3 tỷ tham số, nhưng trong mỗi lần suy luận, mô hình chỉ kích hoạt khoảng **570 triệu tham số** (kích hoạt 6/64 chuyên gia định tuyến và 2 chuyên gia chia sẻ). Cơ chế này giúp bộ giải mã đạt được sức mạnh biểu diễn của mô hình lớn trong khi vẫn

duy trì tốc độ suy luận nhanh của mô hình nhỏ.

Cơ chế "Context Optical Compression" cho phép DeepSeek-OCR đạt được tỷ lệ nén thông tin văn bản cực cao (lên tới 7-20 lần so với token văn bản truyền thống) mà vẫn giữ được độ chính xác nhận dạng (OCR precision) lên tới 97% ở tỷ lệ nén 10x.

1.3 Mục tiêu Đồ án

Mục tiêu chính của đồ án này là:

1. Xây dựng pipeline tinh chỉnh mô hình DeepSeek-OCR cho dữ liệu tiếng Việt.
2. Đánh giá hiệu quả của kỹ thuật LoRA trong việc thích nghi mô hình ngôn ngữ lớn với ngôn ngữ mới (tiếng Việt).
3. Phân tích chi tiết các lỗi sai để đề xuất hướng cải thiện.

2 Phương pháp

2.1 Tập dữ liệu

2.1.1 Tập dữ liệu UIT-HWDB

Tập dữ liệu UIT-HWDB (University of Information Technology - Handwritten Database) (Nguyen et al., 2021) được sử dụng cho đồ án này. Tập dữ liệu này chứa văn bản viết tay tiếng Việt ở ba mức độ phức tạp:

- **Cấp từ (UIT-HWDB-word)**: Các từ đơn lẻ (110,745 mẫu)
- **Cấp dòng (UIT-HWDB-line)**: Các dòng văn bản đơn (7,273 mẫu)
- **Cấp đoạn (UIT-HWDB-paragraph)**: Các đoạn văn nhiều dòng (1,144 mẫu)

2.1.2 Xử lý dữ liệu

Quy trình chuẩn bị và xử lý dữ liệu được thực hiện qua các bước sau:

1. **Chiến lược lấy mẫu**: Do tập dữ liệu UIT-HWDB rất lớn, chiến lược lấy mẫu ngẫu nhiên được áp dụng để đảm bảo tính đa dạng về người viết nhưng vẫn phù hợp với tài nguyên tính toán:
   - `max_folders=200`: tương ứng với 200 người viết khác nhau cho mỗi loại dữ liệu.
   - `max_per_folder=20`: tương ứng với 20 ảnh ngẫu nhiên từ mỗi thư mục con.
   - Quy trình này áp dụng cho cả ba tập con: *Line*, *Paragraph*, và *Word*.
2. **Conversation Formatting**: Dữ liệu được chuyển đổi sang định dạng conversation tương thích với mô hình DeepSeek-OCR.

   Mỗi mẫu dữ liệu được biểu diễn dưới dạng một danh sách các message, trong đó mỗi message bao gồm trường `role` và `content`. Riêng message của người dùng còn chứa thêm trường `images` để cung cấp dữ liệu hình ảnh cho mô hình.

   Cụ thể, mỗi conversation bao gồm hai message:
   - *User message*:
     - **role**: `<|User|>`
     - **content**: `<image>\n Free OCR.`
     - **images**: ảnh đầu vào tương ứng
   - *Assistant message*:
     - **role**: `<|Assistant|>`
     - **content**: văn bản nhãn Ground Truth
3. **Mã hóa dữ liệu (Tokenization)**: Sử dụng `DeepSeekOCRDataCollator` để mã hóa văn bản và hình ảnh. Đặc biệt, quy trình áp dụng kỹ thuật **Masking** trên các token của người dùng (User prompt), đảm bảo mô hình chỉ tính toán loss và học dựa trên câu trả lời của trợ lý (Assistant response).

2.1.3 Phân chia tập dữ liệu

Dựa trên chiến lược lấy mẫu nêu trên, tổng số lượng mẫu thu được cho quá trình huấn luyện là **8,537** mẫu. Toàn bộ tập dữ liệu đã lấy mẫu này được áp dụng cho quá trình huấn luyện và đánh giá trên một tập `test` (sample từ các tập test của `UIT-HWDB`). Quy trình chuẩn bị tập `test` cũng tương tự tập train, với `max_folders=20` và `max_per_folder=5`.

2.1.4 Xử lý hình ảnh

Quy trình xử lý ảnh được thực hiện thông qua `DeepSeekOCRDataCollator` với cơ chế xử lý độ phân giải cao:

- **Kích thước cơ sở (Global View)**: 1024×1024 pixels. Hình ảnh được đệm (pad) để giữ nguyên tỷ lệ khung hình.

Tập con	Số mẫu	Mô tả
Train	8,537	Dữ liệu lấy mẫu từ UIT-HWDB-(word/line/paragraph)/../train
Test	231	Dữ liệu lấy mẫu từ UIT-HWDB-(word/line/paragraph)/../test

Bảng 1: Thống kê số lượng mẫu dữ liệu thực tế

- **Kích thước patch (Local View)**: 640×640 pixels.
- **Cắt động (Dynamic Crop)**: Được kích hoạt (`crop_mode=True`). Ảnh đầu vào có thể được cắt thành từ 2 đến 9 patches tùy thuộc vào tỷ lệ khung hình và độ phân giải, giúp mô hình nắm bắt chi tiết tốt hơn đối với các văn bản dài hoặc nhỏ.
- **Chuẩn hóa**: Giá trị pixel được chuẩn hóa với Mean=(0.5, 0.5, 0.5) và Std=(0.5, 0.5, 0.5).

2.2 Cấu hình tinh chỉnh

2.2.1 Tham số LoRA

Trong quá trình tinh chỉnh mô hình, kỹ thuật LoRA (Low-Rank Adaptation (Hu et al., 2021)) được áp dụng nhằm giảm số lượng tham số cần huấn luyện và tiết kiệm tài nguyên tính toán. Thay vì cập nhật toàn bộ trọng số của mô hình, LoRA chèn các ma trận hạng thấp vào một số lớp tuyến tính đã chọn, cho phép mô hình thích nghi với dữ liệu mới trong khi vẫn giữ nguyên các trọng số gốc.

Chi tiết cấu hình LoRA được mô tả trong Bảng 2.

Cụ thể, LoRA được áp dụng lên các lớp chiếu tuyến tính trong cơ chế self-attention (`q_proj`, `k_proj`, `v_proj`, `o_proj`) và các lớp tuyến tính trong khối feed-forward (`gate_proj`, `up_proj`, `down_proj`). Hạng LoRA được thiết lập là $r = 16$, với hệ số mở rộng $\alpha = 16$, đảm bảo cân bằng giữa khả năng biểu diễn và nguy cơ quá khớp. Quy trình không sử dụng dropout trong LoRA và không áp dụng các biến thể như Rank-Stabilized LoRA hay LoftQ.

2.2.2 Siêu tham số huấn luyện

Giá trị của các siêu tham số được mô tả chi tiết trong Bảng 3

Tham số	Giá trị
Hạng LoRA (r)	16
Hệ số LoRA (α)	16
LoRA dropout	0.0
Bias	none
Seed ngẫu nhiên	3407
Rank-Stabilized LoRA	Không sử dụng
LoftQ	Không sử dụng
Module đích	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj

Bảng 2: Cấu hình LoRA sử dụng trong quá trình tinh chỉnh mô hình

Siêu tham số	Giá trị
Kích thước batch (mỗi thiết bị)	2
Bước tích lũy gradient	4
Kích thước batch hiệu dụng	16
Tốc độ học	1e-4
Bộ tối ưu	AdamW (8-bit)
Suy giảm trọng số	0.001
Bộ lập lịch LR	Linear
Bước khởi động (Warmup)	5
Số epoch huấn luyện	1
Độ dài chuỗi tối đa	1024
Seed ngẫu nhiên	3407

Bảng 3: Siêu tham số huấn luyện

2.3 Kỹ thuật tối ưu hóa

Để giải quyết bài toán huấn luyện mô hình đa phương thức lớn trên tài nguyên hạn chế (Kaggle GPU – NVIDIA Tesla T4 16GB VRAM), tổ hợp các kỹ thuật tối ưu hóa tiên tiến dựa trên thư viện Unsloth và Hugging Face Transformers được áp dụng:

1. **Tối ưu hóa Kernel với Unsloth (Unsloth Team, 2024)**: Sử dụng `FastVisionModel` từ thư viện Unsloth, thay thế các lớp tính toán tiêu chuẩn của PyTorch bằng các kernel Triton được viết lại thủ công. Kỹ thuật này giúp tối ưu hóa quá trình lan truyền ngược, tăng tốc độ huấn luyện lên khoảng 2 lần và giảm mức tiêu thụ bộ nhớ VRAM tới 60% so với phương pháp truyền thống mà không làm giảm độ chính xác của mô hình.
2. **Kỹ thuật QLoRA (Quantized Low-Rank Adaptation (Dettmers et al., 2023))**: Áp

dụng QLoRA thay vì tinh chỉnh toàn bộ tham số:

- **Lượng tử hóa 4-bit**: Mô hình gốc được tải và đóng băng trọng số ở định dạng 4-bit (NF4 - Normal Float 4), giúp giảm kích thước mô hình trong bộ nhớ xuống 4 lần.
- **LoRA Adapters**: Chỉ các ma trận hạng thấp được thêm vào các lớp Attention và MLP là có thể huấn luyện. Cấu hình chi tiết đã nêu ở Bảng 2

3. **Chiến lược bộ nhớ**:
   - **Tối ưu hóa AdamW (Loshchilov and Hutter, 2017) 8-bit**: Sử dụng bộ tối ưu `adamw_8bit` thay vì 32-bit tiêu chuẩn. Kỹ thuật này giảm lượng bộ nhớ cần thiết để lưu trữ trạng thái của bộ tối ưu xuống còn 1/4, giải phóng VRAM đáng kể cho dữ liệu ảnh độ phân giải cao.
   - **Gradient Accumulation (Lamy-Poirier, 2021)**: Do kích thước ảnh lớn (1024x1024) chiếm nhiều bộ nhớ, các siêu tham số `per_device_train_batch_size=2` và `gradient_accumulation_steps=4` được thiết lập. Điều này mô phỏng kích thước batch hiệu dụng là 8 (hoặc 16 tùy số lượng GPU) nhưng chỉ chiếm kích thước tương đương với `batch_size=2`, cho phép huấn luyện trên GPU hạn chế bộ nhớ mà không làm thay đổi đáng kể hành vi hội tụ của mô hình.

3 Thiết kế Thí nghiệm

3.1 Các chỉ số đánh giá

3.1.1 Tỷ lệ lỗi ký tự (Character Error Rate – CER)

Chỉ số đánh giá chính được sử dụng trong nghiên cứu này là **tỷ lệ lỗi ký tự (Character Error Rate - CER)**, được tính theo công thức:

$$\text{CER} = \frac{S + D + I}{N} \tag{1}$$

trong đó:

- S là số lượng ký tự bị thay thế (substitution),
- D là số lượng ký tự bị xóa (deletion),
- I là số lượng ký tự bị chèn (insertion),
- N là tổng số ký tự trong văn bản tham chiếu (ground truth).

Trong thực nghiệm, CER được tính trực tiếp giữa chuỗi văn bản dự đoán và chuỗi tham chiếu ở cấp độ ký tự, và được sử dụng làm thước đo chính để so sánh hiệu suất giữa các mô hình OCR.

3.1.2 Phân loại lỗi và đánh giá tỉ lệ khớp hoàn hảo

Ba loại lỗi Insertion, Deletion và Substitution được thông kê chi tiết để đánh giá khuynh hướng sai lệch của mô hình và cải thiện sau khi tinh chỉnh. Bên cạnh đó, tỉ lệ khớp hoàn hảo (CER=0%) cũng được tính toán để đo khả năng dự đoán hoàn hảo của mô hình.

3.1.3 Phân tích định tính kết quả dự đoán

Ngoài giá trị CER trung bình và phân loại lỗi, kết quả dự đoán cũng được phân tích qua các thống kê mô tả và phân tích định tính, bao gồm:

- Phân phối giá trị CER trên tập kiểm tra (trung bình, trung vị, độ lệch chuẩn).
- Giá trị CER nhỏ nhất và lớn nhất quan sát được.
- Phân phối hiệu suất theo ngưỡng CER.
- Phân tích các mẫu có CER thấp nhất và cao nhất thông qua so sánh trực tiếp giữa văn bản dự đoán và văn bản tham chiếu.

Các phân tích này nhằm làm rõ hành vi của mô hình trong các trường hợp dự đoán tốt và kém, từ đó cung cấp cái nhìn định tính về chất lượng nhận dạng.

3.2 So sánh với mô hình gốc

- **Mô hình gốc**: mô hình DeepSeek-OCR nguyên bản, không tinh chỉnh, được đánh giá theo chế độ zero-shot.
- **Mô hình tinh chỉnh**: mô hình DeepSeek-OCR được tinh chỉnh bằng kỹ thuật LoRA trên dữ liệu tiếng Việt.

Cả hai mô hình được đánh giá trên cùng một tập kiểm tra độc lập. Để đảm bảo tính nhất quán và khả năng so sánh, số lượng mẫu được sử dụng trong quá trình đánh giá là như nhau cho cả hai mô hình.

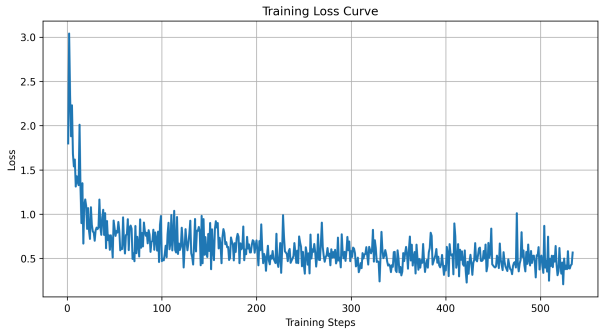
3.3 Quy trình thực hiện

Quy trình thực nghiệm được tiến hành theo các bước sau:

1. Tải mô hình DeepSeek-OCR đã được huấn luyện trước.
2. Thực hiện suy luận trên tập kiểm tra để thu thập kết quả dự đoán của mô hình gốc.
3. Áp dụng các adapter LoRA vào các module đích của mô hình.
4. Huấn luyện mô hình trên tập huấn luyện tiếng Việt.
5. Thực hiện suy luận và đánh giá mô hình tinh chỉnh trên cùng tập kiểm tra.
6. Tổng hợp kết quả, tính toán CER và các thống kê mô tả liên quan.

4 Kết quả

4.1 Đường cong mất mát huấn luyện



Hình 1: Đường cong hội tụ của training loss theo số bước

Đường cong training loss giảm nhanh ở giai đoạn đầu và dần ổn định về sau, cho thấy mô hình hội tụ tốt. Các dao động nhỏ quanh giá trị thấp là bình thường và không ảnh hưởng đến tính ổn định của quá trình huấn luyện.

4.2 Hiệu suất tổng thể

Bảng 4 trình bày kết quả đánh giá tổng thể của hai mô hình trên tập kiểm tra, dựa trên chỉ số CER trung bình.

Kết quả cho thấy mô hình gốc hoàn toàn thất bại với dữ liệu này (CER > 800%). Trong khi đó, mô hình tinh chỉnh đạt mức cải thiện rất lớn về CER so với mô hình gốc, cho thấy hiệu quả của việc thích nghi mô hình với dữ liệu tiếng Việt thông qua LoRA.

Mô hình	Mean CER (%)
Mô hình gốc (Zero-shot)	810.95
Mô hình tinh chỉnh (LoRA)	**32.41**
Mức cải thiện	↓ 778.54

Bảng 4: So sánh CER trung bình trên tập kiểm tra

4.3 Thống kê phân phối CER

Để đánh giá chi tiết mức độ cải thiện và sự ổn định của mô hình, các chỉ số thống kê mô tả của Tỷ lệ lỗi ký tự (CER) giữa mô hình gốc và mô hình sau tinh chỉnh được so sánh trên tập test. Đồng thời, phân phối theo ngưỡng CER cũng được tính toán để đánh giá chi tiết mức độ cải thiện của mô hình sau tinh chỉnh.

4.3.1 Thống kê mô tả

Metric	Baseline	Fine-tuned	Cải thiện
Mean CER	810.95%	**32.41%**	↓ 778.54% (96.0%)
Median CER	60.13%	**12.62%**	↓ 47.51% (79.0%)
Std CER	4645.85%	87.92%	↓ 4557.93%
Min CER	6.06%	**0.00%**	-
Max CER	61100.00%	1188.66%	↓ 59911.34%
Số mẫu	200	200	-

Bảng 5: So sánh thống kê mô tả phân phối CER và mức độ cải thiện giữa hai mô hình

Bảng 5 cho thấy sự khác biệt rất lớn về hiệu suất:

- **Độ chính xác:** CER trung bình giảm mạnh từ 810.95% xuống còn 32.41%. Đáng chú ý, CER trung vị của mô hình tinh chỉnh chỉ là 12.62%, cho thấy trên phần lớn các mẫu dữ liệu, mô hình hoạt động khá hiệu quả. Ngược lại, mô hình gốc có CER trung vị lên tới 60.13%, đồng nghĩa với việc phần lớn văn bản sinh ra đều sai lệch nghiêm trọng so với nhãn gốc.
- **Độ ổn định:** Độ lệch chuẩn của mô hình gốc cực kỳ lớn (4645.85%), phản ánh sự bất ổn định cao và sự xuất hiện của các trường hợp lỗi "ảo giác" nghiêm trọng (CER tối đa lên tới 61100%). Sau khi tinh chỉnh, độ lệch chuẩn giảm xuống còn 87.92%, cho thấy mô hình dự đoán ổn định hơn nhiều.
- **Khả năng khớp hoàn hảo:** Mô hình tinh chỉnh đạt được CER tối thiểu là 0.00%, tức là có khả năng nhận dạng chính xác hoàn toàn một số mẫu, điều mà mô hình gốc không làm được (CER tối thiểu 6.06%).

4.3.2 Phân phối hiệu suất theo ngưỡng CER

Để đánh giá tính ứng dụng thực tế của mô hình, tỷ lệ các mẫu đạt được ngưỡng CER nhất định được phân tích. Kết quả so sánh giữa mô hình gốc và mô hình tinh chỉnh được trình bày trong Bảng 6.

Ngưỡng CER	Baseline	Fine-tuned	Thay đổi
$\leq$ 5% (Xuất sắc)	0 (0.0%)	**57 (28.5%)**	$\uparrow$ 57 mẫu
$\leq$ 10% (Tốt)	6 (3.0%)	**90 (45.0%)**	$\uparrow$ 84 mẫu
$\leq$ 20% (Khá)	29 (14.5%)	**123 (61.5%)**	$\uparrow$ 94 mẫu
$\leq$ 50% (Trung bình)	95 (47.5%)	**165 (82.5%)**	$\uparrow$ 70 mẫu

Bảng 6: So sánh phân phối số lượng mẫu theo các ngưỡng CER

Phân tích cải thiện từng mẫu:

- **Mức độ cải thiện tổng quát:** Có tới **190/200 mẫu (95.0%)** cho thấy kết quả nhận dạng tốt hơn sau khi tinh chỉnh. Chỉ có một tỷ lệ rất nhỏ (2.5%) số mẫu bị suy giảm chất lượng và 2.5% không thay đổi.
- **Khả năng ứng dụng:** Với mô hình tinh chỉnh, 45.0% số mẫu đạt CER dưới 10%, ngưỡng thường được coi là chấp nhận được cho các ứng dụng OCR thực tế cần ít sự chỉnh sửa hậu kỳ. Trong khi đó, mô hình gốc chỉ có 3% số mẫu đạt tiêu chuẩn này.
- **Độ bao phủ:** Hơn 80% số mẫu của mô hình tinh chỉnh có CER dưới 50%, cho thấy mô hình đã học được các đặc trưng thị giác cơ bản của chữ viết tay tiếng Việt, so với chưa đầy một nửa ở mô hình gốc.

4.4 Thống kê phân loại lỗi và tỉ lệ khớp hoàn hảo

Nhằm hiểu rõ hơn về đặc điểm sai lệch của các mô hình, ba loại lỗi cơ bản trong nhận dạng văn bản gồm Insertion (Chèn), Deletion (Xóa) và Substitution (Thay thế) được phân tích chi tiết. Đồng thời, khả năng dự đoán chính xác tuyệt đối được đánh giá thông qua tỉ lệ khớp hoàn hảo (CER=0%). Kết quả thống kê trên tập kiểm tra được trình bày trong Bảng 7 và Bảng 8.

4.4.1 Phân tích loại lỗi

Số liệu cho thấy quá trình tinh chỉnh đã làm thay đổi đáng kể cấu trúc lỗi của mô hình:

- **Giảm lỗi Substitution và Deletion:** Lỗi thay thế giảm mạnh 51.8% và lỗi xóa giảm 50.7%. Điều này chứng tỏ mô hình tinh chỉnh đã cải thiện đáng kể khả năng nhận diện hình dáng ký tự (giảm nhầm lẫn) và khả năng phát hiện đầy đủ các ký tự trong ảnh (giảm bỏ sót), đặc biệt quan trọng đối với các dấu thanh nhỏ trong tiếng Việt.
- **Gia tăng lỗi Insertion:** Ngược lại với xu hướng giảm tổng lỗi (giảm 11.6%), lỗi chèn lại tăng 34.4%, chiếm tới 70.7% tổng số lỗi của mô hình tinh chỉnh. Hiện tượng này cho thấy mô hình sau khi học đôi khi có xu hướng sinh ra các ký tự hoặc từ ngữ dư thừa ("ảo giác"), có thể do cố gắng giải mã các nhiễu trong ảnh hoặc do đặc tính sinh văn bản tự hồi quy của LLM.

Loại lỗi	Baseline	Fine-tuned	Thay đổi	% Thay đổi
Insertion	4,420	5,939	$\uparrow$ 1,519	+34.4%
Deletion	1,159	571	$\downarrow$ 588	-50.7%
Substitution	3,915	1,887	$\downarrow$ 2,028	-51.8%
Tổng lỗi	**9,494**	**8,397**	$\downarrow$ **1,097**	**-11.6%**

Bảng 7: Thống kê số lượng lỗi theo loại (đơn vị: kí tự) của hai mô hình

4.4.2 Tỉ lệ khớp hoàn hảo

Chỉ số	Baseline	Fine-tuned	Cải thiện
Số mẫu khớp hoàn hảo	0	**41**	$\uparrow$ 41
Tỉ lệ	0.00%	**20.50%**	$\uparrow$ 20.50%

Bảng 8: So sánh khả năng dự đoán chính xác tuyệt đối (CER=0%)

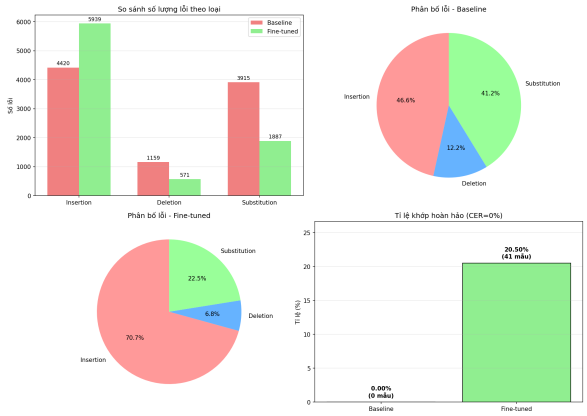
Bảng 8 làm nổi bật sự vượt trội về độ chính xác tuyệt đối của mô hình sau tinh chỉnh. Trong khi mô hình gốc không thể dự đoán chính xác hoàn toàn bất kỳ mẫu nào (0%), mô hình tinh chỉnh đã đạt được tỉ lệ khớp hoàn hảo lên tới **20.50%** (41/200 mẫu). Kết quả này khẳng định rằng với các mẫu dữ liệu rõ ràng, mô hình đã học được cách chuyển đổi hình ảnh sang văn bản một cách hoàn hảo, không sai lệch dù chỉ một ký tự.

4.5 Phân tích định tính các trường hợp điển hình

4.5.1 Các trường hợp thành công (CER = 0%)

Kết quả đánh giá cho thấy mô hình tinh chỉnh đạt độ chính xác tuyệt đối (CER = 0%) trên nhiều mẫu có độ phức tạp cao, bao gồm các câu dài và chứa đầy đủ dấu tiếng Việt. Một số ví dụ tiêu biểu được trích trực tiếp từ log đánh giá như sau:

- *Mẫu 9 (CER = 0.00%)*



Hình 2: Biểu đồ phân tích lỗi: (Trên-Trái) Số lượng lỗi theo loại; (Trên-Phải) Phân bố lỗi Baseline; (Dưới-Trái) Phân bố lỗi Fine-tuned; (Dưới-Phải) Tỉ lệ khớp hoàn hảo.

  - *Văn bản tham chiếu*: "gần và xa bờ, hợp thành phòng tuyến bảo vệ, kiểm soát và làm chủ."
  - *Văn bản dự đoán*: "gần và xa bờ, hợp thành phòng tuyến bảo vệ, kiểm soát và làm chủ."

- *Mẫu 19 (CER = 0.00%)*
  - *Văn bản tham chiếu*: "xã hội, có liên quan trực tiếp đến sự phồn vinh của đất nước, đến văn."
  - *Văn bản dự đoán*: "xã hội, có liên quan trực tiếp đến sự phồn vinh của đất nước, đến văn."

- *Mẫu 205 (CER = 0.00%)*
  - *Văn bản tham chiếu*: "Hà."
  - *Văn bản dự đoán*: "Hà."

Các kết quả này cho thấy mô hình tinh chỉnh có khả năng nhận dạng chính xác cả nội dung và dấu tiếng Việt, ngay cả với các câu dài hoặc có cấu trúc phức tạp.

4.5.2 Các trường hợp lỗi nghiêm trọng (CER cao)

Mặc dù đạt được cải thiện đáng kể về CER trung bình, mô hình tinh chỉnh vẫn gặp một số trường hợp lỗi nghiêm trọng với giá trị CER rất cao.

Lỗi ảo giác Một trong những trường hợp làm CER cao đột biến là do chuỗi văn bản được sinh ra không liên quan đến nội dung hình ảnh đầu vào:

- *Mẫu 104 (CER = 1188.66%)*
  - *Văn bản tham chiếu*: "Chúng ta hoan nghênh và biểu dương những việc làm tốt của công nhân xây dựng trong việc thông tin,"
  - *Văn bản dự đoán*: "Chúng tôi cơ hội nghiệp vụ du lịch chương trình văn phòng, chương trình văn phòng, chương trình văn ."

Lỗi nhận dạng sai từ ngắn Ngoài lỗi ảo giác, mô hình còn gặp khó khăn trong việc nhận dạng các từ rất ngắn, khi chỉ một sai lệch nhỏ cũng làm CER tăng mạnh.

- *Mẫu 134 (CER = 150.00%)* "vụ." → "anh."
- *Mẫu 224 (CER = 100.00%)* "chia." → "kinh."

Các trường hợp này thường xuất hiện khi độ dài văn bản tham chiếu rất ngắn, khiến chỉ số CER trở nên nhạy cảm với từng lỗi ký tự đơn lẻ.

5 Thảo luận

5.1 Cải thiện hiệu suất

5.1.1 Cải thiện tổng thể

So sánh định lượng giữa hai mô hình cho thấy hiệu quả rõ rệt của phương pháp tinh chỉnh dựa trên LoRA đối với miền dữ liệu chữ viết tay tiếng Việt:

- **Giảm thiểu sai số cực lớn:** Tỷ lệ lỗi ký tự (Mean CER) giảm **778.54** điểm phần trăm, tương ứng với mức cải thiện tương đối lên tới **96.0%**. Điều này xác nhận rằng mô hình gốc hoàn toàn không có khả năng xử lý đầu vào là chữ viết tay tiếng Việt, thường xuyên sinh ra các chuỗi ký tự vô nghĩa hoặc bị lỗi lặp từ, trong khi mô hình tinh chỉnh đã hội tụ về vùng biểu diễn văn bản chính xác.

- **Khả năng dự đoán chính xác tuyệt đối:** Tỷ lệ khớp hoàn hảo tăng từ **0.00%** lên **20.50%**. Kết quả này chứng minh mô hình không chỉ học được cách nhận dạng ký tự rời rạc mà còn nắm bắt được cấu trúc toàn vẹn của từ và câu trong 1/5 số lượng mẫu kiểm tra.

5.1.2 Cơ chế thích nghi

Sự cải thiện vượt bậc này có thể được giải thích thông qua các cơ chế thích nghi mà mô hình đã học được trong quá trình huấn luyện:

- **Chuẩn hóa dấu thanh và mã hóa:** Việc giảm mạnh lỗi Thay thế (Substitution giảm 51.8%) cho thấy các ma trận trọng số LoRA đã điều chỉnh lại không gian vector của Language Decoder, giúp mô hình phân biệt chính xác các ký tự tiếng Việt có hình dáng tương tự nhưng khác dấu (ví dụ: *a, ă, â* hoặc *hỏi, ngã*), khắc phục hoàn toàn lỗi encoding thường gặp ở các mô hình Latin.

- **Xử lý biến thể nét viết:** Lỗi Xóa (Deletion) giảm 50.7% cho thấy Vision Encoder (thông qua gradient truyền ngược từ Decoder) đã học cách trích xuất đặc trưng tốt hơn từ các nét viết dính hoặc mờ, vốn là đặc thù của dữ liệu UIT-HWDB.

- **Tận dụng tri thức ngôn ngữ:** Mô hình ngôn ngữ lớn (LLM) bên trong DeepSeek-OCR đóng vai trò như một bộ sửa lỗi ngữ nghĩa, sử dụng xác suất chuỗi từ để lựa chọn các từ có nghĩa trong tiếng Việt thay vì các chuỗi ký tự ngẫu nhiên.

5.2 Phân tích hiện tượng Ảo giác

Một phát hiện đáng chú ý từ thực nghiệm là sự gia tăng của lỗi Chèn (Insertion tăng 34.4% và chiếm 70.7% tổng lỗi). Nguyên nhân của hiện tượng này có thể xuất phát từ:

1. **Đặc tính sinh tự hồi quy:** Khi gặp các mẫu nhiễu hoặc khó đọc, mô hình có xu hướng "đoán" và sinh thêm từ dựa trên ngữ cảnh đã học, dẫn đến việc thêm các từ hư từ hoặc lặp lại nội dung không có trong ảnh.

2. **Nhiễu trong nhãn dữ liệu (Ground Truth):** Một số mẫu trong tập dữ liệu có thể chứa nhiễu (ví dụ: vết bẩn, dòng kẻ) mà mô hình cố gắng giải mã thành văn bản.

3. **Overfitting với văn phong hành chính:** Do một phần lớn dữ liệu huấn luyện là văn bản hành chính/báo chí, mô hình có xu hướng tự động điền các từ ngữ trang trọng hoặc các cụm từ phổ biến ngay cả khi chúng không xuất hiện đầy đủ trong ảnh.

5.3 Hạn chế của nghiên cứu

5.3.1 Hạn chế về Dữ liệu

- **Thiên lệch trong phân bố người viết:** Tập dữ liệu UIT-HWDB chủ yếu được thu thập từ đối tượng sinh viên và giảng viên. Do đó, phong cách viết tay có thể thiếu sự đa dạng về độ tuổi, trình độ văn hóa và các kiểu chữ viết tháu/biến dạng thường gặp trong các văn bản đời sống thực tế (như đơn thuốc, ghi chú nhanh).

- **Vấn đề cắt vùng ảnh:** Đối với các mẫu cấp dòng (Line-level) và cấp từ (Word-level), quá trình cắt ảnh tự động đôi khi làm mất một phần các ký tự có nét vươn dài (như g, y, h, l) hoặc dính nhiễu từ các dòng lân cận, khiến mô hình gặp khó khăn trong việc học trọn vẹn đặc trưng hình dạng ký tự.

- **Kích thước mẫu:** Số lượng mẫu huấn luyện (8,537 mẫu) là tương đối nhỏ so với tham số của một mô hình 3 tỷ tham số. Điều này giới hạn khả năng tổng quát hóa của mô hình trên các phong cách viết tay hiếm gặp.

5.3.2 Hạn chế của Phương pháp huấn luyện

Phương pháp tinh chỉnh bằng LoRA, mặc dù hiệu quả về bộ nhớ, nhưng chỉ cập nhật khoảng 1-2% tổng số tham số. Điều này có thể hạn chế khả năng thay đổi căn bản cách mô hình xử lý các đặc trưng thị giác phức tạp so với phương pháp tinh chỉnh toàn bộ. Mặt khác, do ràng buộc tài nguyên tính toán (GPU đơn), mô hình chỉ được huấn luyện trong 1 epoch với dữ liệu hạn chế. Việc huấn luyện nhiều epoch hoặc tăng cường dữ liệu có thể giúp mô hình cho hiệu suất cao hơn.

5.4 So sánh với kết quả PaddleOCR ở LAB 1

Trong bài thực hành LAB 1, hiệu năng của thư viện PaddleOCR (sử dụng mô hình `latin` PP-OCRv3 mặc định) được khảo sát trên các văn bản in. Khi đối chiếu với mô hình DeepSeek-OCR đã được tinh chỉnh trên dữ liệu viết tay trong đồ án này, kết quả cho thấy những cải thiện rõ rệt.

5.4.1 Khả năng nhận diện dấu Tiếng Việt

- **PaddleOCR (LAB 1)**: Đây là hạn chế lớn nhất. Do sử dụng trọng số đa ngôn ngữ (latin) chưa được tối ưu cho tiếng Việt, mô hình thường xuyên gặp lỗi encoding hoặc bỏ sót dấu thanh.
  - *Ví dụ trong Lab 1*: Từ "xuất bản" bị nhận diện thành chuỗi ký tự lỗi "xuĂAt bĂ˘n", "Việt" thành "Vit".
  - *Nguyên nhân*: Bộ tokenizer và từ điển của mô hình latin không bao phủ tốt các tổ hợp ký tự Unicode dựng sẵn của tiếng Việt.

- **DeepSeek-OCR (Fine-tuned)**: Khắc phục hoàn toàn lỗi hiển thị dấu. Nhờ quá trình tinh chỉnh với LoRA, mô hình DeepSeek đã học được cách ánh xạ các đặc trưng thị giác sang đúng các token tiếng Việt (ví dụ: các từ "xã hội", "phồn vinh" được nhận diện chính xác kể cả dấu mũ và dấu thanh).

5.4.2 Độ phức tạp của miền dữ liệu

- **PaddleOCR (LAB 1)**: Được thử nghiệm trên **văn bản in** (Printed text) từ file PDF. Đây là miền dữ liệu có độ rõ nét cao, font chữ chuẩn, nhưng mô hình vẫn thất bại trong việc trích xuất nội dung có nghĩa do rào cản ngôn ngữ.

- **DeepSeek-OCR (Fine-tuned)**: Được huấn luyện và kiểm thử trên **chữ viết tay** (Handwritten text - UIT-HWDB). Mặc dù chữ viết tay có độ biến thiên hình dáng lớn và khó nhận dạng hơn nhiều so với chữ in, mô hình DeepSeek-OCR tinh chỉnh vẫn đạt Median CER 12.62%. Điều này chứng minh năng lực vượt trội của kiến trúc Vision-Language Model khi được huấn luyện đúng cách.

5.4.3 Kiến trúc và Cách tiếp cận

- **PaddleOCR**: Tiếp cận theo hướng modular truyền thống (Text Detection + Text Recognition). Ưu điểm là phát hiện bố cục (layout) rất tốt, tách dòng chính xác.

- **DeepSeek-OCR**: Tiếp cận theo hướng End-to-End (sinh văn bản trực tiếp từ ảnh). Mặc dù đôi khi gặp hiện tượng "ảo giác" - điều ít gặp ở PaddleOCR, nhưng DeepSeek-OCR thể hiện khả năng hiểu ngữ cảnh tốt hơn hẳn, giúp sửa lỗi chính tả dựa trên mô hình ngôn ngữ lớn đi kèm.

Kết luận so sánh: Trong khi PaddleOCR ở LAB 1 chỉ dừng lại ở mức phát hiện vị trí văn bản tốt nhưng thất bại trong việc "đọc hiểu" tiếng Việt, thì DeepSeek-OCR sau khi tinh chỉnh đã giải quyết được bài toán cốt lõi: **nhận dạng chính xác ngữ nghĩa và ký tự tiếng Việt**, ngay cả trên miền dữ liệu chữ viết tay đầy thách thức.

6 Kết luận

6.1 Tóm tắt các phát hiện

Đồ án này trình bày kết quả nghiên cứu về việc tinh chỉnh mô hình DeepSeek-OCR cho bài toán nhận dạng chữ viết tay tiếng Việt, một miền dữ liệu giàu thách thức. Các phát hiện chính bao gồm:

- **Hiệu quả của phương pháp thích nghi:** Kỹ thuật tinh chỉnh dựa trên LoRA kết hợp với tối ưu hóa Unsloth đã thành công trong việc chuyển đổi DeepSeek-OCR từ một mô hình hoàn toàn không hiểu tiếng Việt viết tay sang một hệ thống nhận dạng khả dụng.

- **Cải thiện hiệu suất vượt bậc:** Mô hình sau tinh chỉnh đạt **32.41%** Mean CER và **12.62%** Median CER. So với mô hình gốc, kết quả này thể hiện **mức giảm sai số lên tới 96.0%**, khẳng định khả năng học tập mạnh mẽ của mô hình chỉ sau 1 epoch huấn luyện.

- **Độ chính xác tuyệt đối:** Tỉ lệ khớp hoàn hảo đạt **20.50%**, cho thấy mô hình có khả năng nhận dạng chính xác toàn vẹn cả câu trong nhiều trường hợp phức tạp.

- **Thay đổi đặc trưng lỗi:** Phân tích lỗi tiết lộ một sự chuyển dịch quan trọng: trong khi các lỗi nhận dạng ký tự cơ bản (Thay thế, Xóa) giảm mạnh, mô hình lại đối mặt với thách thức mới về **lỗi ảo giác (Insertion)**, xuất phát từ đặc tính sinh văn bản tự hồi quy của mô hình ngôn ngữ lớn.

6.2 Đóng góp

Nghiên cứu này đóng góp vào lĩnh vực OCR tiếng Việt ở các khía cạnh:

1. **Kiểm chứng thực nghiệm:** Chứng minh tính khả thi và hiệu quả của việc tinh chỉnh mô hình thị giác-ngôn ngữ (VLM) hiện đại cho tác vụ OCR tiếng Việt với tài nguyên tính toán giới hạn (GPU đơn, 16GB VRAM).

2. **Phân tích chuyên sâu:** Cung cấp một báo cáo phân tích toàn diện không chỉ dừng lại ở chỉ số CER mà còn đi sâu vào các loại lỗi (Insertion/Deletion/Substitution) và hành vi của mô hình trên các mức độ dữ liệu khác nhau.

6.3 Hướng phát triển trong tương lai

Để khắc phục các hạn chế hiện tại và nâng cao hơn nữa hiệu suất, các hướng nghiên cứu tiềm năng bao gồm:

- **Mở rộng quy mô huấn luyện**: Tăng số lượng epoch và mở rộng kích thước tập dữ liệu huấn luyện để giúp mô hình hội tụ sâu hơn, giảm thiểu các lỗi ngẫu nhiên.

- **Khắc phục lỗi ảo giác**: Áp dụng các kỹ thuật như *DPO (Direct Preference Optimization)* hoặc điều chỉnh hàm loss để phạt nặng các lỗi chèn từ (Insertion), giúp mô hình trung thành hơn với nội dung hình ảnh.
- **Tăng cường dữ liệu**: Tạo ra các biến thể tổng hợp (nhiễu, mờ, nghiêng) để cải thiện độ bền vững của mô hình trước các điều kiện đầu vào kém chất lượng.
- **Hậu xử lý thông minh**: Kết hợp với các mô hình ngôn ngữ nhỏ hơn hoặc từ điển tiếng Việt để lọc bỏ các từ vô nghĩa hoặc các chuỗi lặp lại trong kết quả đầu ra.
- **Đánh giá chéo (Cross-evaluation)**: Thử nghiệm khả năng khái quát hóa của mô hình trên các tập dữ liệu chữ in (như trong Lab 1) hoặc các tập dữ liệu viết tay khác ngoài UIT-HWDB.

References

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *ArXiv*, abs/2305.14314.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685.

Joel Lamy-Poirier. 2021. Layered gradient accumulation and modular pipeline parallelism: fast and efficient training of large language models. *ArXiv*, abs/2106.02679.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *ArXiv*, abs/1711.05101.

Hung Tuan Nguyen, Cong Thanh Nguyen, and Masaki Nakagawa. 2021. Uit-hwdb: A database for vietnamese handwriting recognition.

Unsloth Team. 2024. Unsloth: 2x faster, 60% less vram fine-tuning. Accessed: 2024-12-16.

Haoran Wei, Yaofeng Sun, and Yukun Li. 2025. Deepseek-ocr: Contexts optical compression. *ArXiv*, abs/2510.18234.

A Mã nguồn

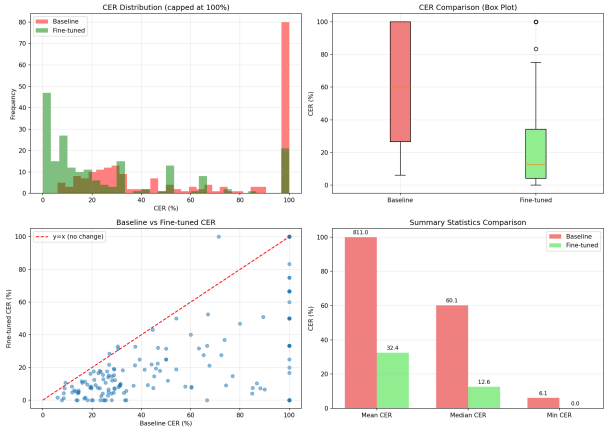
Mã nguồn hoàn chỉnh cho dự án này có sẵn tại:

GitHub Repository: gracebml/deepseek-ocr_fine-tuning

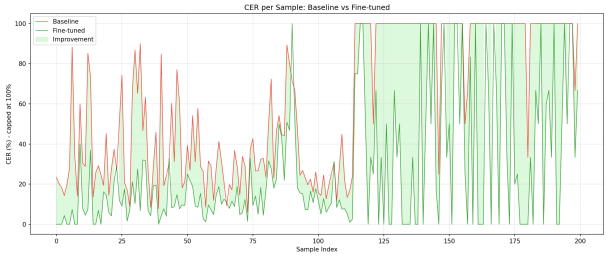
Kaggle Notebook: mlinhbng/deepseek-ocr-fine-tuning

B Các biểu đồ trực quan hóa bổ sung

Hình 3 và hình 4 trực quan hóa sự cải thiện của mô hình sau tinh chỉnh so với mô hình gốc theo phân phối CER.



Hình 3: Biểu đồ so sánh phân phối CER



Hình 4: Trực quan hóa sự cải thiện hiệu suất trên từng mẫu