ĐẠI HỌC QUỐC GIA TPHCM

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

23TNT1

Lab 2

Đề tài: Fine-tuning DeepSeek-OCR

Môn học: Nhập môn xử lý ngôn ngữ tự nhiên

Sinh viên thực hiện:

Lưu Thượng Hồng (23122006)

Giáo viên hướng dẫn:

PGS.TS. Đinh Điền

TS. Nguyễn Hồng Bửu Long

Ngày 29 tháng 12 năm 2025

Mục lục

1 Giới thiệu

1.1 Tổng quan về OCR và Thách thức với Tiếng Việt

Nhận dạng ký tự quang học (OCR - Optical Character Recognition) là công nghệ chuyển đổi hình ảnh chứa văn bản (tài liệu in, viết tay, ảnh chụp) thành định dạng văn bản máy tính có thể chỉnh sửa và tìm kiếm được.

Tuy nhiên, việc áp dụng OCR cho tiếng Việt gặp nhiều thách thức do đặc thù ngôn ngữ và chữ viết. Một số thách thức chính bao gồm:

- Đặc điểm ngôn ngữ: Tiếng Việt có nhiều dấu thanh và ký tự đặc biệt, điều này làm cho việc nhận diện trở nên khó khăn hơn so với các ngôn ngữ khác.
- Chất lượng hình ảnh: Các tài liệu quét có thể bị mờ, nghiêng hoặc có độ phân giải thấp, ảnh hưởng đến khả năng nhận diện của hệ thống OCR.
- Tính đa dạng của font chữ: Tiếng Việt sử dụng nhiều kiểu chữ khác nhau, từ chữ in đến chữ viết tay, điều này đòi hỏi hệ thống OCR phải được huấn luyện với một tập dữ liệu phong phú và đa dạng.

1.2 Tổng quan về mô hình DeepSeek-OCR

(Wei et al., 2025) DeepSeek-OCR là một mô hình Ngôn ngữ - Thị giác (Vision-Language Model - VLM) tiên tiến, được thiết kế theo hướng tiếp cận "nén ngữ cảnh quang học" (optical context compression). Thay vì chỉ nhận dạng ký tự đơn lẻ, mô hình xử lý toàn bộ hình ảnh tài liệu như một chuỗi token thị giác nén để giải mã ra văn bản.

1.2.1 Kiến trúc

Mô hình sử dụng kiến trúc End-to-End gồm hai thành phần chính nối tiếp nhau:

Encoder (DeepEncoder)

Đây là bộ mã hóa thị giác tùy chỉnh với khoảng 380M tham số, được thiết kế để xử lý ảnh độ phân giải cao nhưng vẫn tối ưu hóa bộ nhớ. DeepEncoder bao gồm 3 module con:

- Visual Perception: Sử dụng SAM-base (80M tham số) với cơ chế *window attention* để trích xuất các đặc trưng chi tiết cục bộ.
- Compressor: Một module tích chập (Conv layer) thực hiện giảm mẫu (downsample) 16 lần, giúp nén đáng kể số lượng vision tokens (ví dụ giảm từ 4096 xuống còn 256 tokens).
- Visual Knowledge: Sử dụng CLIP-large (300M tham số) với cơ chế *global attention* để nắm bắt ngữ nghĩa toàn cục và tri thức từ các token đã nén.

Decoder

Về decoder, DeepSeek-OCR sử dụng mô hình ngôn ngữ lớn DeepSeek3B-MoE (Mixture-of-Experts). Mặc dù có tổng cộng 3B tham số, mô hình chỉ kích hoạt khoảng 570M tham số trong quá trình suy luận, đảm bảo tốc độ xử lý nhanh và hiệu quả cao

1.2.2 Đầu vào

DeepSeek-OCR có khả năng xử lý linh hoạt các loại đầu vào thông qua cơ chế Đa phân giải (Multi-resolution support):

- Native Resolution: Hỗ trợ các chế độ phân giải cố định như Tiny, Small, Base và Large (tối đa 1280x1280 pixel).
- Dynamic Resolution (Gundam Mode): Hỗ trợ xử lý ảnh kích thước cực lớn hoặc tỷ lệ khung hình đặc biệt (như trang báo, tài liệu dài) bằng cách chia nhỏ ảnh (tiling) kết hợp với cái nhìn toàn cảnh, giúp không bị mất chi tiết.

1.2.3 Đầu ra

Mô hình hỗ trợ đa dạng định dạng đầu ra phục vụ nhiều mục đích khác nhau:

- Văn bản thuần & Markdown: Cho các tài liệu văn bản thông thường.
- HTML: Để nhận dạng và tái tạo cấu trúc bảng biểu (Tables) phức tạp.
- LaTeX / SMILES: Dành cho việc nhận dạng công thức toán học và công thức hóa học.
- Tọa độ (Bounding boxes): Cung cấp vị trí của đối tượng/văn bản cho các tác vụ grounding hoặc detection.

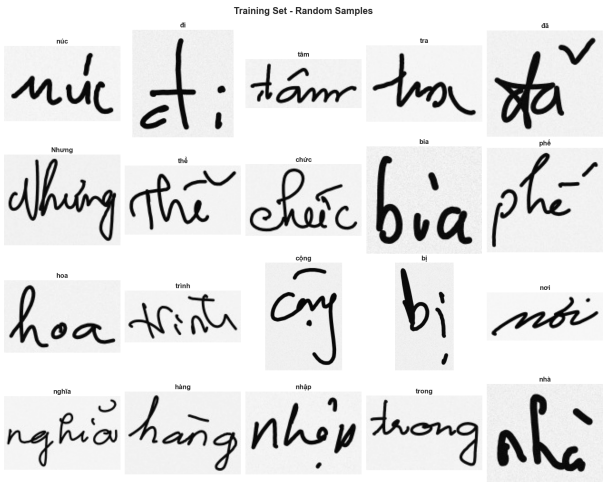
2 Phương pháp

2.1 Bộ dữ liệu: UIT-HWDB-word

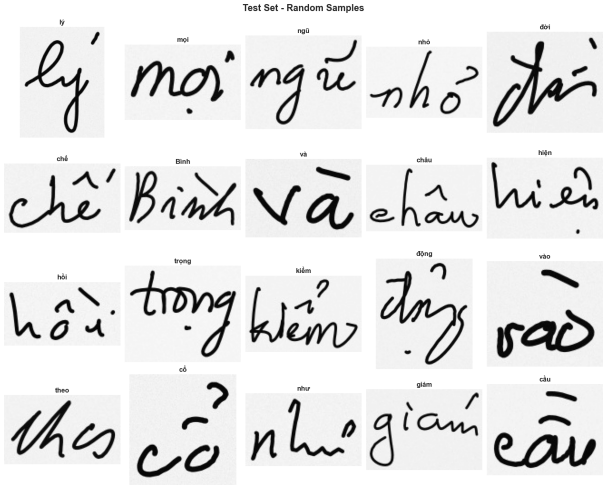
2.1.1 Tổng quan bộ dữ liệu

Ta sử dụng bộ dữ liệu UIT-HWDB-word, một phần của bộ dữ liệu chữ viết tay tiếng Việt UIT-HWDB (Nguyen et al., 2022).

Một số mẫu chữ viết tay từ bộ dữ liệu được minh họa trong Hình 3.



Hình 1: (a) Tập huấn luyện



Hình 2: (b) Tập kiểm thử

Hình 3: Một số mẫu chữ viết tay từ bộ dữ liệu UIT-HWDB-word

2.1.2 Cấu trúc dữ liệu

Bộ dữ liệu được tổ chức thành hai tập chính: tập huấn luyện (train) và tập kiểm tra (test). Cấu trúc thư mục như sau:

```
UIT_HWDB_word/
    train_data/
        1/
            image_001.png
            ...
            label.json
        ...
```

```
test_data/
    250/
        ...
        label.json
    ...
```

Mỗi thư mục con chứa các ảnh chữ viết tay và một file `label.json` chứa nhãn tương ứng cho từng ảnh. Định dạng của file nhãn như sau:

```
{
    "1.jpg": "xin chào",
    "2.jpg": "cảm ơn",
    ...
}
```

2.2 Pipeline

Quy trình fine-tuning mô hình DeepSeek-OCR trên bộ dữ liệu UIT-HWDB-word bao gồm các bước chính:

1. Chuẩn bị dữ liệu: Chuẩn bị bộ dữ liệu UIT-HWDB-word với ảnh và nhãn tương ứng.
2. Tiền xử lý dữ liệu: Tiền xử lý ảnh và định dạng dữ liệu theo chuẩn hội thoại.
3. Huấn luyện: Huấn luyện mô hình DeepSeek-OCR trên tập huấn luyện.
4. Đánh giá: Đánh giá mô hình trên tập kiểm tra sử dụng các metrics như CER, Accuracy và Exact Match.

Tuy nhiên, trước khi vào pipeline chính, ta sẽ tiến hành phân tích dữ liệu để hiểu rõ hơn về đặc điểm của bộ dữ liệu và từ đó đưa ra các quyết định tiền xử lý và cấu hình mô hình phù hợp.

2.3 Phân tích dữ liệu

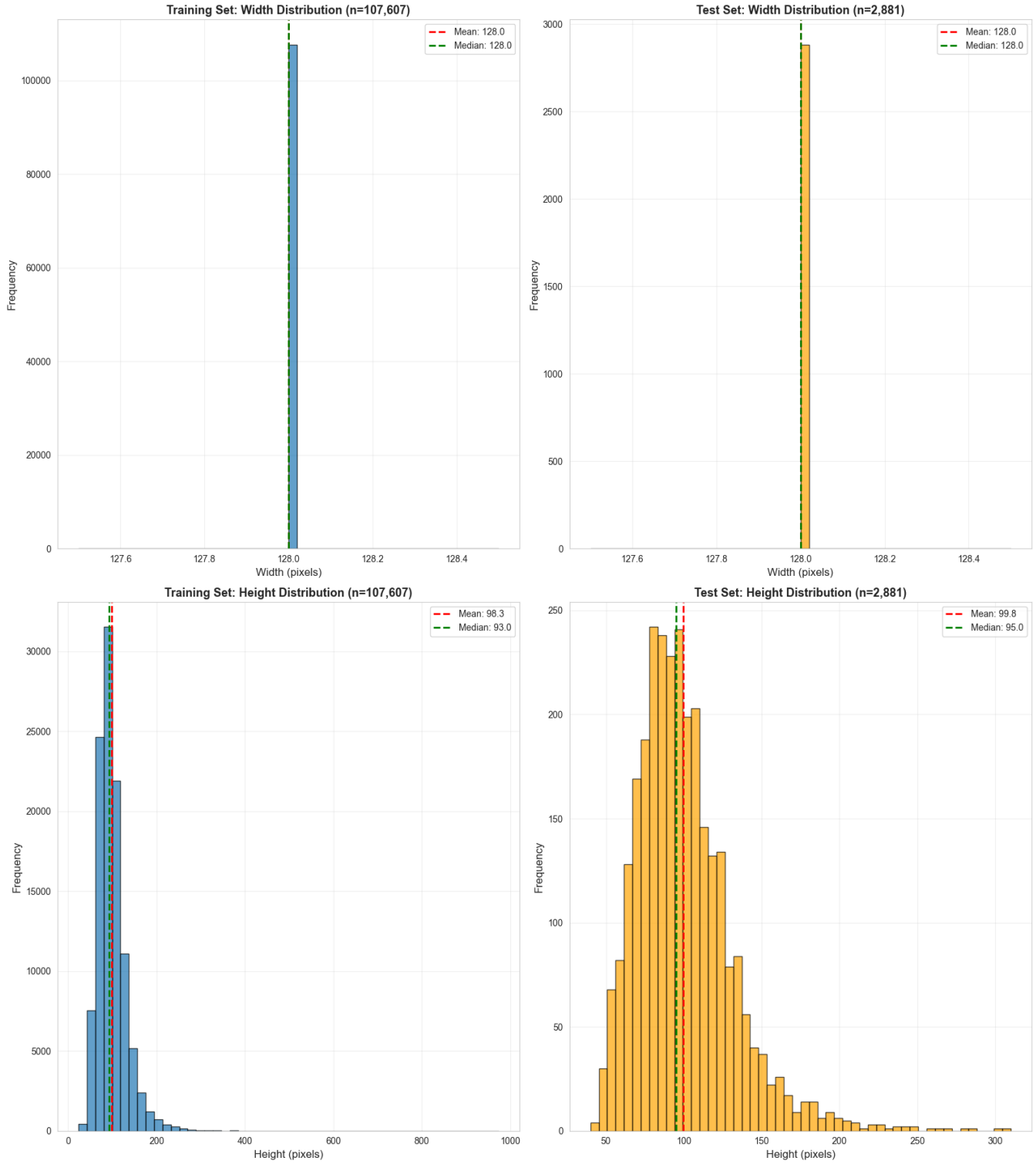
Quá trình phân tích dữ liệu được thực hiện chi tiết trên tập huấn luyện (107,607 mẫu) và tập kiểm tra (2,881 mẫu) nhằm định hướng cho các quyết định tiền xử lý và cấu hình mô hình.

2.3.1 Đặc điểm hình ảnh

Phân tích thống kê kích thước ảnh cho thấy những đặc trưng quan trọng:

- Chiều rộng cố định: Tất cả các ảnh trong bộ dữ liệu đều có chiều rộng 128 pixels.
- Chiều cao biến thiên: Chiều cao ảnh có sự dao động lớn, từ 23 pixels đến 974 pixels, với giá trị trung bình khoảng 98 pixels và trung vị là 93 pixels.
- Điểm ngoại lai (Outliers): Mặc dù phần lớn ảnh có chiều cao dưới 128 pixels, sự tồn tại của các ảnh có chiều cao lên tới gần 1000 pixels cho thấy có những mẫu chữ viết tay rất dài hoặc được viết theo chiều dọc.

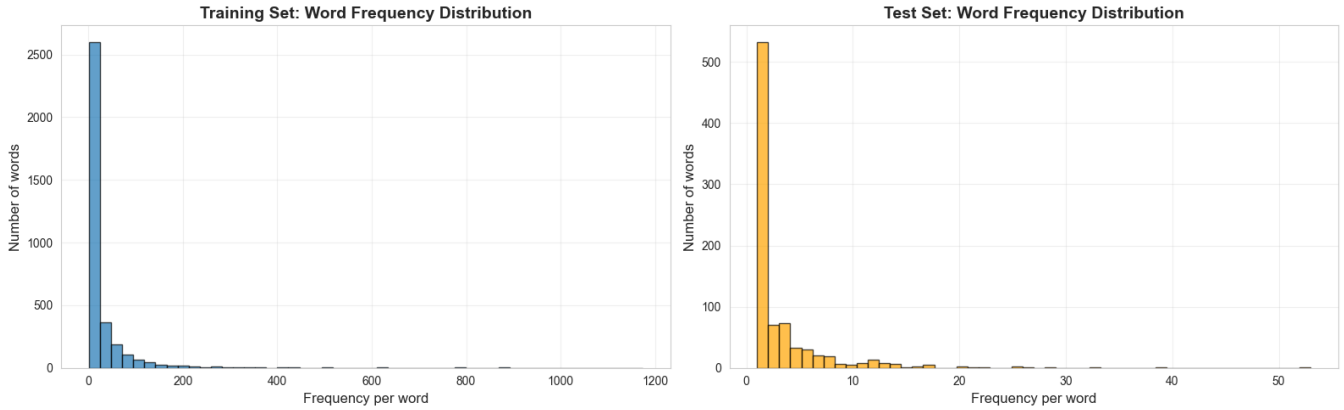
Từ kết quả này, việc lựa chọn kích thước ảnh đầu vào cho mô hình là 384x384 được đánh giá là tối ưu. Kích thước này đủ lớn để chứa trọn vẹn đa số các ảnh (với chiều cao trung bình $\sim$98px) mà không cần co giãn quá nhiều, đồng thời hạn chế lãng phí tài nguyên tính toán cho phần padding của các ảnh nhỏ.



Hình 4: Phân bố kích thước ảnh trong bộ dữ liệu UIT-HWDB-word

2.3.2 Phân bố nhãn

- Phân phối Long-tail: Tần suất xuất hiện của các nhãn từ tuân theo quy luật phân phối đuôi dài. Một lượng nhỏ các từ thông dụng chiếm tỷ trọng lớn trong tập dữ liệu, trong khi có những từ chỉ xuất hiện rất ít lần.



Hình 5: Phân bố nhãn

2.4 Tiền xử lý dữ liệu

Dựa trên notebook hướng dẫn fine-tuning mô hình DeepSeek-OCR của Unsloth (Daniel Han and team, 2023), ta sẽ thực hiện các bước tiền xử lý dữ liệu, bao gồm chuẩn bị dữ liệu, định dạng hội thoại và xử lý ảnh.

2.4.1 Chuẩn bị dữ liệu

Quá trình chuẩn bị dữ liệu bao gồm:

1. Đọc các file `label.json` từ tập dữ liệu
2. Tải ảnh và chuyển đổi sang định dạng RGB.

2.4.2 Định dạng hội thoại

Để fine-tune mô hình DeepSeek-OCR (một mô hình Vision-Language Model), dữ liệu được chuyển đổi sang định dạng hội thoại (conversation format) chuẩn:

```
{
    "messages": [
        {
            "role": "<|User|>",
            "content": "<image>\nFree OCR.",
            "images": [<PIL.Image>]
        },
        {
            "role": "<|Assistant|>",
            "content": "nhãn_của_ảnh"
        }
    ]
}
```

Trong đó, token `<image>` đại diện cho vị trí chèn các embedding của ảnh, và câu lệnh "Free OCR." đóng vai trò là prompt hướng dẫn mô hình thực hiện tác vụ nhận dạng ký tự quang học.

2.4.3 Xử lý ảnh

Trong quá trình huấn luyện, ảnh đầu vào được xử lý thông qua `DeepSeekOCRDataCollator`. Các tham số chính được thiết lập như sau:

- `crop_mode=False`: Tắt chế độ cắt ảnh động (dynamic cropping). DeepSeek-OCR mặc định hỗ trợ cắt ảnh lớn thành nhiều mảnh để xử lý chi tiết. Tuy nhiên, với bộ dữ liệu UIT-HWDB-word gồm các ảnh từ đơn lẻ kích thước nhỏ, việc này không cần thiết. Thiết lập `False` giúp mô hình tập trung vào toàn cục (global view) và giảm chi phí tính toán.
- `base_size=384`: Kích thước khung hình cơ sở. Ảnh đầu vào được thay đổi kích thước (resize) hoặc thêm viền (padding) để đạt kích thước 384×384 pixels. Giá trị 384 được chọn dựa trên phân tích phân bố kích thước ảnh, đảm bảo bao quát được phần lớn các mẫu dữ liệu (chiều cao trung bình $\sim$98px) mà không cần co nhỏ làm mất thông tin chi tiết.
- `image_size=384`: Kích thước chuẩn hóa đầu vào cho Vision Encoder. Trong chế độ không cắt ảnh, tham số này đồng bộ với `base_size` để định hình tensor đầu vào cho mạng nơ-ron.

2.5 Cấu hình huấn luyện

2.5.1 Mô hình cơ sở

Ta sử dụng mô hình DeepSeek-OCR thông qua thư viện Unsloth (`FastVisionModel`). Mô hình được huấn luyện ở độ chính xác 16-bit (BF16/FP16) với thiết lập `load_in_4bit = False`.

2.5.2 LoRA

Vì tài nguyên có hạn, ta sử dụng kỹ thuật LoRA (Low-Rank Adaptation). Ta sẽ chỉ tinh chỉnh các ma trận hạng thấp (adapters) được thêm vào các lớp attention và feed-forward, trong khi đóng băng toàn bộ trọng số của mô hình gốc. Chi tiết cấu hình được trình bày trong Bảng 1.

Bảng 1: Cấu hình LoRA

Tham số	**Giá trị**
Rank (r)	16
Alpha	16
Dropout	0
Target Modules	`q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj`

2.5.3 Hyperparameter

Quá trình fine-tuning được thực hiện với các hyperparameter được liệt kê trong Bảng 2.

Bảng 2: Hyperparameters

Tham số	**Giá trị**
Số epochs	2
Batch size	32 (per device)
Gradient Accumulation Steps	2
Learning Rate	1e-4
Optimizer	AdamW 8-bit
Scheduler	Linear decay (5 warmup steps)
Độ chính xác	BF16 hoặc FP16

2.6 Môi trường huấn luyện

Quá trình fine-tuning được tiến hành trên nền tảng Kaggle với cấu hình phần cứng bao gồm hai GPU NVIDIA T4 và 16 GB bộ nhớ RAM.

3 Thực nghiệm

3.1 Phân chia dữ liệu

Bộ dữ liệu UIT-HWDB-word đã được tác giả chia thành hai tập train/test riêng biệt. Ta sẽ không sử dụng tập validation mà thực hiện đánh giá trên tập test sau khi kết thúc quá trình huấn luyện. Trong quá trình huấn luyện, dữ liệu sẽ được xử lý bởi lớp `DeepSeekOCRDataCollator`, sau đó được chuyển cho `Trainer` để thực hiện quá trình huấn luyện.

3.2 Phương pháp thực nghiệm

Ta dùng tập test để đánh giá mô hình trước và sau khi huấn luyện:

1. Baseline (Zero-shot): Mô hình DeepSeek-OCR gốc (pre-trained) chưa qua bất kỳ quá trình huấn luyện thêm nào.
2. Fine-tuned Model: Mô hình sau khi đã được huấn luyện với kỹ thuật LoRA trên tập dữ liệu UIT-HWDB-word.

3.3 Metrics

Sau khi huấn luyện, mô hình được đánh giá trên tập kiểm thử dựa trên CER. CER (Character Error Rate) là đủ để đánh giá với dữ liệu ở cấp độ từ (word level). Nếu sử dụng WER (Word Error Rate) sẽ không phù hợp vì mỗi mẫu chỉ chứa một từ duy nhất.
Ngoài CER là metric chính, ta cũng xem xét thêm exact match rate (tỷ lệ dự đoán chính xác hoàn toàn), và accuracy (= 1 - CER) để có cái nhìn tổng quan hơn về hiệu suất mô hình.

4 Kết quả

4.1 Quá trình huấn luyện

Quá trình fine-tuning kéo dài gần 6 giờ. Lượng tài nguyên sử dụng trong quá trình huấn luyện được tóm tắt như sau:

- Thời gian: 355.23 phút

- Bộ nhớ đỉnh: 13.66 GB (92.667%)
- Bộ nhớ cho LoRA: 6.93 GB (47.012%)

Việc tinh chỉnh tham số huấn luyện đã giúp tận dụng tối đa tài nguyên trên Kaggle.
Training loss trong quá trình fine-tuning được thể hiện trong hình 6.

Hình 6: Training Loss trong quá trình fine-tuning DeepSeek-OCR

Ta có thể thấy training loss giảm dần qua các epoch (có 3 epochs). Loss ban đầu là 1.1172, sau 3 epochs giảm xuống còn 0.1101, tương ứng với mức giảm 90.15%.
Điều này cho thấy mô hình đã học tốt hơn từ dữ liệu huấn luyện trong quá trình fine-tuning. Để trực quan hơn, ta sẽ đánh giá mô hình trên tập kiểm thử và so sánh với mô hình gốc chưa được fine-tuning.

4.2 Phân tích định lượng

Kết quả sau khi đánh giá mô hình trên tập kiểm thử gồm 2,881 mẫu được tóm tắt trong bảng 3.

Bảng 3: So sánh hiệu năng giữa mô hình Baseline và Finetuned

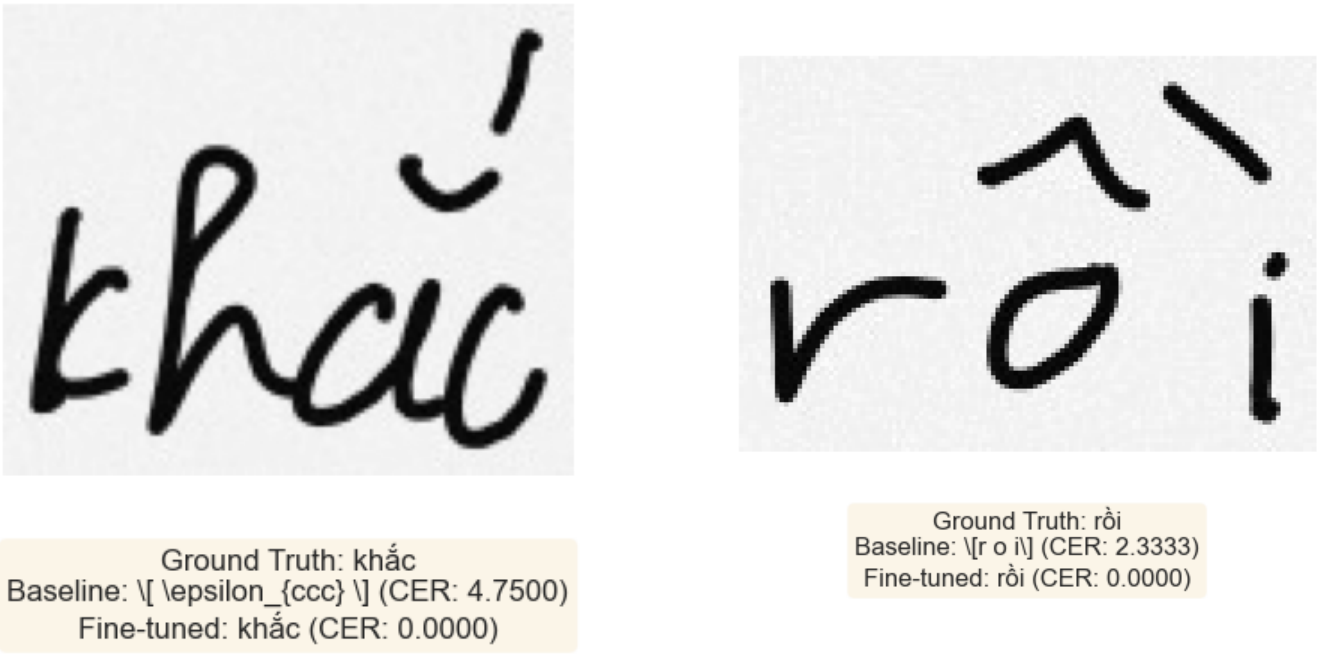
Mô hình	Overall CER	Accuracy	Exact Match
Baseline	2.2673	-1.2673	63 (2.19%)
Finetuned	**0.1075**	**0.8925**	**2296 (79.69%)**

Ta có thể thấy rằng mô hình sau khi fine-tuning có sự cải thiện đáng kể về các chỉ số đánh giá so với mô hình gốc (baseline). Overall CER giảm đáng kể từ 2.2673 xuống còn 0.1075, tương ứng với mức giảm 95.26%. Các chỉ số Accuracy và Exact Match cũng được cải thiện.

4.3 Phân tích định tính

4.3.1 Cải thiện

Để minh họa cho sự cải thiện của mô hình sau khi fine-tuning, ta sẽ xem xét các mẫu mà mô hình fine-tuned dự đoán đúng trong khi mô hình baseline dự đoán sai.



Hình 7: Ví dụ về các mẫu mà mô hình fine-tuned dự đoán đúng trong khi mô hình baseline dự đoán sai

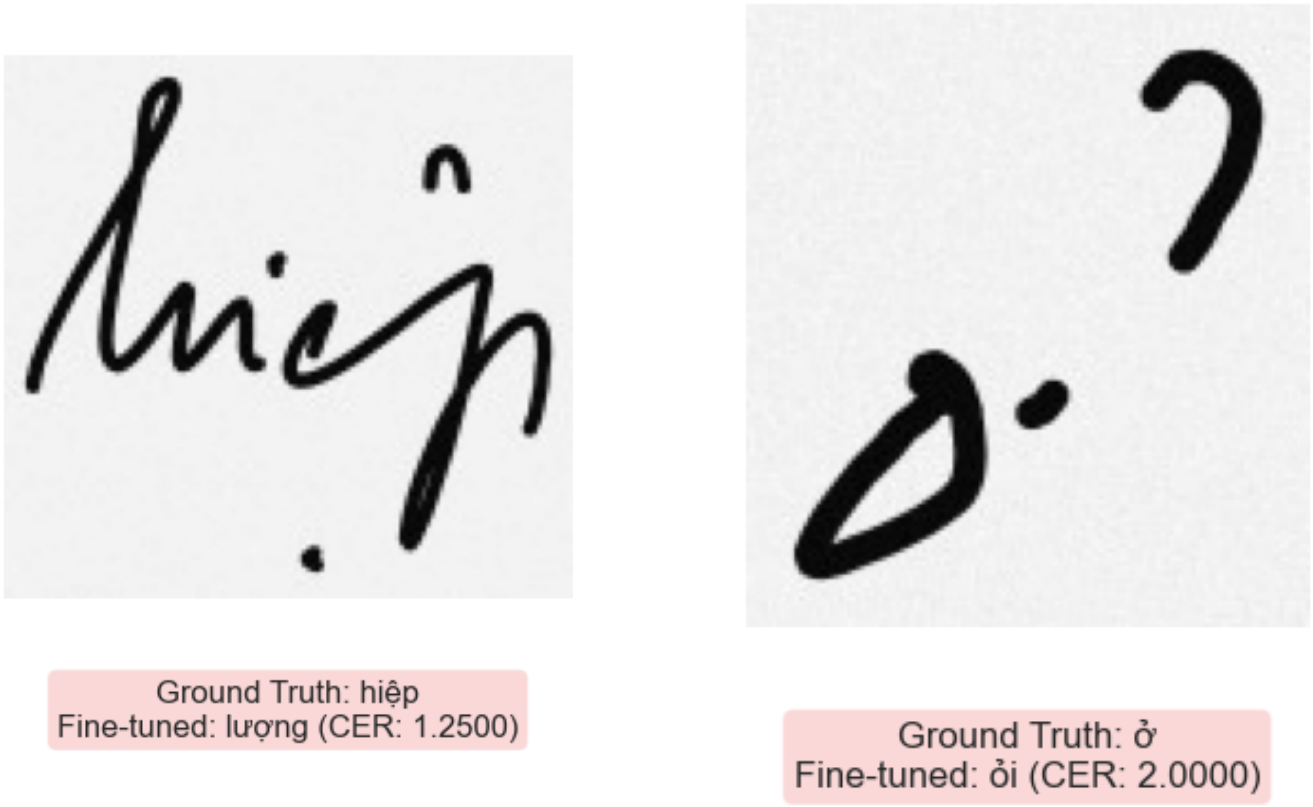
Đối với mô hình Baseline, do chưa được huấn luyện (zero-shot) trên bộ dữ liệu mới, mô hình chưa nắm bắt được các đặc trưng của chữ viết tay tiếng Việt. Điều này dẫn đến các dự đoán sai lệch lớn so với nhãn thực tế.

Cụ thể, các lỗi điển hình bao gồm việc sinh ra các chuỗi ký tự nhiễu hoặc định dạng sai. Ví dụ: từ "khắc" bị dự đoán thành `\[ \epsilon_{ccc} \]`, từ "rồi" bị dự đoán thành `\[r o i\]`.

Sau khi fine-tuning, mô hình đã học được các đặc trưng này và đưa ra các dự đoán chính xác hơn. Hai từ "khắc" và "rồi" đều được mô hình fine-tuned dự đoán đúng, khắc phục hoàn toàn các lỗi trên.

4.3.2 Trường hợp khó

Tuy nhiên, vẫn còn một số trường hợp mà model fine-tuned gặp khó khăn khi dự đoán những mẫu khó, như ví dụ trong hình 8.



Hình 8: Ví dụ về các mẫu mà mô hình fine-tuned gặp khó khăn trong việc dự đoán chính xác

5 Thảo luận

5.1 Đánh giá hiệu quả

Kết quả thực nghiệm cho thấy kỹ thuật Fine-tuning (sử dụng LoRA) đã mang lại hiệu quả vượt trội so với mô hình gốc (Zero-shot).

- Cải thiện độ chính xác: Chỉ số CER giảm mạnh từ 2.2673 xuống 0.1075, tương ứng với mức giảm lỗi hơn 95%. Điều này chứng tỏ mô hình đã học thành công các đặc trưng hình thái phức tạp của chữ viết tay tiếng Việt, đặc biệt là các dấu thanh và kiểu nét nối (ligatures) vốn là điểm yếu của mô hình gốc.

- Hiệu quả của LoRA: Việc chỉ huấn luyện một lượng nhỏ tham số (adapters) nhưng đạt được độ chính xác cao (Accuracy ~89.25%) cho thấy DeepSeek-OCR có nền tảng tri thức thị giác rất tốt, chỉ cần tinh chỉnh nhẹ để thích nghi với miền dữ liệu mới.

5.2 Hạn chế

Mặc dù mô hình fine-tuned đã đạt được kết quả ấn tượng, vẫn còn một số hạn chế đã được trình bày trong phần 4.3.2. Hạn chế này là do sự nhập nhằng của chữ viết tay, một số mẫu chữ viết quá biến dạng khiến mô hình dự đoán sai. Đây là thách thức cố hữu của bài toán HTR (Handwritten Text Recognition) mà ngay cả mắt người cũng khó phân biệt nếu thiếu ngữ cảnh.

6 Kết luận

Đồ án đã trình bày quy trình ứng dụng và tinh chỉnh mô hình DeepSeek-OCR cho bài toán nhận dạng chữ viết tay tiếng Việt. Thông qua việc huấn luyện trên tập dữ liệu UIT-HWDB-word, mô hình đạt được mức độ lỗi ký tự (CER) thấp là 10.75% và tỷ lệ khớp tuyệt đối (Exact Match) đạt gần 80%.
Kết quả này khẳng định tiềm năng lớn của các mô hình Vision-Language hiện đại như DeepSeek-OCR trong việc giải quyết các bài toán OCR cho tiếng Việt.

7 Mã nguồn

GitHub repository: ThuongHong/deepseek-ocr-vi.
Kaggle notebooks:

- Fine-tuning DeepSeek-OCR: deepseek-ocr-vi
- Đánh giá mô hình: eval-deepseek-ocr. Version được ghim (version 7) là version đánh giá mô hình fine-tuned. Version 8 là version đánh giá mô hình baseline.

Tài liệu

Daniel Han, M. H., & team, U. (2023). *Unsloth*. http://github.com/unslothai/unsloth

Nguyen, N. H., Vo, D. T. D., & Nguyen, K. V. (2022). UIT-HWDB: using transferring method to construct A novel benchmark for evaluating unconstrained handwriting image recognition in vietnamese. *RIVF International Conference on Computing and Communication Technologies, RIVF 2022, Ho Chi Minh City, Vietnam, December 20-22, 2022*, 659–664. https://doi.org/10.1109/RIVF55975.2022.10013898

Wei, H., Sun, Y., & Li, Y. (2025). Deepseek-ocr: Contexts optical compression. https://arxiv.org/abs/2510.18234