ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Báo cáo Deep Learning Research

DeepSeek-OCR

Môn: 23KHMT1 - Nhập môn học máy

Thành viên:

Nhóm 5

Nguyễn Trần Quốc Duy - 23127181

Đặng Đăng Khoa - 23127207

Võ Ngọc Bích Trâm - 23127271

Phan Quốc Thịnh - 23127486

Võ Hoàng Thương - 23127493

Giảng viên:

Lê Nhựt Nam

Võ Nhật Tân

Ngày 19 tháng 12 năm 2025

MỤC LỤC

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

DANH MỤC CÁC BẢNG

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Nghĩa đầy đủ (Tiếng Anh/Tiếng Việt)
CER	Character Error Rate (Tỷ lệ lỗi ký tự)
EDA	Exploratory Data Analysis (Phân tích dữ liệu khám phá)
FFN	Feed-Forward Network (Mạng truyền thẳng)
GPU	Graphics Processing Unit (Đơn vị xử lý đồ họa)
LLMs	Large Language Models (Các mô hình ngôn ngữ lớn)
LoRA	Low-Rank Adaptation (Thích ứng hạng thấp)
MLP	Multilayer Perceptron (Mạng perceptron đa lớp)
MoE	Mixture-of-Experts (Mô hình hỗn hợp chuyên gia)
OCR	Optical Character Recognition (Nhận dạng ký tự quang học)
PEFT	Parameter-Efficient Fine-Tuning (Tinh chỉnh tham số hiệu quả)
SFT	Supervised Fine-Tuning (Tinh chỉnh có giám sát)
TRL	Transformer Reinforcement Learning
VLMs	Vision-Language Models (Các mô hình ngôn ngữ - thị giác)
VQA	Visual Question Answering (Hỏi đáp trực quan)
WER	Word Error Rate (Tỷ lệ lỗi từ)

Tóm tắt nội dung

Trong quá trình nghiên cứu bài toán nhận dạng chữ viết tay tiếng Việt, yêu cầu đặt ra không chỉ là độ chính xác mà còn là khả năng khai thác hiệu quả đặc trưng thị giác trong điều kiện dữ liệu ngắn và phong cách viết đa dạng. Trước những hạn chế của các phương pháp OCR truyền thống khi xử lý các biến thể chữ viết tay phức tạp, mô hình DeepSeek-OCR được lựa chọn để phân tích do đề xuất cách tiếp cận kết hợp chặt chẽ giữa biểu diễn thị giác và mô hình ngôn ngữ, cho phép học được mối liên hệ hiệu quả giữa ảnh đầu vào và chuỗi ký tự đầu ra. Báo cáo tập trung phân tích kiến trúc DeepSeek-OCR, các thành phần chính của mô hình và tiến hành thực nghiệm trên tập dữ liệu chữ viết tay tiếng Việt ở mức từ/cụm từ, qua đó đánh giá tính phù hợp của mô hình đối với bài toán nghiên cứu.

CHƯƠNG 1

GIỚI THIỆU

1.1. Bối cảnh nghiên cứu

Nhận dạng ký tự quang học (Optical Character Recognition – OCR) là một bài toán quan trọng trong lĩnh vực thị giác máy tính, với nhiều ứng dụng như số hóa tài liệu và hỗ trợ nhập liệu. Đối với tiếng Việt, OCR chữ viết tay gặp nhiều thách thức do đặc thù ngôn ngữ có dấu và sự đa dạng trong phong cách viết.

Mặc dù các mô hình học sâu đã đạt được nhiều tiến bộ trong OCR, việc xử lý chữ viết tay vẫn còn khó khăn khi hình dạng và nét chữ biến thiên mạnh. Bên cạnh đó, các mô hình Ngôn ngữ lớn (Large Language Models – LLMs) cho thấy khả năng khai thác ngữ cảnh hiệu quả, nhưng chi phí tính toán tăng nhanh khi áp dụng trực tiếp cho các bài toán OCR.

1.2. Động cơ và phạm vi nghiên cứu

Bài báo **DeepSeek-OCR** được lựa chọn để phân tích do đề xuất cách tiếp cận nén thông tin thông qua biểu diễn thị giác, giúp giảm số lượng token cần xử lý trong các mô hình ngôn ngữ. Kiến trúc DeepSeek-OCR gồm hai thành phần chính là **DeepEncoder** và bộ giải mã **DeepSeek3B-MoE-A570M**, cho phép liên kết hiệu quả giữa đặc trưng thị giác và chuỗi ký tự đầu ra.

Mặc dù DeepSeek-OCR được đề xuất cho bài toán nén ngữ cảnh dài, báo cáo này tập trung khai thác kiến trúc mô hình cho bài toán nhận dạng chữ viết tay tiếng Việt ở mức từ và cụm từ. Việc giới hạn phạm vi ở các đơn vị ngắn giúp tập trung đánh giá khả năng nhận dạng ký tự tiếng Việt, thay vì xử lý văn bản dài như trong bài báo gốc.

1.3. Mục tiêu

Mục tiêu của báo cáo là phân tích kiến trúc DeepSeek-OCR và đánh giá khả năng áp dụng mô hình cho bài toán OCR chữ viết tay tiếng Việt thông qua các thực nghiệm trên tập dữ

liệu phù hợp.

1.4. Cấu trúc báo cáo

Ngoài chương Giới thiệu, báo cáo được tổ chức như sau:

Chương 2. Cơ sở lý thuyết

Chương 3. Các công trình liên quan

Chương 4. Phương pháp đề xuất

Chương 5. Mô hình đề xuất

Chương 6. Thực nghiệm và kết quả của tác giả

Chương 7. Thực nghiệm của nhóm

Chương 8. Kết luận

CHƯƠNG 2

CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về bài toán OCR và DeepSeek-OCR

Nhận dạng ký tự quang học- OCR là một lĩnh vực then chốt trong thị giác máy tính, đóng vai trò cầu nối giữa dữ liệu hình ảnh và văn bản số. Đối với tiếng Việt, thách thức không chỉ nằm ở cấu trúc dấu thanh phức tạp mà còn ở sự biến thiên mạnh mẽ của phong cách viết tay [1].

Mô hình DeepSeek-OCR được xây dựng trên triết lý nén ngữ cảnh quang học [1]. Thay vì chỉ sử dụng các mô hình Ngôn ngữ - Thị giác (VLMs) cho các tác vụ hỏi đáp cơ bản, DeepSeek-OCR tận dụng các bộ mã hóa thị giác để tối ưu hóa việc truyền tải thông tin văn bản dưới dạng các token thị giác, giúp giảm đáng kể chi phí tính toán khi suy luận mà vẫn bảo toàn độ chính xác cao [1].

2.2. Kiến trúc mô hình nền tảng

Hệ thống dựa trên sự kết hợp giữa các thành phần mã hóa tiên tiến và bộ giải mã ngôn ngữ mạnh mẽ:

2.2.1. DeepEncoder

DeepEncoder là một kiến trúc mới lạ được thiết kế để xử lý hình ảnh độ phân giải cao với chi phí bộ nhớ thấp [1]. Kiến trúc này bao gồm:

- **Cơ chế Attention:** Kết hợp giữa window attention và global attention để khai thác các đặc trưng từ cấp độ nét chữ đến bố cục toàn trang [2].
- **Bộ nén 16x Convolutional Compressor:** Giúp làm giảm số lượng vision tokens trước khi đưa vào các lớp xử lý sâu hơn, đảm bảo hiệu suất nén vượt trội so với các phương pháp truyền thống [1].

2.2.2. DeepSeek-V2/V3 và cơ chế MoE

Bộ giải mã sử dụng kiến trúc Mixture-of-Experts (MoE) từ dòng mô hình DeepSeek-V2 [3] và DeepSeek-V3 [4]. Cơ chế này cho phép mô hình kích hoạt các "chuyên gia" phù hợp cho từng loại dữ liệu đầu vào, giúp tối ưu hóa hiệu năng xử lý ngôn ngữ tự nhiên và giải mã văn bản từ các đặc trưng thị giác [3].

2.3. Kỹ thuật tinh chỉnh tham số hiệu quả (PEFT)

Để thích nghi mô hình với tập dữ liệu chữ viết tay tiếng Việt đặc thù, kỹ thuật LoRA được sử dụng để tinh chỉnh mô hình một cách hiệu quả mà không cần huấn luyện lại toàn bộ tham số.

2.3.1. Nguyên lý LoRA

LoRA giả định rằng quá trình cập nhật trọng số trong các mô hình lớn có thể được mô phỏng qua các ma trận hạng thấp [5]. Công thức cập nhật trọng số được xác định bởi:

$$W' = W + \Delta W = W + BA \tag{2.1}$$

Trong đó W là ma trận trọng số gốc được giữ cố định, còn A và B là các ma trận có hạng r nhỏ, giúp giảm thiểu số lượng tham số cần huấn luyện và tiết kiệm tài nguyên VRAM [6].

2.3.2. Các thành phần mục tiêu trong Transformer

Quá trình tinh chỉnh tập trung vào các thành phần cốt lõi của khối Transformer:

- **Module Attention:** Bao gồm Query, Key, và Value để định hướng mô hình tập trung vào các vùng hình ảnh chứa nét chữ [1].
- **Mạng Feed-Forward (FFN/MLP):** Sử dụng các lớp như *up_proj* và *down_proj* để ánh xạ các đặc trưng thị giác sang không gian ký tự tiếng Việt [3].

2.4. Các chỉ số đánh giá độ chính xác

Để đo lường hiệu quả của mô hình trên tập dữ liệu thực nghiệm, các chỉ số sai số chuẩn hóa được áp dụng:

- **WER:** Tỷ lệ lỗi ở cấp độ từ, phản ánh khả năng nhận diện đúng các cụm từ tiếng Việt [1].
- **CER:** Tỷ lệ lỗi ở cấp độ ký tự, giúp đánh giá chi tiết khả năng nhận diện các dấu thanh

và ký tự đặc biệt [1].

CHƯƠNG 3

CÁC CÔNG TRÌNH LIÊN QUAN

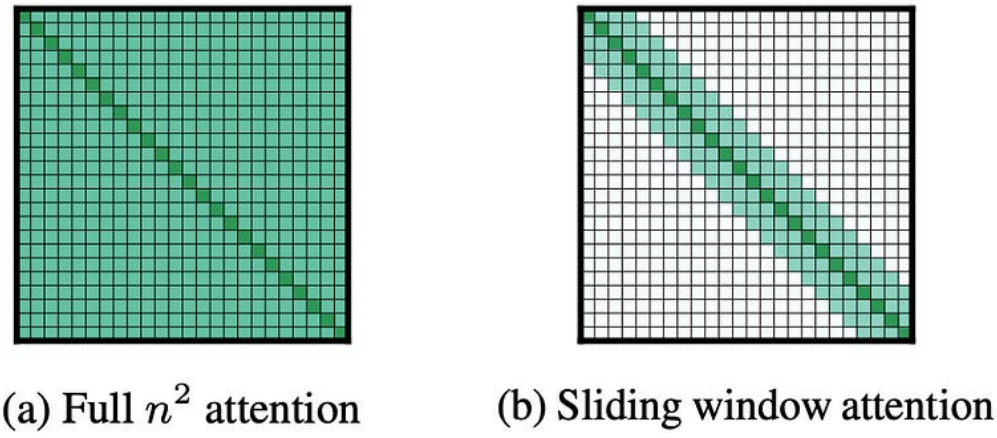
3.1. Typical Vision Encoders (Bộ mã hóa thị giác điển hình) trong VLMs

3.1.1. Vary

Kiến trúc hai luồng của Vary (Dual-Tower Encoder)

Vary sử dụng kiến trúc **dual-tower** [1], bao gồm hai nhánh xử lý song song nhằm mở rộng visual vocabulary và duy trì khả năng xử lý ảnh độ phân giải lớn. Cụ thể:

- **Luồng Local (SAM / ViTDet)**: sử dụng window attention, trong đó ảnh được chia thành nhiều cửa sổ (windows) và attention chỉ được tính trong từng cửa sổ. Cách này làm giảm activation memory từ $O(N^2)$ xuống $O((N/k)^2)$ và giúp mô hình học tốt các đặc trưng hình dạng, cạnh và layout chi tiết.

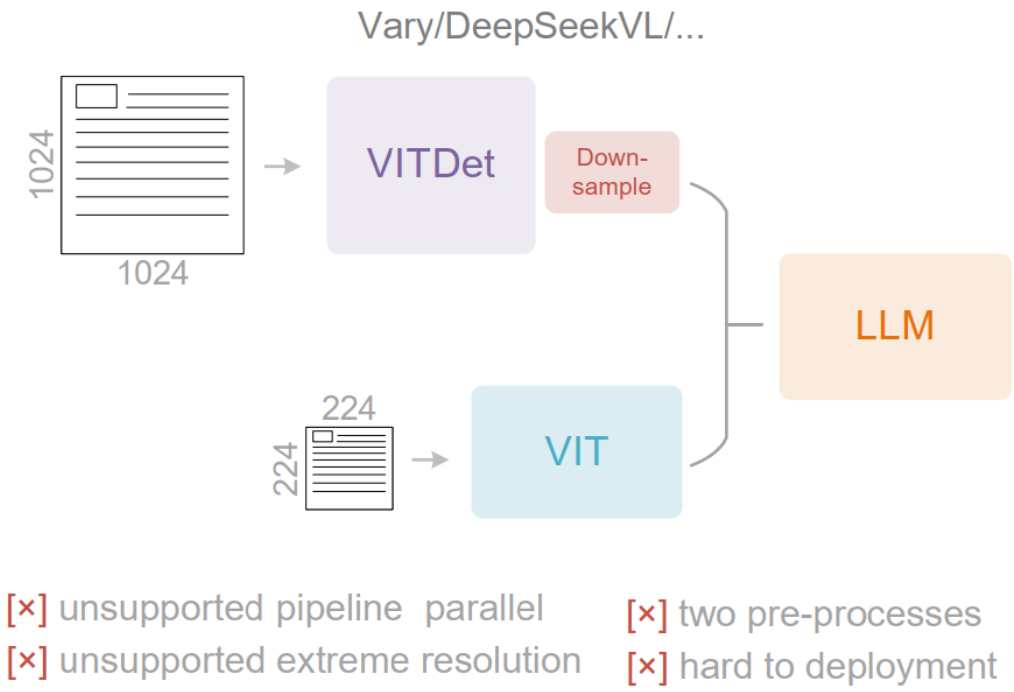


Hình 3.1: So sánh cơ chế attention. (Nguồn: [1])

- **Luồng Global (ViT / CLIP)**: ảnh được resize nhỏ hơn và đưa vào backbone ViT/CLIP để thu nhận ngữ cảnh tổng quát và thông tin semantic cấp cao.

Hai luồng được chạy song song và ghép lại, giúp mô hình đồng thời nắm bắt thông tin cục bộ lẫn tổng thể. Tuy vậy, kiến trúc dual-tower của Vary gặp nhiều hạn chế [1]:

- cần hai quy trình tiền xử lý ảnh độc lập;
- tiêu tốn nhiều GPU memory vì phải chạy hai mô hình cùng lúc;
- khó triển khai trong thực tế do pipeline song song không tương thích tốt với pipeline parallelism.



Hình 3.2: Kiến trúc hai luồng (dual-tower) của Vary. Ảnh độ phân giải lớn được xử lý bằng ViTDet (local tower), trong khi phiên bản ảnh thu nhỏ được xử lý bằng ViT/CLIP (global tower). Hai luồng được kết hợp trước khi đưa vào LLM, nhưng mô hình gặp hạn chế như hai bước tiền xử lý, khó triển khai và không hỗ trợ pipeline parallelism. (Nguồn: [1])

Cách DeepSeek-OCR kế thừa và cải tiến so với Vary

DeepSeek-OCR tiếp tục khai thác ý tưởng kết hợp **local + global**, nhưng thay vì sử dụng hai tower song song như Vary, mô hình áp dụng kiến trúc **nối tiếp (serial)** nhằm giảm chi phí bộ nhớ và đơn giản hóa triển khai [1].

- **Bước 1: SAM-base (Local Perception)** SAM-base xử lý ảnh độ phân giải lớn và trích

xuất các đặc trưng local bằng window attention, tương tự như nhánh local của Vary. Với patch-size 16, ảnh 1024 × 1024 được chuyển thành 4096 patch tokens [1].

- **Bước 2: Bộ nén 16× (Convolutional Compression)** Thay vì đưa trực tiếp hàng nghìn token vào CLIP, DeepSeek-OCR sử dụng module nén gồm hai lớp convolution stride 2, giảm số token từ 4096 xuống còn 256 (tương đương nén 16×). Việc này làm giảm mạnh activation memory trước khi đưa vào global attention [1].
- **Bước 3: CLIP-Large (Global Semantics)** CLIP-Large nhận các token đã nén và trích xuất thông tin ngữ nghĩa tổng quát. Điều này tương tự vai trò của global tower trong Vary nhưng với chi phí bộ nhớ thấp hơn đáng kể [1].

Thiết kế nối tiếp này giúp DeepSeek-OCR **vẫn giữ được cả hai loại thông tin local–global như Vary**, nhưng:

- giảm mạnh số lượng vision tokens trước khi vào global attention;
- đơn giản hóa pipeline vì chỉ cần một luồng xử lý duy nhất;
- dễ triển khai hơn và tiêu tốn ít GPU memory hơn.

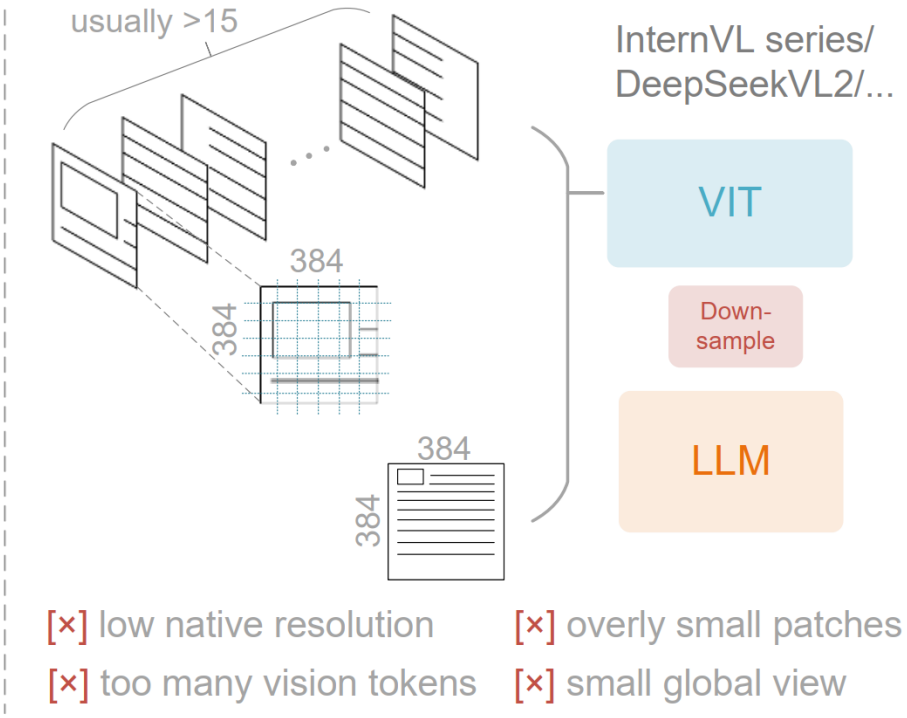
Nhờ cách sắp xếp lại pipeline, DeepSeek-OCR duy trì được ưu điểm của Vary nhưng khắc phục phần lớn hạn chế của kiến trúc dual-tower, đặc biệt là vấn đề memory overhead và độ phức tạp triển khai [1].

3.1.2. InternVL2.0

Áp dụng ý tưởng tile-based (InternVL2.0) trong DeepSeek-OCR

Một hướng tiếp cận phổ biến để xử lý ảnh độ phân giải cao là chia ảnh thành nhiều ô nhỏ (tiles) để mô hình có thể xử lý song song, từ đó giảm activation memory. Phương pháp tile-based này, được tiêu biểu bởi InternVL2.0 [1], giúp giảm chi phí trên các ảnh lớn nhờ chia tách không gian và phân rã attention theo từng vùng cục bộ. Cách làm này giúp tăng khả năng xử lý các ảnh có độ phân giải rất cao mà không gây tràn bộ nhớ GPU. Tuy nhiên, việc chia ảnh thành quá nhiều tiles kéo theo số lượng patch tăng mạnh, dẫn đến số lượng vision tokens rất lớn. Khi đó, LLM decoder có nguy cơ quá tải do phải xử lý chuỗi tokens dài, đồng thời làm chậm tốc độ suy luận [1].

Các mô hình tile-based (như InternVL2.0) thường có nhược điểm cố hữu: độ phân giải gốc thấp (thường dưới 512×512), ảnh lớn bị chia quá nhỏ, và tổng số vision tokens tăng vượt



Hình 3.3: Minh họa cơ chế tile-based của InternVL2.0 trong tài liệu gốc. Mô hình chia ảnh thành nhiều tile nhỏ (thường >15 tiles), sau đó xử lý bằng encoder ViT, dẫn đến số lượng vision tokens rất lớn và thiếu global view. (Nguồn: [1])

kiểm soát [1]. Điều này khiến mô hình dễ rơi vào tình trạng dư thừa tokens trong khi lại mất cấu trúc bố cục toàn trang.

DeepSeek-OCR vẫn kế thừa ý tưởng chia patch giống InternVL2.0, tuy nhiên thực hiện theo một cách tối ưu và có kiểm soát hơn [1]. Thay vì luôn chia tile cho mọi ảnh, mô hình chỉ kích hoạt tile mode trong các trường hợp cần thiết, đặc biệt là khi gặp ảnh cực lớn. Cơ chế này được tác giả gọi là **Gundam mode** (Figure 4) [1], trong đó ảnh được xử lý đồng thời qua:

- **Global View**: một ảnh 1024×1024 duy nhất nhằm bảo toàn bố cục toàn trang, tránh mất ngữ cảnh như ở các mô hình tile-local;
- **Local Tile Views**: chia ảnh thành n tiles 640×640 (với $n \in [2, 9]$) giúp mô hình thu nhận chi tiết cục bộ mà vẫn giữ số lượng tokens trong phạm vi cho phép.

Khác với InternVL2.0 – nơi mọi ảnh đều bị chia thành tiles và encoder phải xử lý một số lượng tokens rất lớn – DeepSeek-OCR kết hợp thêm cơ chế phân giải động (dynamic resolution) và

bộ nén 16×, giúp số lượng vision tokens ở Gundam mode chỉ khoảng $n \times 100 + 256$ tokens thay vì hàng nghìn tokens như tile-based thuần túy [1]. Điều này giúp mô hình:

- duy trì được **global layout** của tài liệu (nhờ global view);
- giữ số lượng tokens luôn trong phạm vi nhỏ và ổn định;
- tránh việc LLM decoder bị quá tải khi xử lý các tài liệu có hàng chục nghìn ký tự;
- xử lý được ảnh độ phân giải rất cao mà vẫn kiểm soát activation memory hiệu quả.

Nhờ cách áp dụng có chọn lọc ý tưởng tile-based từ InternVL2.0, kết hợp global view và nén token, DeepSeek-OCR đạt được khả năng xử lý tài liệu lớn với hiệu quả bộ nhớ cao mà không phải đánh đổi chi phí tính toán hay độ ổn định trong giải mã [1].

3.1.3. Qwen2-VL

Encoder kiểu NaViT (Qwen2-VL) và triết lý kế thừa trong DeepSeek-OCR

Một nhánh encoder phổ biến khác trong các VLM hiện đại là nhóm mô hình kiểu NaViT, tiêu biểu như Qwen2-VL [1]. Thay vì chia ảnh thành nhiều tile như InternVL2.0, các mô hình này **giữ nguyên toàn bộ ảnh** rồi chia trực tiếp thành các patch cố định, sau đó áp dụng self-attention lên *tất cả* patch tokens. Số lượng vision tokens vì thế tỉ lệ trực tiếp với độ phân giải của ảnh:

$$\text{tokens} = \left\lfloor \frac{w}{p} \right\rfloor \times \left\lfloor \frac{h}{p} \right\rfloor,$$

trong đó w, h là chiều rộng và chiều cao ảnh, p là kích thước patch (thường là 14 hoặc 16). Khi w, h tăng lên, số lượng tokens tăng rất nhanh, dẫn tới:

- activation memory của encoder tăng mạnh, dễ gây tràn bộ nhớ GPU với ảnh độ phân giải cao;
- chuỗi tokens đưa vào LLM decoder trở nên rất dài, làm chậm cả pha prefill lẫn generation.

Cách DeepSeek-OCR kế thừa triết lý của Qwen2-VL. DeepSeek-OCR không sao chép trực tiếp kiến trúc NaViT, mà chỉ **kế thừa triết lý về tính linh hoạt với độ phân giải**. Thay vì cố định một input size duy nhất hoặc luôn chia nhỏ ảnh thành tile, DeepSeek-OCR cho phép đưa ảnh vào ở nhiều độ phân giải khác nhau (Tiny, Small, Base, Large, Gundam), sao cho [1]:

- bố cục toàn trang (global layout) được giữ lại, không bị cắt vụn như các phương pháp

tile-based;

- số lượng vision tokens vẫn được khống chế ở mức nhỏ và ổn định nhờ kết hợp với bộ nén 16× phía sau DeepEncoder, nên chuỗi tokens gửi cho LLM không bị phình to theo độ phân giải ảnh;
- mô hình vẫn xử lý được nhiều kích thước ảnh khác nhau trong thực tế, tương tự tinh thần "any resolution" của Qwen2-VL, nhưng với chi phí bộ nhớ và độ dài context dễ kiểm soát hơn.

Như vậy, DeepSeek-OCR chỉ vay mượn *triết lý linh hoạt theo độ phân giải* từ Qwen2-VL/NaViT, nhưng thiết kế lại encoder và chuỗi nén token để tránh nhược điểm activation memory lớn và chuỗi vision tokens quá dài khi xử lý các tài liệu độ phân giải cao [1].

3.2. End-to-End OCR

3.2.1. Các mô hình OCR end-to-end trước DeepSeek-OCR

Sự phát triển của các mô hình thị giác–ngôn ngữ (VLM) trong những năm gần đây đã thúc đẩy sự xuất hiện của hàng loạt mô hình OCR end-to-end, thay thế cho pipeline truyền thống gồm hai bước tách biệt là phát hiện (detection) và nhận dạng (recognition). Thay vì cần nhiều mô hình chuyên biệt, các hệ thống này trực tiếp ánh xạ ảnh tài liệu sang chuỗi ký tự đầu ra [7, 1].

Nougat. Nougat là một trong những mô hình đầu tiên áp dụng kiến trúc end-to-end cho bài toán OCR học thuật, tập trung vào tài liệu dạng bài báo arXiv [7]. Mô hình cho thấy rằng một VLM đủ mạnh có thể xử lý trực tiếp bố cục dày đặc (dense layouts) của giấy tờ khoa học mà không cần tách riêng bước phát hiện vùng văn bản.

GOT-OCR2.0. GOT-OCR2.0 mở rộng phạm vi OCR sang nhiều dạng ảnh tổng hợp và các nhiệm vụ OCR 2.0 (chẳng hạn như biểu đồ, công thức hoá học, hình học phẳng), đồng thời được thiết kế theo hướng cân bằng giữa hiệu năng và tài nguyên tính toán [7]. Trên OmniDocBench, mô hình sử dụng khoảng 256 vision tokens cho mỗi trang tài liệu [1].

InternVL và Qwen-VL. Bên cạnh các mô hình chuyên OCR, các VLM tổng quát như InternVL và Qwen-VL không được thiết kế riêng cho OCR, nhưng ngày càng cải thiện năng lực phân tích tài liệu (dense visual perception) [1]. Khi được huấn luyện trên các tập

PDF/scan lớn, chúng có thể thực hiện các tác vụ như trích xuất văn bản, phân tích bảng, hoặc trả lời câu hỏi trên tài liệu.

Hạn chế chung: quá nhiều vision tokens. Mặc dù đã đơn giản hoá pipeline OCR và đạt chất lượng tốt, phần lớn các mô hình nói trên không coi việc *tối ưu số lượng vision tokens* là mục tiêu chính. Chúng thường sinh ra một lượng vision tokens rất lớn trên mỗi trang tài liệu:

- Nougat sử dụng encoder ViT chuẩn với số patch/token tỷ lệ trực tiếp theo diện tích ảnh đầu vào [7];
- GOT-OCR2.0 dùng khoảng 256 vision tokens/trang [7, 1];
- các mô hình lớn như InternVL2-76B hay Qwen2.5-VL-7B trên OmniDocBench có thể cần tới 3 000–7 000 vision tokens mỗi ảnh [1].

Số lượng vision tokens lớn khiến:

- **activation memory** tăng mạnh do self-attention phải xử lý chuỗi vision dài;
- **tốc độ suy luận chậm** do cả pha prefill lẫn generation đều bị kéo dài;
- khó mở rộng sang các thiết lập tài liệu độ phân giải cực cao hoặc xử lý hàng loạt tài liệu trong môi trường sản xuất.

Những hạn chế này là một trong các động lực chính dẫn tới việc DeepSeek-OCR tập trung vào *contexts optical compression*, đặt câu hỏi: với một tài liệu chứa khoảng 1 000 từ, **cần ít nhất bao nhiêu vision tokens để giải mã (decode) chính xác?** [1].

CHƯƠNG 4

PHƯƠNG PHÁP ĐỀ XUẤT

4.1. Ý tưởng chính

Nhóm nghiên cứu nhận thấy, một bức ảnh có chứa văn bản tài liệu có thể cho ra cùng lượng thông tin bằng cách sử dụng số token thị giác (vision token) **ít hơn đáng kể** so với các mô hình sử dụng văn bản số tương đương, cho thấy rằng việc **Nén ngữ cảnh quang học (Contexts Optical Compression)** thông qua các token thị giác có thể đạt được tỷ lệ nén cao hơn nhiều. Ví dụ, cùng một đoạn nội dung, nếu mã hoá bằng token văn bản thì cần khoảng 1000 token, còn khi biểu diễn bằng các vision token thì có thể chỉ cần khoảng 100 token (hoặc ít hơn). Như vậy, số lượng token giảm đi giúp cắt giảm đáng kể chi phí tính toán khi suy luận.

Nhận định trên thúc đẩy họ xem xét lại các mô hình Ngôn ngữ - Thị giác (VLMs) từ góc nhìn của mô hình Ngôn ngữ lớn (LLMs), vận dụng các bộ **Mã hóa thị giác (vision encoder)** để nâng cao hiệu quả **xử lý thông tin văn bản** của mô hình LLMs, thay vì chỉ vận dụng xử lý các tác vụ VQA cơ bản mà con người vốn đã thực hiện tốt.

Phương thức **Nhận dạng kí tự quang học (OCR)** là một phương thức **cầu nối** giữa Thị giác và Ngôn ngữ, cung cấp **môi trường kiểm thử lý tưởng** cho mô hình nén văn bản này, vì chúng thiết lập ánh xạ nén - giải nén tự nhiên giữa thông tin thị giác và văn bản, đồng thời cung cấp các thông số đánh giá định lượng.

4.2. Đầu vào của mô hình

Đầu vào **hình ảnh:** có hỗ trợ chế độ đa phân giải (chế độ phân giải tự nhiên và các chế độ phân giải động)

- **Chế độ phân giải tự nhiên (Native Resolution):** Chế độ này bao gồm bốn tùy chọn với kích thước và số lượng vision token tương ứng: `Tiny (512×512, 64 tokens)`, `Small (640×640, 100 tokens)`, `Base (1024×1024, 256 tokens)` và `Large (1280×1280, 400`

tokens)

- **Chế độ phân giải động (Dynamic Resolution):** Chế độ này được thiết kế chủ yếu cho các ứng dụng thực tế, đặc biệt là với các đầu vào có độ phân giải siêu cao (ví dụ: hình ảnh báo chí)

4.3. Đầu ra của mô hình

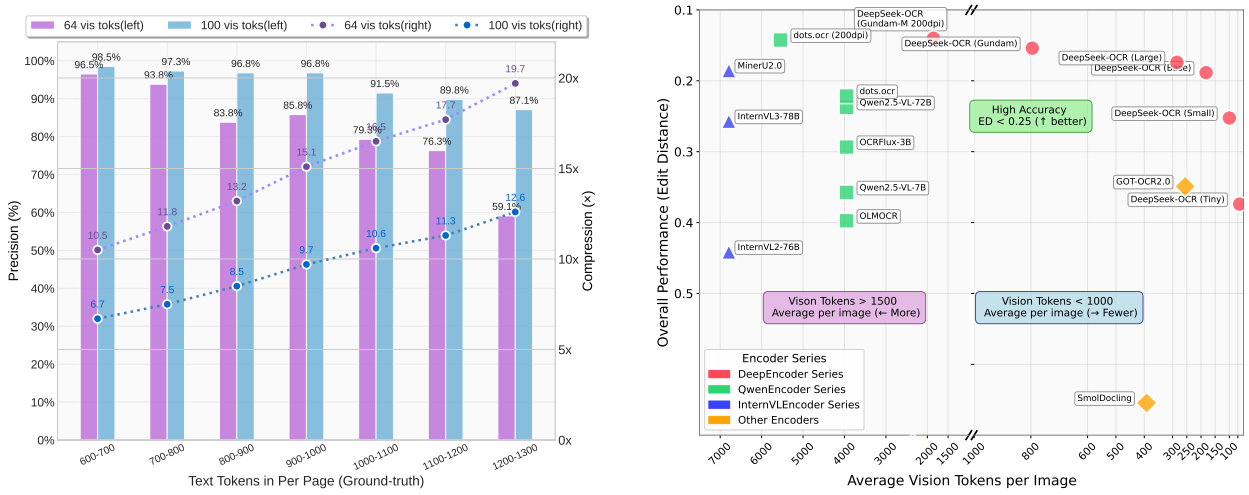
Văn bản được tái tạo: Bộ giải mã (DeepSeek3B-MoE) sử dụng các token thị giác đã được nén cùng với các câu lệnh prompt để tạo ra kết quả.

4.4. Điểm cải tiến so với các phương pháp hiện tại

Nén văn bản dài sử dụng phương thức thị giác, tỷ lệ nén cao nhưng vẫn bảo toàn thông tin, cụ thể: Mô hình đạt độ chính xác giải mã trên 96% ở mức nén văn bản 9 - 10×, ∼90% ở mức nén 10 - 12×, và ∼60% ở mức nén 20× trên các bộ kiểm thử Fox[2] với bố cục tài liệu đa dạng (độ chính xác thực tế thậm chí còn cao hơn khi xét đến sự khác biệt giữa đầu ra và dữ liệu thực tế), như minh hoạ trong Hình 4.1a.

Kiến trúc DeepEncoder được thiết kế hiệu quả: Đây là một kiến trúc mới lạ bảo đảm duy trì bộ nhớ kích hoạt thấp và số lượng token thị giác tối thiểu ngay cả khi hình ảnh đầu vào có độ phân giải cao bằng cách tích hợp các thành phần mã hóa sử dụng chú ý cục bộ (window attention) và chú ý toàn cục (global attention) thông qua Bộ nén 16x (16x convolutional compressor). Thiết kế này đảm bảo thành phần chú ý cục bộ sẽ xử lý lượng lớn token thị giác ban đầu, trong khi bộ nén sẽ làm giảm số lượng token thị giác trước khi đi vào thành phần chú ý toàn cục, từ đó cải thiện hiệu quả nén và hiệu suất bộ nhớ.

Hiệu suất dẫn đầu và giá trị thực tiễn: kiến trúc DeepSeek-OCR dựa trên DeepEncoder và DeepSeek3B-MoE [3] [4]. Hình 4.1b cho thấy, đây là mô hình đạt hiệu suất hàng đầu trong các mô hình đầu cuối trên bộ dữ liệu OmniDocBench trong khi sử dụng ít token thị giác nhất. Bên cạnh đó, mô hình cũng có khả năng phân tích biểu đồ, công thức hóa học, hình học cơ bản và hình ảnh tự nhiên để nâng cao khả năng áp dụng thực tiễn.Trong sản xuất, DeepSeek-OCR có thể tạo ra 33 triệu trang dữ liệu mỗi ngày cho các LLMs hoặc VLMs khi sử dụng 20 node (mỗi node gồm 8 GPU A100-40G).

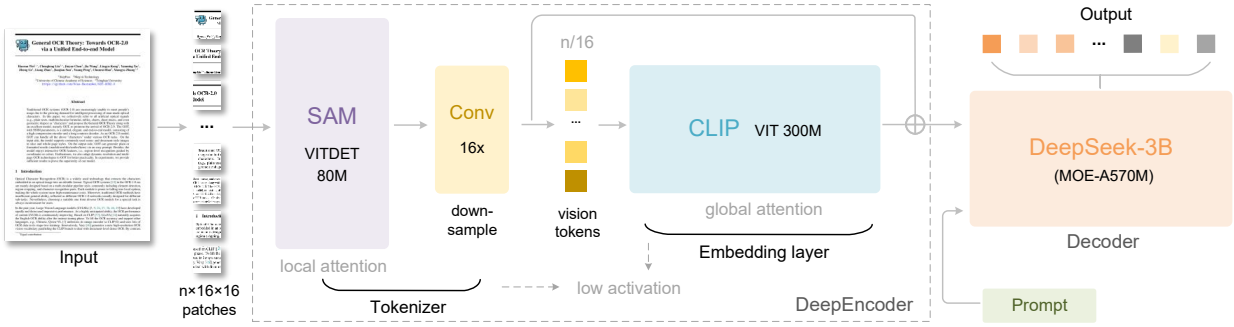


(a) (b)

Hình 4.1: Hình (a) thể hiện tỷ lệ nén (số lượng token văn bản trong dữ liệu thực tế / số lượng token thị giác mà mô hình sử dụng) được kiểm thử trên bộ benchmark Fox; Hình (b) thể hiện các so sánh hiệu suất trên OmniDocBench. DeepSeek-OCR đạt được hiệu suất hàng đầu trong số các mô hình đầu cuối trong khi sử dụng số lượng token thị giác ít nhất.

CHƯƠNG 5

MÔ HÌNH ĐỀ XUẤT



Hình 5.1: Kiến trúc tổng thể mô hình DeepSeek-OCR (Nguồn: [1])

5.1. Tổng quan

Các mô hình hiện tại vẫn còn gặp một số khó khăn chưa được giải quyết hoàn toàn:

1. Khó xử lý hình ảnh có độ phân giải cao (ví dụ: 1024×1024), yêu cầu chi phí tính toán lớn.
2. Bộ nhớ kích hoạt tăng nhanh ở độ phân giải cao. Nhiều mô hình hiện nay có chi phí suy luận tăng theo hàm bậc hai. Cần một mô hình với chi phí suy luận thấp.
3. Sử dụng số lượng vision tokens lớn.
4. Chưa hỗ trợ nhiều độ phân giải đầu vào khác nhau.
5. Số lượng tham số quá lớn.

Để giải quyết các vấn đề trên, mô hình **DeepSeek-OCR** được đề xuất như trong Hình 5.1. Mô hình hỗ trợ nhiều chế độ độ phân giải và số lượng vision tokens khác nhau:

- Tiny: 512×512, 64 tokens
- Small: 640×640, 100 tokens

- Base: 1024 × 1024, 256 tokens
- Large: 1280 × 1280, 400 tokens
- Chế độ tự động (Gundam) cho các ứng dụng thực tế

Mô hình gồm hai thành phần chính: **DeepEncoder** và **Decoder**:

DeepEncoder (khoảng 380M tham số)

DeepEncoder bao gồm các thành phần như:

- **SAM (Segment Anything Model)** [8]: 80M tham số, học các đặc trưng cục bộ.
- **Convolution 16×** [1]: nén token, giảm số lượng vision tokens đầu vào.
- **CLIP (ViT)** [9]: 300M tham số, học các đặc trưng toàn cục.

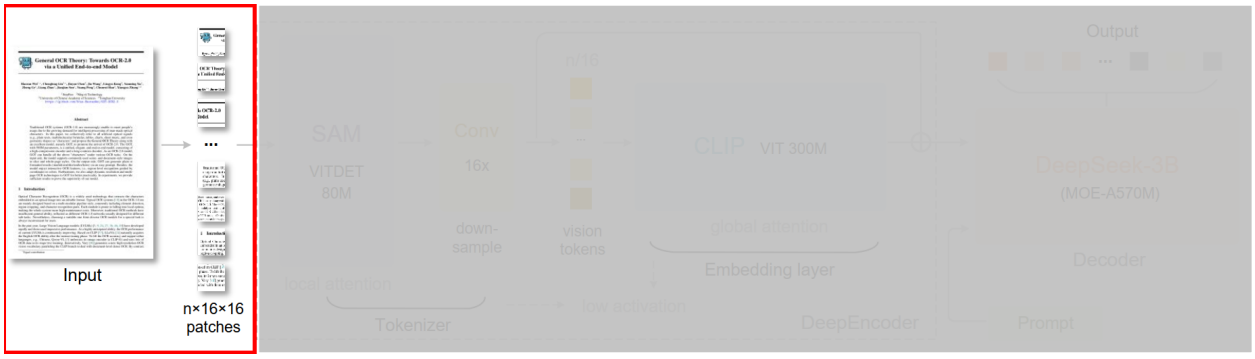
Decoder

Phần Decoder sử dụng mô hình **DeepSeek-3B-MoE (3B tham số khi huấn luyện và khi suy luận sử dụng 570M tham số**

- Sử dụng kiến trúc MoE (Mixture of Experts) với tổng 3B tham số, đóng vai trò như một mô hình ngôn ngữ lớn để giải mã các tokens đã học phía trước nhờ vào câu promt.
- Trong quá trình suy luận chỉ kích hoạt 6 experts + 2 shared experts ⇒ hiệu quả chi phí tính toán như mô hình 570M.
- Giúp tận dụng sức mạnh của một mô hình 3B tham số nhưng chi phí khi suy luận như mô hình 570M tham số (do khi suy luận chỉ các chuyên gia (expert) phù hợp thì mới hoạt động).

Như vậy, mô hình giải quyết các vấn đề:

- **Vấn đề 1:** nhờ cơ chế nén token.
- **Vấn đề 2:** giảm bộ nhớ kích hoạt.
- **Vấn đề 3 và 4:** hỗ trợ nhiều độ phân giải và ít token.
- **Vấn đề 5:** kích thước mô hình vừa phải.



Hình 5.2: Phần đầu vào của mô hình (Nguồn: [1])

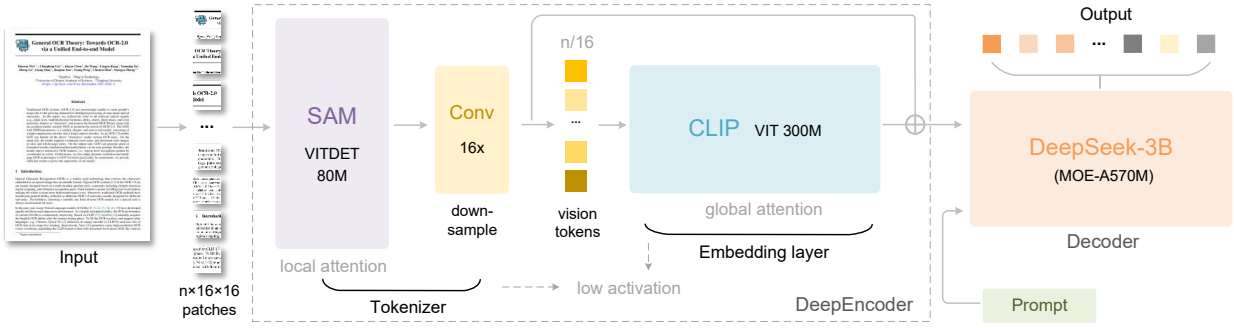
5.2. Đầu vào mô hình

Đầu vào là ảnh (ví dụ: một trang tài liệu) được chia thành các *patches* kích thước 16×16, tức là đầu vào là $n \times 16 \times 16$ patches.

Ví dụ: ảnh 1024×1024 chia thành:

$$\left(\frac{1024}{16}\right)^2 = 64 \times 64 = 4096 \text{ patches}$$

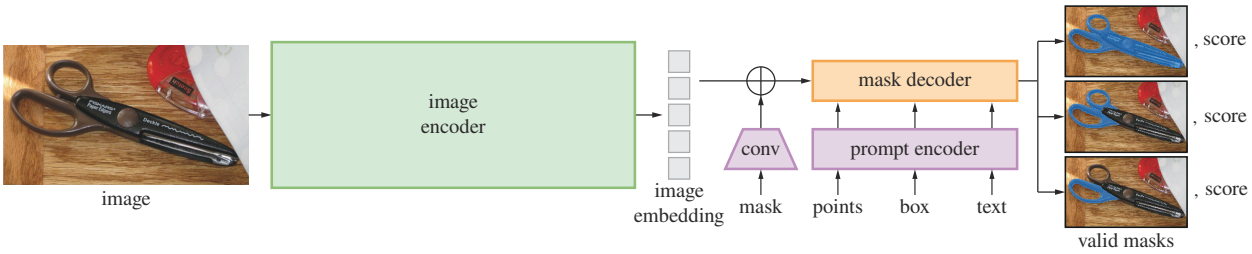
5.3. DeepEncoder



Hình 5.3: Kiến trúc phần DeepEncoder (Nguồn: [1])

DeepEncoder (∼380M tham số) gồm 3 thành phần chính:

- **SAM-base** [8]: 80M tham số
- **Convolution 16×** [1]: giảm token với số lượng tham số nhỏ.
- **CLIP-large** [9]: 300M tham số

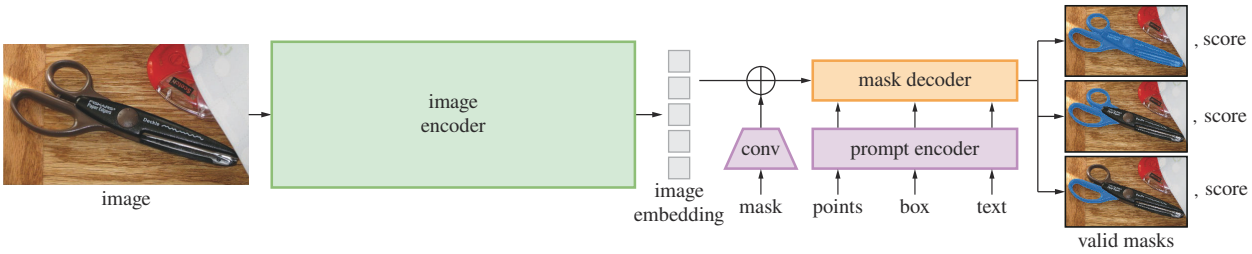


Hình 5.4: Kiến trúc mô hình SAM (Nguồn: [8])

5.3.1. SAM (Segment Anything Model) [8]

- Dùng kiến trúc SAM-base (80M tham số) để trích xuất đặc trưng cục bộ. Việc học các đặc trưng cục bộ (học những đặc trưng trong một window) từ đó tiết kiệm chi phí hơn so với học toàn cục (global).
- Sử dụng **window attention** để giữ chi phí bộ nhớ thấp ở độ phân giải cao.
- Chỉ sử dụng phần **image encoder** (dựa trên ViT) từ SAM, không dùng phần segment head và sử dụng pretrain của mô hình gốc.
- Về bản chất, theo góc nhìn của nhóm đây là một mô hình khá mạnh trong lĩnh vực segment ảnh nên từ đó mô hình sẽ giữ lại các đặc trưng cục bộ một cách mạnh mẽ.
- Input: $n \times 16 \times 16$
- Output: $n \times \text{patch} \times \text{dim}_{\text{SAM}}$

5.3.2. Convolution 16×



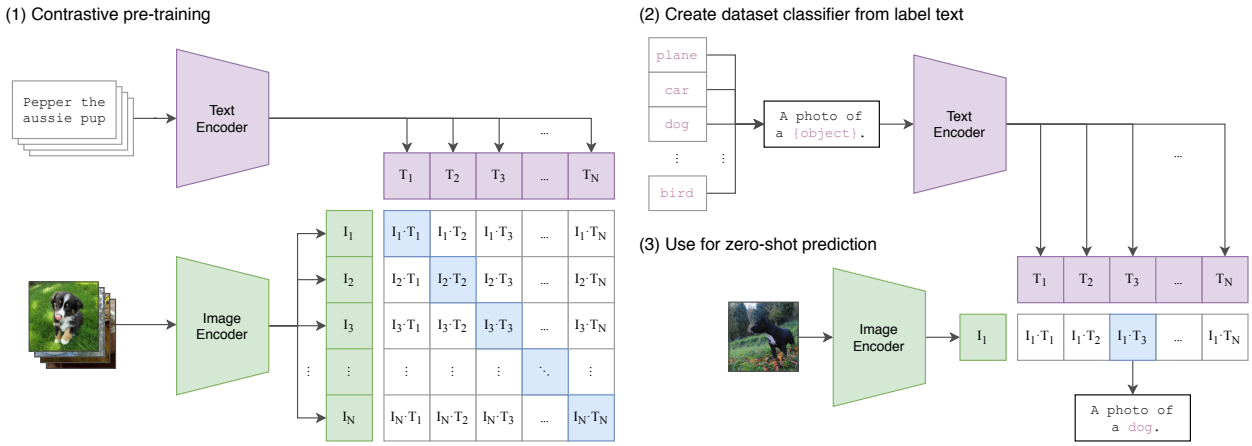
Hình 5.5: Kiến trúc nén Conv16× (Nguồn: [1])

- Ở đây các tokens được nén nhờ vào 2 lớp conv liên tiếp với kernel size = 3, stride = 2, padding = 1 được lấy ý tưởng từ mô hình Vary [1]. Nhờ vào cơ chế nén này đã giảm đi số lượng vision tokens, tiết kiệm khi xử lý global attention phía sau.
- Mặc dù là nén 16 lần nhưng nhờ vào mô hình SAM ở trước đã giữ lại khá mạnh các đặc

trưng nổi trổi nến việc nén thì vẫn đảm bảo các thông tin. Channels tăng 4 lần.

- Input: $n \times \text{patch} \times \text{dim}_{\text{SAM}}$
- Output: $n \times \frac{\text{patch}}{16} \times \text{dim}_{\text{Conv}}$

5.3.3. CLIP (Contrastive Language-Image Pre-training)

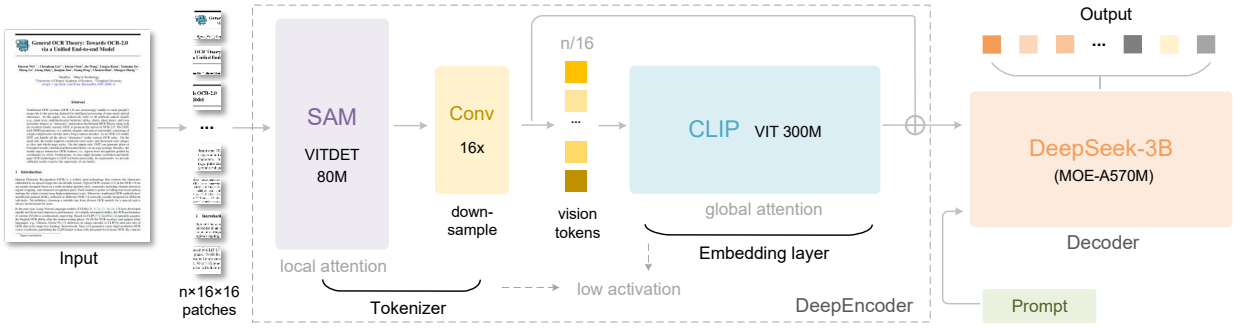


Hình 5.6: Kiến trúc CLIP (Nguồn: [9])

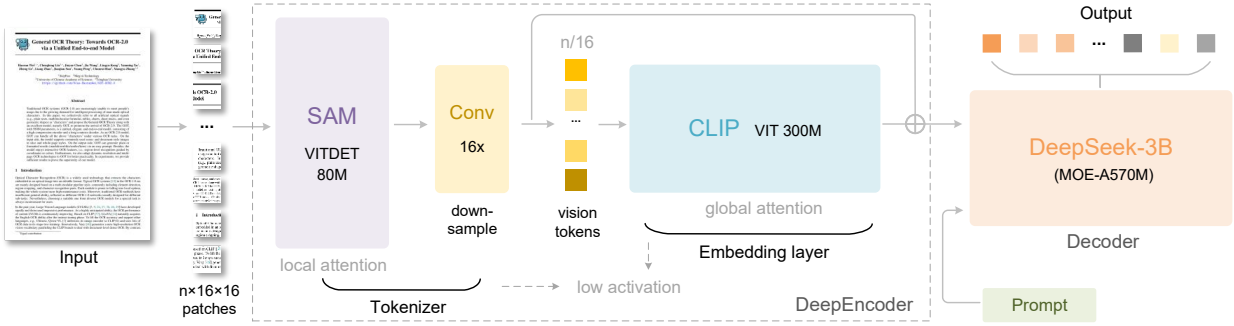
- Dựa trên CLIP-large (300M thamsố) để trích xuất kiến thức thị giác.
- Sử dụng dense global attention với vai trò học các đặc trưng mang ý nghĩa toàn cục và học quan hệ ngữ nghĩa sâu bên trong.
- Bỏ lớp patch embedding đầu tiên vì input không còn là ảnh gốc mà ở đây áp dụng cho các tokens.
- Output kết hợp với **skip connection** để giữ thông tin (tránh bị mất mát trong quá trình học toàn cục). Từ đó mà giúp output giàu ngữ nghĩa hơn rất nhiều.
- Mô hình sử dụng Clip đóng vai trò học các đặc trưng toàn cục. Theo góc nhìn của nhóm, mô hình CLIP bản chất là một mô hình học có sự tương tác giữa ảnh và chữ. Vì thế, dù sao khi nén nhưng mô hình vân học tốt, do có sự giữ lại phần ngữ nghĩa của chữ vào cac tokens nhờ vào cơ chế của CLIP.

5.4. Decoder

- Sử dụng mô hình **DeepSeek-3B-MoE** [3, 4], kiến trúc Mixture of Experts. Có thể xem đây là một mô hình ngôn ngữ lớn phiên bản nhỏ do chỉ sử dụng mô hình 3B tham số. Với kiến trúc đặc trưng của DeepSeek là Mixture of Expert (MoE), mô hình hình gồm nhiều



Hình 5.7: Skip-connection trong CLIP (Nguồn: [1])



Hình 5.8: Kiến trúc Decoder [1]

expert và mỗi expert sẽ chuyên biệt giải quyết các nhiệm vụ cụ thể.

- Mô hình khi huấn luyện mô hình 3B tham số, khi suy luận chỉ sử dụng 500M tham số. Đây là một tính chất khá đặc biệt của kiến trúc MoE giúp mô hình sử dụng sức mạnh của một mô hình 3B tham số nhưng suy luận chỉ hoạt động 500M tham số. Các expert không liên quan sẽ không hoạt động.
- Mô hình sử dụng 64 experts, trong suy luận chỉ kích hoạt 570M tham số (6 expert + 2 shared). Việc sử dụng expert chia sẻ để tránh mất mát các thông tin.
- Tái tạo văn bản từ vision tokens + prompt đầu vào.
- Input: vision tokens từ encoder + prompt
- Output: văn bản OCR (dạng $n \times T$, với T là số tokens sinh ra)

CHƯƠNG 6

THỰC NGHIỆM VÀ KẾT QUẢ CỦA TÁC GIẢ

6.1. Dữ liệu huấn luyện của mô hình DeepSeek-OCR

Nhóm tác giả xây dựng một tập dữ liệu huấn luyện lớn, phức tạp và đa dạng cho DeepSeek-OCR, bao gồm ba nhóm chính: dữ liệu OCR 1.0, dữ liệu OCR 2.0 và dữ liệu thị giác tổng quát. Ngoài ra, một phần dữ liệu văn bản thuần cũng được bổ sung nhằm duy trì khả năng xử lý ngôn ngữ của mô hình.

Trong tổng thể quá trình huấn luyện, dữ liệu OCR chiếm khoảng 70%, dữ liệu thị giác tổng quát chiếm 20%, và dữ liệu văn bản thuần chiếm 10%.

6.1.1. Tổng quan các bộ dữ liệu sử dụng

Bảng 6.1: Tổng hợp các bộ dữ liệu OCR và Vision

Tên bộ dữ liệu	**Nguồn**	**Kích thước**	**Đặc điểm chính**
OCR 1.0: Documents	Internet (PDFs)	30 triệu trang	Hơn 100 ngôn ngữ; gán nhãn thô bằng fitz và gán nhãn tinh bằng PP-DocLayout, MinerU, GOT-OCR2.0.
OCR 1.0: Scene Images	LAION, Wukong	20 triệu ảnh	Ảnh cảnh vật tự nhiên, hỗ trợ tiếng Trung–Anh, gán nhãn bằng PaddleOCR.
OCR 1.0: Word Data	Trích xuất trực tiếp	3 triệu mẫu	Cặp ảnh–văn bản chất lượng cao, tập trung vào công thức và bảng HTML.
OCR 2.0: Charts	pyecharts, matplotlib	10 triệu ảnh	Biểu đồ đường, cột, tròn; tác vụ chuyển ảnh sang bảng HTML.
OCR 2.0: Chemical Formulas	PubChem	5 triệu cặp	Biểu diễn công thức hóa học bằng định dạng SMILES.
OCR 2.0: Geometry	Slow Perception	1 triệu mẫu	Hình học phẳng với nhãn đoạn thẳng, tọa độ, loại hình.
General Vision Data	LAION	20% dữ liệu	Caption, detection, grounding để duy trì khả năng hiểu ảnh tổng quát.
Text-only Data	Nội bộ	10% dữ liệu	Chuỗi văn bản độ dài 8192 token nhằm bảo toàn năng lực ngôn ngữ.

6.1.2. OCR 1.0

Dữ liệu tài liệu là ưu tiên hàng đầu của DeepSeek-OCR. Nhóm tác giả thu thập 30 triệu trang PDF đa ngôn ngữ từ Internet, trong đó tiếng Trung và tiếng Anh chiếm khoảng 25 triệu trang, phần còn lại là các ngôn ngữ thiểu số.

Hai dạng chú thích được xây dựng gồm:

- **Chú thích thô**: Trích xuất trực tiếp bằng fitz, nhằm dạy mô hình nhận diện văn bản quang học, đặc biệt hữu ích cho các ngôn ngữ ít tài nguyên.
- **Chú thích tinh**: Khoảng 2 triệu trang cho mỗi ngôn ngữ Trung và Anh, được gán nhãn bằng các mô hình layout nâng cao và OCR để xây dựng dữ liệu xen kẽ phát hiện–nhận dạng.

Ngoài ra, nhóm tác giả xây dựng 3 triệu mẫu Word Data chất lượng cao bằng cách trích xuất trực tiếp nội dung, giúp cải thiện khả năng nhận diện công thức toán học và bảng biểu HTML. Với OCR cảnh vật tự nhiên, dữ liệu được lấy từ LAION và Wukong, tập trung chủ yếu vào tiếng Trung và tiếng Anh.

6.1.3. OCR 2.0

OCR 2.0 mở rộng từ OCR truyền thống sang các tác vụ phân tích hình ảnh nhân tạo phức tạp. Dữ liệu bao gồm:

- **Biểu đồ**: 10 triệu hình ảnh được kết xuất từ pyecharts và matplotlib, với mục tiêu chuyển ảnh thành bảng HTML.
- **Công thức hóa học**: 5 triệu cặp ảnh–văn bản, sử dụng dữ liệu SMILES từ PubChem và kết xuất bằng RDKit.
- **Hình học phẳng**: 1 triệu mẫu tổng hợp bằng Slow Perception, kết hợp tăng cường dữ liệu dựa trên bất biến tịnh tiến hình học.

6.1.4. Dữ liệu thị giác tổng quát

Mặc dù DeepSeek-OCR không phải là mô hình VLM đa năng, nhóm tác giả vẫn bổ sung dữ liệu caption, detection và grounding nhằm duy trì giao diện thị giác tổng quát. Nhóm dữ liệu này chiếm khoảng 20% tổng dữ liệu huấn luyện.

6.1.5. Dữ liệu văn bản thuần

Khoảng 10% dữ liệu huấn luyện là dữ liệu văn bản thuần từ nguồn nội bộ, được chuẩn hóa về độ dài 8192 token để phù hợp với kiến trúc DeepSeek-OCR.

6.2. Quy trình huấn luyện

Quy trình huấn luyện gồm hai giai đoạn chính:

1. Huấn luyện DeepEncoder độc lập.

2. Huấn luyện mô hình DeepSeek-OCR hoàn chỉnh.

DeepEncoder được huấn luyện theo khung dự đoán token kế tiếp với dữ liệu OCR 1.0, OCR 2.0 và 100 triệu mẫu dữ liệu thị giác tổng quát. Sau đó, DeepSeek-OCR được huấn luyện trên nền tảng HAI-LLM với pipeline parallelism, sử dụng 20 node GPU A100-40G.

6.3. Dữ liệu và kết quả thực nghiệm của bài báo

6.3.1. Fox Benchmarks

Fox Benchmarks gồm 100 trang tài liệu tiếng Anh với độ dài 600–1300 token, dùng để đánh giá khả năng nén và giải nén văn bản của DeepSeek-OCR.

Hai thước đo được sử dụng:

- Tỷ lệ nén (Compression Ratio)

- Độ chính xác (Precision)

Kết quả cho thấy với văn bản 600–700 token, mô hình đạt tỷ lệ nén 10.5 lần với độ chính xác 96.5%. Ngay cả khi tỷ lệ nén tăng lên gần 20 lần, độ chính xác vẫn duy trì ở mức chấp nhận được.

6.3.2. OmniDocBench

Trên OmniDocBench, DeepSeek-OCR được đánh giá bằng thước đo khoảng cách chỉnh sửa (Edit Distance). Kết quả cho thấy mô hình vượt qua GOT-OCR2.0 dù chỉ sử dụng 100 vision token mỗi trang, khẳng định hiệu quả của kiến trúc OCR 2.0.

CHƯƠNG 7

THỰC NGHIỆM CỦA NHÓM

7.1. Quan điểm của nhóm về bài báo, ứng dụng

- Nhóm đánh giá phương pháp **nén quang học (Optical Compression)** là một ý tưởng tiềm năng và khả thi cho bài toán xử lý ngữ cảnh dài. Mô hình có thể đạt hiệu quả nén tới 10× mà vẫn duy trì độ chính xác OCR khoảng 97%, cho thấy giá trị ứng dụng thực tế rõ rệt.
- Về mặt định hướng nghiên cứu, phương pháp này có thể mở ra hướng phát triển cho các kiến trúc xử lý ngữ cảnh dài theo cơ chế "nén" các đoạn ngữ cảnh cũ về dạng ảnh độ phân giải thấp để tiết kiệm bộ nhớ và tài nguyên, trong khi vẫn giữ thông tin cốt lõi.
- Cơ chế giảm dần độ phân giải để nén cũng gợi liên tưởng đến cơ chế "lãng quên" của con người, là một hướng đáng chú ý nếu phát triển các hệ thống học sâu có bộ nhớ ngữ cảnh động.
- Nhận thấy kiến trúc hiện tại hoạt động tốt trên tiếng Trung và tiếng Anh, nhưng chưa có thử nghiệm đầy đủ trên các ngôn ngữ khác. Nhóm quyết định **thích nghi và huấn luyện lại** DeepSeek-OCR cho **tiếng Việt**.
- Bên cạnh đó, nhóm quan tâm mở rộng mô hình để xử lý **chữ viết tay**. Với đặc tính mạnh về nhận dạng và học đặc trưng cục bộ, nhóm kỳ vọng có thể phát triển các biến thể chuyên cho **handwriting OCR**.

7.2. Dữ liệu

7.2.1. Dữ liệu thực nghiệm của nhóm

Bộ dữ liệu sử dụng trong dự án là HANDS-VNOnDB2018 – một bộ dữ liệu chữ viết tay tiếng Việt được phát triển bởi nhóm nghiên cứu tại Đại học Bách Khoa Hà Nội cùng các đối tác. Bộ dữ liệu này đã được xây dựng, kiểm định và công bố phục vụ cho mục đích nghiên

cứu về nhận dạng chữ viết tay tiếng Việt. Thông tin chi tiết và đường dẫn truy cập bộ dữ liệu: VNOnDB.

7.2.2. Phân tích dữ liệu khám phá (EDA)

Thống kê tổng quan chất lượng dữ liệu

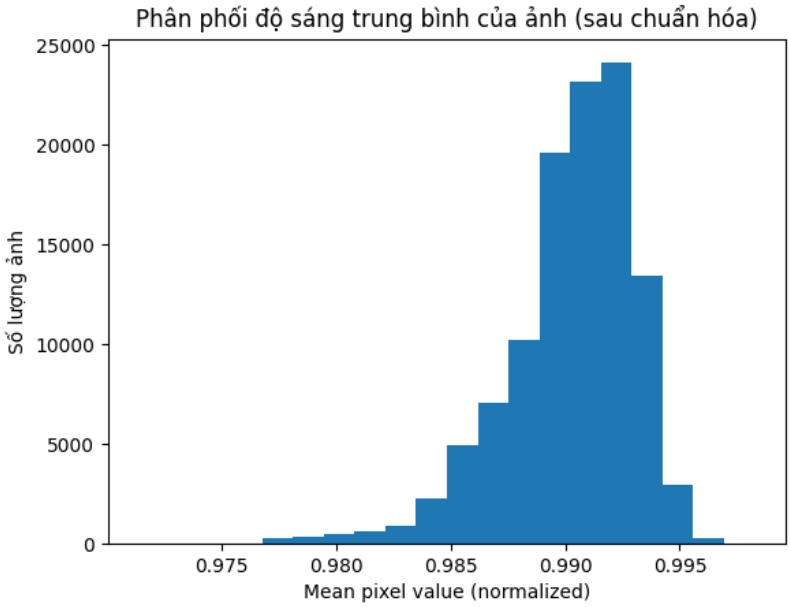
- **Tổng số ảnh**: 110746
- **Tổng số ảnh duy nhất**: 110746
- **Tổng số nhãn khác nhau**: 3511

Độ phân giải ảnh

- **Kích thước (shape) của mỗi ảnh**: 32×128
- Tất cả ảnh đều có cùng độ phân giải, đảm bảo tính đồng nhất cho mô hình học máy

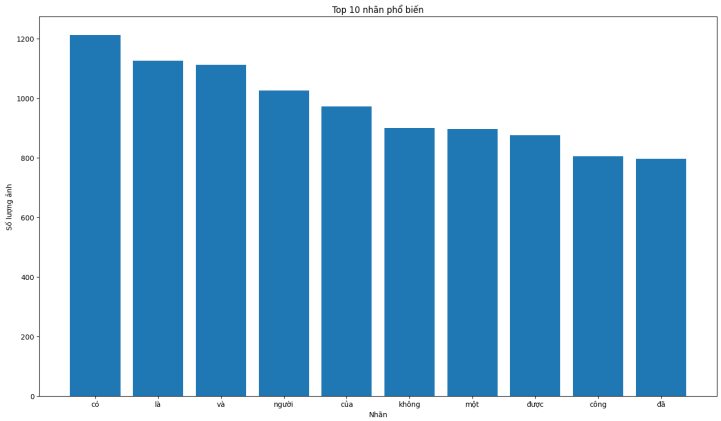
Phân bố độ sáng của ảnh

- **Giá trị pixel trung bình (mean)**: 0.99
- **Giá trị pixel nhỏ nhất (min)**: 0.0
- **Giá trị pixel lớn nhất (max)**: 1.0



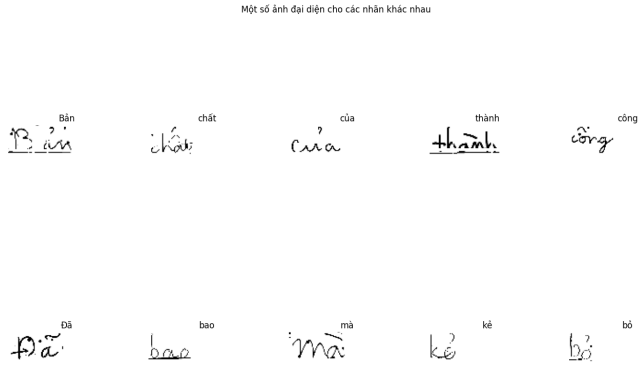
Hình 7.1: Mô tả cho hình ảnh

Phân bố nhãn



Hình 7.2: Top 10 nhãn phổ biến

Visualize một số ảnh mẫu



Hình 7.3: Một số ảnh và nhãn

7.3. Thiết lập thực nghiệm

7.3.1. Framework và thư viện sử dụng

PyTorch: Framework deep learning chính để huấn luyện mô hình.

Unsloth: Tối ưu fine-tuning, giúp tăng tốc 2–5× và giảm tiêu thụ VRAM.

Hugging Face Transformers: Quản lý model, tokenizer và pipeline huấn luyện.

PEFT (LoRA): Fine-tuning hiệu quả tham số, chỉ huấn luyện adapter thay vì toàn bộ model.

TRL: Hỗ trợ Supervised Fine-Tuning (SFT).

PIL (Pillow): Xử lý và tiền xử lý ảnh đầu vào.

jiwer: Tính toán các chỉ số đánh giá OCR như CER và WER.

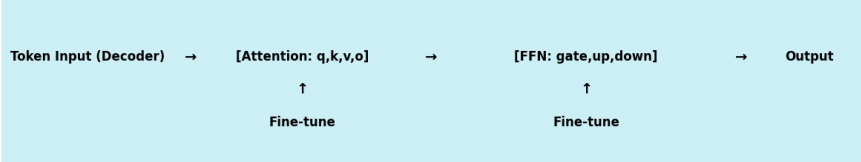
7.3.2. Các quyết định thiết kế

Base model

unsloth/DeepSeek-OCR – mô hình Vision-Language chuyên cho OCR.

Phương pháp fine-tuning

LoRA (Low-Rank Adaptation) [5], chỉ train thêm các adapter nhỏ thay vì toàn bộ model (chủ yếu fine tune ở decoder trong các block transformer).



Hình 7.4: Quyết định thiết kế quan trọng

Chọn Modules:

Bảng 7.1: Feed-Forward Network (FFN / MLP) Modules

Module	Tên đầy đủ	Chức năng
gate_proj	Gate Projection	Điều khiển luồng thông tin (kết hợp Swish/GELU activation)
up_proj	Up Projection	Mở rộng chiều hidden (expand dimension)
down_proj	Down Projection	Thu nhỏ về chiều ban đầu (project back)

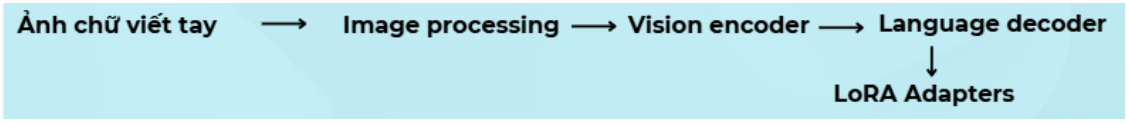
Bảng 7.2: Attention Modules

Module	**Tên đầy đủ**	**Chức năng**
q_proj	Query Projection	Tạo vector Query – biểu diễn *"Tôi đang tìm gì?"*
k_proj	Key Projection	Tạo vector Key – biểu diễn *"Tôi chứa thông tin gì?"*
v_proj	Value Projection	Tạo vector Value – biểu diễn *"Thông tin thực sự của tôi"*
o_proj	Output Projection	Kết hợp kết quả attention để tạo output cuối cùng

Lý do lựa chọn

- **Bao phủ toàn bộ Transformer block:** Attention + FFN = 2 thành phần chính
- **Hiệu quả cao:** Các projection layers có nhiều parameters nhưng LoRA chỉ thêm 0.1-1
- **Thay đổi hành vi model:**
  - Attention → model học nhìn vào đâu trong ảnh chữ viết tay
  - FFN → model học hiểu và decode chữ tiếng Việt

Pipeline



Hình 7.5: Tổng quan pipeline

Bước 1: Tiền xử lý ảnh

- **Dynamic Cropping**: Chia ảnh có kích thước lớn thành từ 2 đến 9 vùng (patches).
- **Global View**: Toàn bộ ảnh được resize về kích thước 1024×1024 (`base_size`).
- **Local Patches**: Các patch cục bộ được resize về kích thước 640×640 (`image_size`).

• **Tokenization**: Chuyển các image patches thành các image tokens để đưa vào mô hình.

Bước 2: LoRA Adaptation

LoRA (Low-Rank Adaptation) [5] chỉ bổ sung hai ma trận có kích thước nhỏ vào trọng số ban đầu của mô hình:

$$W' = W + \Delta W = W + BA$$

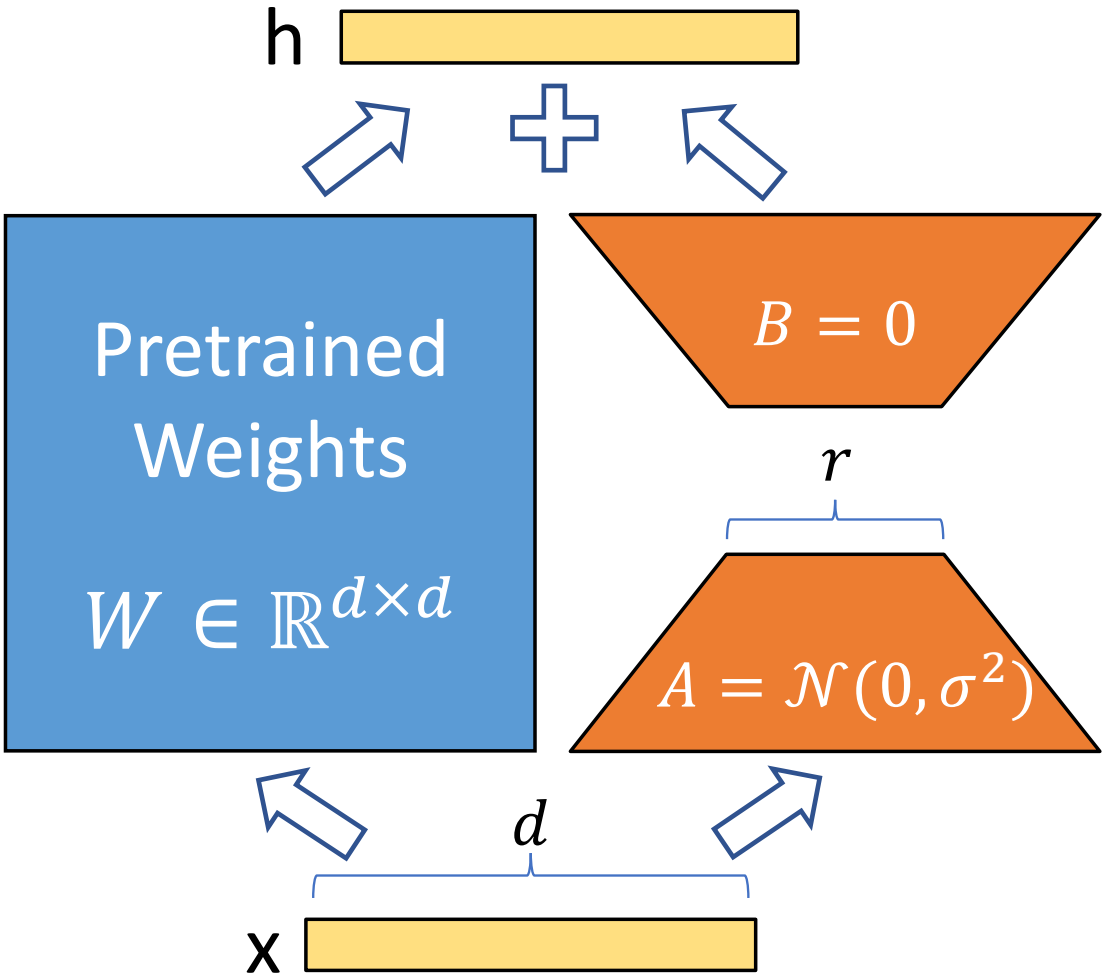
Trong đó:

• $W \in \mathbb{R}^{d \times k}$: Ma trận trọng số gốc của mô hình, được giữ cố định (frozen)

• $B \in \mathbb{R}^{d \times r}$: Ma trận down-projection

• $A \in \mathbb{R}^{r \times k}$: Ma trận up-projection

• $r = 16$: Rank, biểu diễn số chiều thấp được sử dụng trong LoRA



Hình 7.6: Minh họa LoRA Adaptation

Hàm mất mát

Hàm loss được sử dụng là *Cross-Entropy Loss*, trong đó chỉ tính loss trên các token thuộc phần phản hồi của mô hình (`train_on_responses_only`).

DeepSeek-OCR:

- **Khả năng dự đoán:** Mô hình hầu như không dự đoán chính xác được *toàn bộ từ*; các kết quả dự đoán thường chỉ đúng một phần ký tự trong từ.
- **Char Accuracy (Soft):** −0.196. Mặc dù giá trị này âm, kết quả cho thấy mô hình vẫn nhận diện đúng khoảng **80% số ký tự** trong các từ dự đoán.
- **Average Similarity:** 0.043. Độ tương đồng trung bình giữa chuỗi dự đoán và nhãn chuẩn ở mức rất thấp, phản ánh việc mô hình chưa khớp tốt về mặt ngữ nghĩa ở cấp độ từ.

7.3.3. Cấu hình thực nghiệm

Cấu hình thực nghiệm:

- **Môi trường:** Kaggle, Google Colab (1 GPU, Tesla T4)
- **VRAM:** Tận dụng tối ưu bộ nhớ GPU thông qua kỹ thuật Gradient Checkpointing và Quantization.
- **Thư viện hỗ trợ Fine-tuning:** Unsloth (được sử dụng để tăng tốc độ huấn luyện và giảm tiêu thụ bộ nhớ VRAM), kết hợp với thư viện Hugging Face Transformers và PEFT.
- **Framework Deep Learning:** PyTorch.
- **Trình tối ưu hóa:** AdamW 8-bit (giúp tiết kiệm bộ nhớ so với AdamW 32-bit truyền thống).

Hyperparameters:

Bảng 7.3: Tổng hợp các Hyperparameters huấn luyện

Parameter	**Giá trị**	**Mô tả**
Learning Rate	0.0002	Tốc độ học của mô hình
Batch Size	2	Số mẫu trong mỗi batch
Gradient Accumulation	4	Tích lũy gradient; batch hiệu dụng = $2 \times 4 = 8$

Parameter	**Giá trị**	**Mô tả**
Max Steps	1000	Tổng số bước cập nhật trọng số trong quá trình huấn luyện
Warmup Steps	20	Số bước warmup cho learning rate
Epoch	1	Một vòng quét toàn bộ tập dữ liệu huấn luyện; trong thiết lập này epoch chỉ mang tính tham chiếu do quá trình huấn luyện bị giới hạn bởi số bước
Optimizer	AdamW 8-bit	Thuật toán tối ưu hóa trọng số với độ chính xác 8-bit
LoRA Rank (r)	16	Rank của ma trận low-rank trong LoRA
LoRA Alpha	16	Hệ số scale cho LoRA ($\alpha/r = 1$)
LoRA Dropout	0	Không sử dụng dropout trong LoRA
Seed	3407	Giá trị khởi tạo ngẫu nhiên để đảm bảo khả năng tái lập kết quả
Validation Ratio	10%	Tỷ lệ dữ liệu dùng cho validation
Scheduler	Linear	Giảm tuyến tính learning rate theo thời gian huấn luyện

7.4. Kết quả thực nghiệm

7.4.1. Metrics đánh giá

Word Error Rate (WER)

Công thức tính sai số ở cấp độ từ (Word Error Rate - WER) được định nghĩa như sau:

$$\text{WER} = \frac{S + D + I}{N} \tag{7.1}$$

Trong đó:

- S (Substitutions): số từ bị thay thế (nhận diện nhầm).

- D (Deletions): số từ bị xóa (không nhận diện được).
- I (Insertions): số từ bị chèn thêm (nhận diện thừa).
- N: tổng số từ trong nhãn chuẩn (ground truth).

Character Error Rate (CER)

Công thức tính sai số ở cấp độ kí tự (Character Error Rate - CER) tương tự như WER nhưng áp dụng cho từng kí tự:

$$\text{CER} = \frac{S_c + D_c + I_c}{N_c} \tag{7.2}$$

Trong đó:

- S_c: số kí tự bị thay thế.
- D_c: số kí tự bị xóa.
- I_c: số kí tự bị chèn thêm.
- N_c: tổng số kí tự trong nhãn chuẩn (ground truth).

7.4.2. Kết quả của nhóm

Model	**WER**	**CER**
DeepSeek-OCR (original)	1	1,17138599105812
DeepSeek-OCR(fine-tune)	1	1,20588235294117
TrOCR	1,655	1,4327917282127

DeepSeek-OCR (original):

- **WER = 1:** Nghĩa là sai 100% số từ. Mô hình không đoán trúng được từ nào, hoặc sinh ra, thay thế một chuỗi ký tự vô nghĩa không trùng khớp với nhãn.
- **CER ≈ 1.17 (117%):** Số lượng lỗi sửa đổi (thêm, xóa, sửa) còn nhiều hơn cả tổng số ký tự của văn bản gốc.
- **Nhận xét:** Dù là mô hình tốt nhất trong bảng, nhưng kết quả này cho thấy mô hình Pre-train chưa hoạt động được trên tập dữ liệu kiểm thử của nhóm.

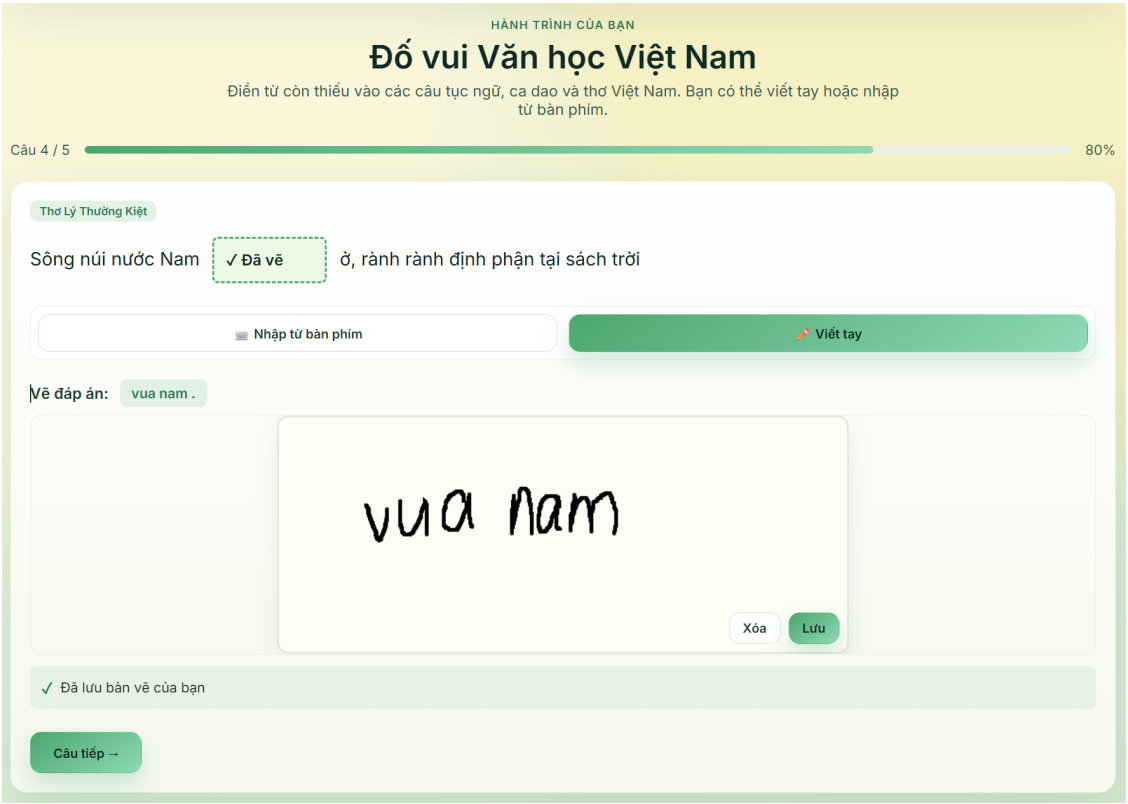
DeepSeek-OCR (fine-tune):

- **WER = 1:** Mô hình vẫn không dự đoán đúng toàn bộ số từ.
- **CER ≈ 1.20 (120%):** Số lượng lỗi sửa đổi (thêm, xóa, sửa) vẫn nhiều hơn cả tổng số ký tự của văn bản gốc.
- **Nhận xét:** Mặc dù đã qua quá trình fine-tune, nhưng tỉ lệ lỗi vẫn không cải thiện, có thể mô hình chưa hoạt động tốt với bộ dữ liệu mà nhóm đã lựa chọn (dữ liệu train quá ít...).
- Ở DeepSeek OCR: dự đoán gần như không đúng nguyên từ nào, như vậy nhóm thử đánh giá thêm 2 phương pháp đánh giá:
  - Char Accuracy (Soft): -0.196 (80% ký tự đúng)
  - Average Similarity: 0.043 (độ tương đồng rất thấp)
- **Nhận xét:** Mặc dù mô hình có thể nhận diện đúng một phần ký tự, hiệu quả tổng thể vẫn ở mức thấp trên tập dữ liệu mà nhóm lựa chọn. Đặc biệt, giá trị *Char Accuracy (Soft)* âm cho thấy chuỗi dự đoán có xu hướng phát sinh nhiều thao tác *chèn/xóa/thay thế* ký tự so với nhãn chuẩn. Điều này phản ánh việc mô hình chưa học được cấu trúc chữ viết tay tiếng Việt và chưa phát huy được sức mạnh như kỳ vọng.

TrOCR: **WER ≈ 1.65** và **CER ≈ 1.43**. Đây là mô hình có kết quả tệ nhất.

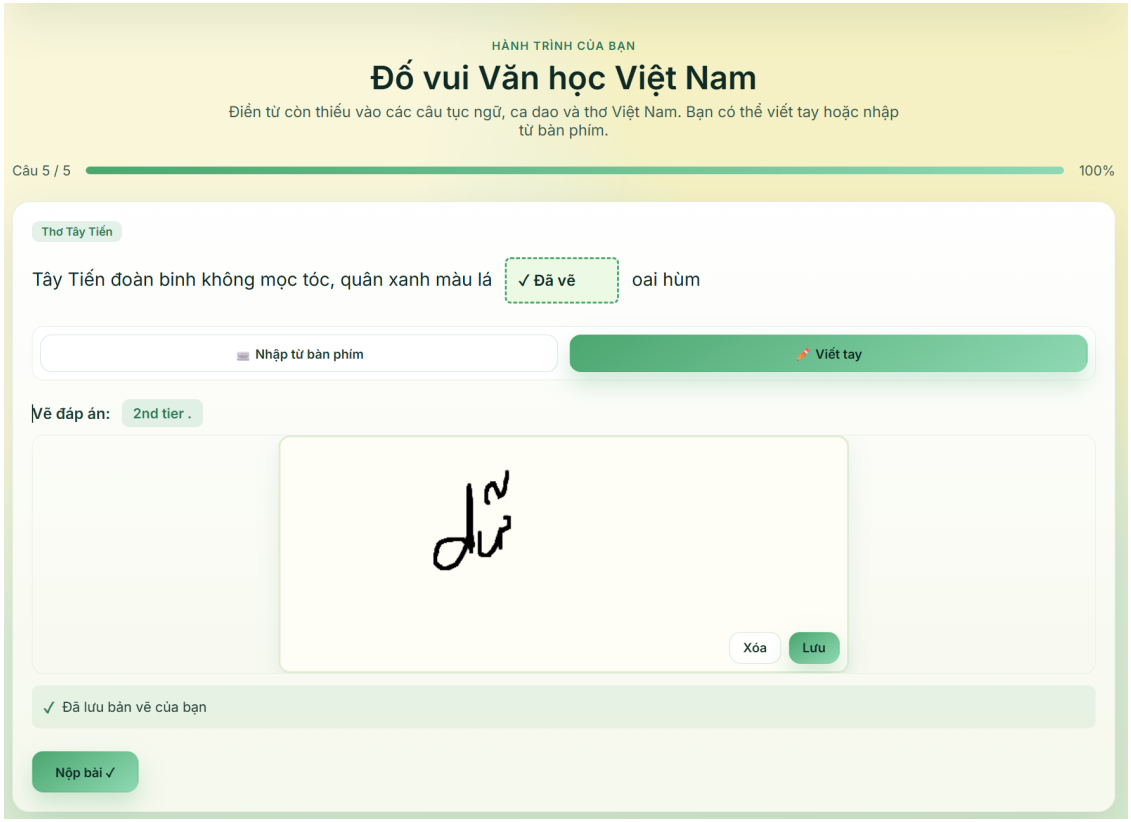
7.5. Ứng dụng

Hệ thống được triển khai trên nền tảng web và tích hợp mô-đun nhận diện chữ viết tay (thông qua mô hình đã được fine-tune). Nhờ đó, người học tiếng Việt có thể viết tay đáp án trực tiếp trên trình duyệt. Kết quả nhận diện sau đó được chuyển đổi thành văn bản và sử dụng để kiểm tra, đánh giá đáp án trong các dạng bài tập như điền từ vào câu, tục ngữ và ca dao.



Hình 7.7: Nhận diện chữ viết tay: "vua nam"

Hình 7.8: Nhận diện chữ viết tay: "sang"



Hình 7.9: Nhận diện chữ viết tay: “dữ”

Hình 7.10: Nhận diện chữ viết tay: "núi cao"

7.6. Thảo luận nhóm

Khi đưa vào thực nghiệm (nhận diện chữ viết tay), mô hình nhận diện tốt chữ không dấu. Tuy nhiên, với chữ có dấu, mô hình fine tuning chưa hoạt động tốt.

Nhìn chung, việc fine tuning chỉ diễn ra một phần của mô hình, và data khoảng 8000 ảnh. Nên hiện mô hình chưa đạt hiệu quả tốt

Bên cạnh đó, việc inference cũng cần một GPU để có thể chạy được. Đây là một hạn chế khá lớn nếu muốn ứng dụng thực tế

CHƯƠNG 8

KẾT LUẬN

8.1. Thảo luận

8.1.1. Nhận xét về mô hình

Điểm mạnh

- **Ứng dụng kiến trúc DeepEncoder hiệu quả:** Mô hình có khả năng xử lý đầu vào với nhiều mức độ phân giải khác nhau, trong khi vẫn duy trì mức tiêu thụ bộ nhớ kích hoạt thấp và số lượng *vision tokens* ở mức tối thiểu.
- **Khả năng đa tác vụ trong môi trường đa phương thức:** Mô hình xử lý tốt nhiều dạng dữ liệu như biểu đồ, hình học, công thức hóa học, tài liệu tự nhiên và ảnh cảnh vật, qua đó mở rộng đáng kể phạm vi ứng dụng trong thực tế.
- **Hiệu quả trên tập dữ liệu lớn:** Mô hình cho thấy hiệu năng tốt khi được huấn luyện và đánh giá trên các tập dữ liệu quy mô lớn bằng tiếng Anh và tiếng Trung.

Điểm yếu

- **Giới hạn về tỉ lệ nén:** Theo kết quả đánh giá, độ chính xác của mô hình suy giảm đáng kể khi tỉ lệ nén vượt quá 10×, gây hạn chế trong các ứng dụng yêu cầu dung lượng biểu diễn cực thấp.
- **Thách thức trong triển khai thời gian thực:** Mặc dù chi phí suy luận đã được tối ưu (chỉ sử dụng khoảng 570 triệu tham số trong quá trình inference), việc triển khai trong các hệ thống thời gian thực vẫn gặp nhiều khó khăn, đặc biệt trong các môi trường yêu cầu độ trễ thấp và tốc độ xử lý cao.
- **Hiệu quả kém trong điều kiện dữ liệu hạn chế:** Mô hình không phù hợp với các bài toán có tập dữ liệu nhỏ, dẫn đến khả năng dự đoán kém. Do đó, việc tinh chỉnh mô hình cần được thực hiện cẩn thận để đạt hiệu quả như mô tả trong bài báo gốc.

- **Chi phí tinh chỉnh cao:** Khi tinh chỉnh hơn 70 triệu tham số, thời gian huấn luyện kéo dài trên 5 giờ, cho thấy việc mở rộng mô hình sang các nhiệm vụ khác là một thách thức lớn về chi phí tính toán.

8.1.2. Quan điểm của nhóm về bài báo, ứng dụng và hướng phát triển tương lai

- Nhóm đánh giá phương pháp *Optical Compression* là một ý tưởng tiềm năng và khả thi cho bài toán xử lý ngữ cảnh dài. Mô hình có thể đạt tỉ lệ nén lên tới 10× trong khi vẫn duy trì độ chính xác OCR khoảng 97%, cho thấy giá trị ứng dụng thực tế rõ rệt.
- Về mặt định hướng nghiên cứu, phương pháp này mở ra hướng phát triển cho các kiến trúc xử lý ngữ cảnh dài dựa trên cơ chế *"nén"* các đoạn ngữ cảnh cũ thành ảnh có độ phân giải thấp nhằm tiết kiệm bộ nhớ và tài nguyên tính toán, đồng thời vẫn bảo toàn thông tin cốt lõi.
- Cơ chế giảm dần độ phân giải để thực hiện nén gợi liên tưởng đến cơ chế *"lãng quên"* của con người, đây là một hướng nghiên cứu đáng chú ý trong việc phát triển các hệ thống học sâu có bộ nhớ ngữ cảnh động.
- Mặc dù kiến trúc hiện tại đạt kết quả tốt trên tiếng Trung và tiếng Anh, khi áp dụng cho tiếng Việt - đặc biệt là chữ viết tay có dấu - mô hình gặp nhiều khó khăn trong quá trình học. Nguyên nhân một phần đến từ độ phức tạp của hệ thống ký tự tiếng Việt và giới hạn tài nguyên khiến nhóm chỉ có thể thử nghiệm trên tập dữ liệu nhỏ. Trong tương lai, việc mở rộng nghiên cứu với các tập dữ liệu tiếng Việt lớn hơn là hướng đi cần thiết, đặc biệt trong các lĩnh vực chuyên biệt như nhận dạng chữ viết tay trên toa thuốc trong ngành y tế.
- Một hạn chế khác của mô hình là tốc độ suy luận còn chậm, gây trở ngại cho việc triển khai trong thực tế. Do đó, cần tiếp tục nghiên cứu các phương pháp rút gọn mô hình nhằm giảm chi phí tính toán và đáp ứng tốt hơn các yêu cầu của ứng dụng thực tiễn.

8.2. Kết luận

Nhóm đánh giá DeepSeek-OCR là một hướng nghiên cứu đầy triển vọng cho bài toán OCR 2.0 [1]. Việc kết hợp giữa kiến trúc tiên tiến của DeepSeek và dữ liệu thực nghiệm đặc thù của nhóm mang lại cái nhìn sâu sắc và đa chiều hơn về khả năng ứng dụng thực tiễn của các mô hình thị giác–ngôn ngữ lớn (Vision-Language Models – VLMs) tại Việt Nam.

Tuy nhiên, để mô hình đạt được hiệu quả cao trong các bài toán thực tế tại Việt Nam, cần có các tập dữ liệu đủ lớn và đa dạng. Bên cạnh đó, các thử nghiệm quy mô lớn hơn là cần thiết nhằm đánh giá một cách toàn diện và chính xác hơn năng lực của mô hình trên dữ liệu tiếng Việt.

TÀI LIỆU THAM KHẢO

[1] Hao Wei, Yifan Sun, and Yifan Li. "DeepSeek-OCR: Contexts Optical Compression". In: *CoRR* abs/2510.18234 (2025). arXiv: 2510.18234. URL: https://arxiv.org/abs/2510.18234.

[2] Chao Liu et al. "Focus Anywhere for Fine-grained Multi-page Document Understanding". In: *CoRR* abs/2405.14295 (2024). arXiv: 2405.14295. URL: https://arxiv.org/abs/2405.14295.

[3] Aobo Liu et al. "DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model". In: *CoRR* abs/2405.04434 (2024). arXiv: 2405.04434. URL: https://arxiv.org/abs/2405.04434.

[4] Aobo Liu et al. "DeepSeek-V3 Technical Report". In: *CoRR* abs/2412.19437 (2024). arXiv: 2412.19437. URL: https://arxiv.org/abs/2412.19437.

[5] Alec Radford et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *CoRR* abs/2106.09685 (2021). arXiv: 2106.09685. URL: https://arxiv.org/abs/2106.09685.

[6] High-Flyer. *HAI-LLM: Efficient and Lightweight Training Tool for Large Models*. 2023. URL: https://www.high-flyer.cn/en/blog/hai-llm.

[7] Hao Wei et al. "General OCR Theory: Towards OCR-2.0 via a Unified End-to-End Model". In: *CoRR* abs/2409.01704 (2024). arXiv: 2409.01704. URL: https://arxiv.org/abs/2409.01704.

[8] Alexander Kirillov et al. "Segment Anything". In: *CoRR* abs/2304.02643 (2023). arXiv: 2304.02643. URL: https://arxiv.org/abs/2304.02643.

[9] Alec Radford et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *CoRR* abs/2103.00020 (2021). arXiv: 2103.00020. URL: https://arxiv.org/abs/2103.00020.