NGUYỄN ĐỨC TRUNG

HỆ THỐNG THÔNG TIN

BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**NGUYỄN ĐỨC TRUNG**

# PHÂN TÍCH VÀ CẢI THIỆN HIỆU SUẤT CỦA THUẬT TOÁN NHẬN DIỆN CHỮ VIẾT TAY DỰA TRÊN MÔ HÌNH NGÔN NGỮ BERT

Hà Nội – 2025

BỘ CÔNG THƯƠNG
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**


**NGUYỄN ĐỨC TRUNG**

# PHÂN TÍCH VÀ CẢI THIỆN HIỆU SUẤT CỦA THUẬT TOÁN NHẬN DIỆN CHỮ VIẾT TAY DỰA TRÊN MÔ HÌNH NGÔN NGỮ BERT

Ngành: Hệ thống thông tin
Mã số: 8480104

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ HỆ THỐNG THÔNG TIN

NGƯỜI HƯỚNG DẪN:
TS. NGUYỄN THỊ DIỆU LINH

Hà Nội – 2025

# LỜI CAM ĐOAN

Tôi xin cam đoan đây là đề tài nghiên cứu của riêng tôi. Các số liệu kết quả nêu trong luận văn là trung thực, các nội dung tham khảo đều được trích dẫn rõ ràng, đầy đủ.

*Hà Nội, ngày  tháng  năm 2025*

Học viên thực hiện

# LỜI CẢM ƠN

Em xin chân thành cảm ơn  Trường Đại học Công nghiệp Hà Nội đã tạo điều kiện cho em được học tập trong suốt thời gian qua. Em cũng xin bày tỏ lòng biết ơn sâu sắc đến tập thể giáo viên Trường Công nghệ Thông tin và Truyền thông đã nhiệt tình dạy dỗ em trong suốt thời gian học tập tại trường, động viên và cổ vũ em trong suốt quá trình nghiên cứu, thực hiện đề án.

Em xin trân trọng cảm ơn TS.Nguyễn Thị Diệu Linh đã tận tình hướng dẫn để em có thể hoàn thiện đề án tốt nghiệp này

Đề án này không chỉ giúp em nâng cao hiểu biết và kỹ năng nghiên cứu mà còn giúp em có cơ hội thực hành và áp dụng các kiến thức đã học vào thực tế. Em tin rằng những kết quả và kinh nghiệm thu được từ đề án sẽ có giá trị thực tiễn cao và có thể áp dụng được trong công việc của em trong tương lai.

Một lần nữa, em xin chân thành cảm ơn các thầy, cô đã giúp đỡ em trong quá trình nghiên cứu và thực hiện đề án này.

Trân trọng!

Học viên thực hiện

# MỤC LỤC

# DANH MỤC CÁC KÝ HIỆU, TỪ VIẾT TẮT

| Ký hiệu | Thuật ngữ tiếng Anh | Thuật ngữ tiếng Việt |
|---|---|---|
| HWR | Hand writing recognition | Nhận diện chữ viết tay |
| OCR | Optical Character recognition | Nhận diện ký hiệu quang học |
| BERT | Bidirectional Encoder Representations from Transformers | Mô hình Transformer hai chiều |
| WER | Word Error Rate | Tỷ lệ nhận sai chữ |
| CER | Character Error Rate | Tỷ lệ nhận sai ký tự |
| CNN | Convolution Neural Network | Mạng nơ ron tích chập |
| LSTM | Long Short-term Memory | Mạng trí nhớ ngắn hạn định hướng dài hạn |
| RNN | Recurrent Neural Network | Mạng nơ ron hồi tiếp |
| TrOCR | Transformers OCR | Nhận diện chữ viết tay dựa trên mô hình Transformers |
| BiLSTM | Bidirectional Long Short-term Memory | Mạng trí nhớ ngắn hạn định hướng dài hạn hai chiều |
| NLP | Natural Language Processing | Xử lý ngôn ngữ tự nhiên. |
| HMM | Hidden Markov Model | Mô hình Markov ẩn |
| SVM | Support Vector Machine | Máy Vector hỗ trợ |

# DANH MỤC CÁC BẢNG

# DANH MỤC HÌNH ẢNH

# LỜI NÓI ĐẦU

Nhận diện chữ viết tay (Handwriting Recognition - HWR) là một trong những bài toán quan trọng trong lĩnh vực thị giác máy tính (Computer Vision) và xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP). Công nghệ này có ứng dụng rộng rãi trong nhiều lĩnh vực như: xử lý tài liệu, xác minh chữ ký, tổ chức thông tin và nhiều ứng dụng khác.

Ngoài ra, trong y tế, nhận diện chữ viết tay của bác sĩ trên đơn thuốc giúp giảm sai sót trong việc kê đơn. Theo báo cáo của MarketsandMarkets (2023), thị trường công nghệ nhận diện văn bản dự kiến đạt giá trị 25 tỷ USD vào năm 2028, phản ánh nhu cầu ngày càng tăng.

Tuy nhiên, nhận diện chữ viết tay đối mặt với nhiều thách thức lớn. Sự đa dạng trong phong cách viết của mỗi cá nhân, từ nét chữ nghiêng, chữ thảo, đến các biến thể ngôn ngữ, khiến việc xây dựng một mô hình chung trở nên phức tạp. Bên cạnh đó, các yếu tố như chất lượng giấy (ố vàng, rách), độ nhiễu từ môi trường (vết bẩn, ánh sáng), và sự khác biệt giữa các ngôn ngữ (ví dụ: chữ Latin so với chữ tượng hình) càng làm tăng độ khó. Trong bối cảnh tiếng Việt, với hệ thống dấu thanh và ký tự đặc trưng, vấn đề càng trở nên đặc thù, đòi hỏi các phương pháp chuyên biệt.

Trong những năm gần đây, sự phát triển mạnh mẽ của Deep Learning đã giúp cải thiện đáng kể hiệu suất của các hệ thống nhận diện chữ viết tay. Đặc biệt, mô hình BERT (Bidirectional Encoder Representations from Transformers) đã chứng minh hiệu quả vượt trội trong xử lý ngôn ngữ tự nhiên (NLP) và có tiềm năng lớn trong việc sửa lỗi nhận diện chữ viết tay.

Chính vì những lý do trên, đề tài "Phân tích và cải thiện hiệu suất của thuật toán nhận diện chữ viết tay dựa trên mô hình ngôn ngữ BERT" được lựa chọn nhằm nghiên cứu và tối ưu hóa quá trình nhận diện chữ viết tay, giúp nâng cao độ chính xác và khả năng ứng dụng trong thực tế.

Nội dung đề án được thực hiện thành 3 chương.

Chương 1: Tổng quan về nhận diện chữ viết tay

Chương 2: Tối ưu thuật toán nhận diện chữ viết tay dựa trên mô hình ngôn ngữ BERT

Chương 3: Triển khai thực nghiệm.

# CHƯƠNG 1: TỔNG QUAN VỀ NHẬN DIỆN CHỮ VIẾT TAY

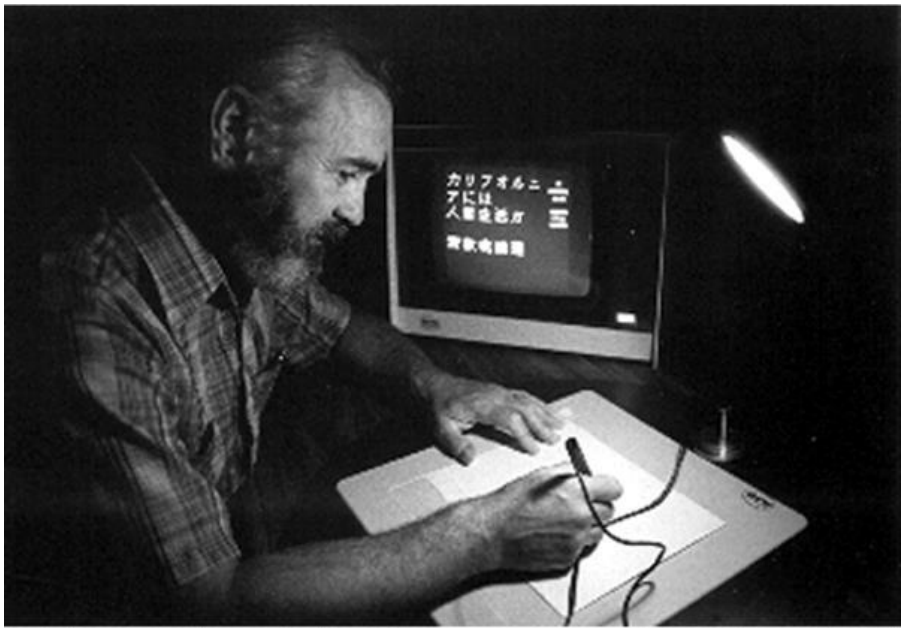
## 1.1. TỔNG QUAN VỀ NHẬN DIỆN CHỮ VIẾT TAY

### 1.1.1. Lịch sử hình thành nhận diện chữ viết tay

Nhận diện chữ viết tay (Handwriting Recognition - HWR) là một lĩnh vực nghiên cứu quan trọng trong trí tuệ nhân tạo (AI) và xử lý hình ảnh, tập trung vào việc phát triển các hệ thống tự động chuyển đổi văn bản viết tay thành dạng văn bản số hóa có thể đọc và xử lý bởi máy tính. Công nghệ này đóng vai trò quan trọng trong việc số hóa tài liệu, cải thiện khả năng tiếp cận thông tin, và hỗ trợ các ứng dụng thực tế trong nhiều lĩnh vực như giáo dục, y tế, ngân hàng, và bảo tồn văn hóa.

HWR không chỉ đơn thuần là nhận diện ký tự mà còn bao gồm việc xử lý các đặc điểm phức tạp của chữ viết tay, như phong cách viết cá nhân, sự không đồng đều trong nét chữ, và ngữ cảnh ngôn ngữ. Với sự phát triển của học máy (machine learning) và học sâu (deep learning), HWR đã đạt được những bước tiến đáng kể, nhưng vẫn đối mặt với nhiều thách thức do tính đa dạng và phức tạp của chữ viết tay con người.

Nhận diện chữ viết tay có lịch sử phát triển lâu dài, bắt đầu từ những năm 1950, khi các nhà nghiên cứu bắt đầu thử nghiệm với các bút điện tử và logic nhận diện ký tự cho các máy tính số mới. Một trong những bước ngoặt đầu tiên là công trình của Hew Crane tại SRI International, người đã phát triển các phương pháp nhận diện chữ viết tay thời gian thực. Công việc này đã đặt nền móng cho các ứng dụng như xác thực chữ ký, nhập liệu trực tiếp các ký tự tượng hình (như tiếng Trung, Nhật, Hàn), và các hệ thống tính toán dựa trên bút, như được mô tả trong tài liệu từ [1].



Hình 1.1: Thiết bị bút điện tử của Hew Crane tại SRI International, một trong những công cụ đầu tiên cho nhận diện chữ viết tay thời gian thực.

Đến những năm 1960, Crane đã đăng ký bằng sáng chế cho một thiết bị bút đặc biệt, và vào đầu những năm 1970, hệ thống của ông đã sử dụng "bút SRI" để nhập các ký tự vào máy tính. Những nỗ lực ban đầu này đã phát triển thành các hệ thống nhận diện chữ viết tay hiện đại ngày nay. Đến những năm 1990, các hệ thống như Apple Newton đã được giới thiệu, mặc dù độ chính xác của chúng chưa đạt được sự hoàn hảo, như được thảo luận trong [2]. Tuy nhiên, các tiến bộ trong học máy và học sâu đã thúc đẩy sự phát triển của công nghệ này, dẫn đến các hệ thống hiện đại với độ chính xác cao hơn, như các công cụ OCR tiên tiến và mô hình ngôn ngữ như BERT.

### 1.1.2. Phân loại nhận diện chữ viết tay:

Nhận diện chữ viết tay được chia thành hai loại chính dựa trên nguồn dữ liệu đầu vào và cách thức thu thập, mỗi loại có ưu điểm và hạn chế riêng:

- **Nhận diện chữ viết tay trực tuyến (Online Handwriting Recognition):**

+ Đặc điểm: Dữ liệu được thu thập trực tiếp trong thời gian thực thông qua các thiết bị cảm ứng như bút stylus (ví dụ: Apple Pencil, Samsung S Pen), bảng vẽ kỹ thuật số (như Wacom), hoặc màn hình cảm ứng của điện thoại và máy tính bảng. Dữ liệu trực tuyến không chỉ bao gồm hình ảnh của nét chữ mà còn ghi lại các thông tin động như thứ tự nét bút, áp lực viết, tốc độ viết, và hướng di chuyển của bút.
+ Ưu điểm: Những thông tin động cung cấp ngữ cảnh phong phú, giúp hệ thống dễ dàng phân tích các đặc điểm của chữ viết tay, từ đó cải thiện độ chính xác của quá trình nhận diện.
+ Ứng dụng: Các ứng dụng điển hình bao gồm ghi chú số trên thiết bị di động, nhập liệu văn bản trên màn hình cảm ứng, hoặc các giao diện điều khiển dựa trên cử chỉ viết tay, đặc biệt hữu ích trong môi trường làm việc hoặc học tập hiện đại.
+ Hạn chế: Yêu cầu thiết bị chuyên dụng và thường bị giới hạn trong các môi trường mà người dùng có thể tương tác trực tiếp với công nghệ cảm ứng, không phù hợp cho việc xử lý tài liệu đã viết sẵn.

- **Nhận diện chữ viết tay ngoại tuyến (Offline Handwriting Recognition):**

Xử lý hình ảnh của chữ viết tay từ tài liệu quét hoặc ảnh chụp.

+ Đặc điểm: Phương pháp này tập trung vào việc xử lý các hình ảnh tĩnh của chữ viết tay, chẳng hạn như tài liệu được quét, ảnh chụp từ máy ảnh, hoặc các bản thảo viết tay trên giấy. Không giống như nhận diện trực tuyến, nhận diện ngoại tuyến không có thông tin về quá trình viết (như thứ tự nét bút hay áp lực), mà chỉ dựa vào đặc điểm hình ảnh của văn bản, chẳng hạn như hình dạng ký tự, khoảng cách giữa các chữ, và bố cục tổng thể.

+ Ưu điểm: Có phạm vi ứng dụng rộng hơn, đặc biệt trong các lĩnh vực như số hóa tài liệu lịch sử, xử lý biểu mẫu hành chính, hoặc phân tích ghi chú viết tay trong y tế và giáo dục, vì không yêu cầu thiết bị cảm ứng và phù hợp với tài liệu đã có sẵn.
+ Hạn chế: Là một bài toán phức tạp hơn do thiếu thông tin động, hệ thống phải đối mặt với các thách thức như chất lượng hình ảnh kém, chữ viết tay không rõ ràng, hoặc sự khác biệt trong phong cách viết của từng người.

Nội dung đề án này tập trung vào nhận diện chữ viết tay ngoại tuyến**.**

## 1.2. Tình hình nghiên cứu trong nước

Trong bối cảnh Việt Nam đang đẩy mạnh quá trình chuyển đổi số quốc gia, công nghệ nhận diện chữ viết tay (Handwritten Text Recognition - HTR) nổi lên như một yếu tố then chốt, có vai trò chiến lược trong việc số hóa và khai thác khối lượng lớn dữ liệu giấy tờ hiện có. Chính phủ Việt Nam đã thể hiện cam kết mạnh mẽ đối với sự phát triển của trí tuệ nhân tạo (AI) thông qua Chiến lược quốc gia về AI đến năm 2030, đặt mục tiêu đưa Việt Nam trở thành trung tâm AI của ASEAN và góp mặt trên bản đồ AI toàn cầu. HTR, một nhánh quan trọng của AI, đang được nghiên cứu và ứng dụng sâu rộng, từ các cơ sở học thuật đến các doanh nghiệp công nghệ hàng đầu trong nước. Mặc dù đã đạt được những thành tựu đáng kể, đặc biệt là việc phát triển các giải pháp nội địa với độ chính xác cao cho tiếng Việt, lĩnh vực này vẫn đối mặt với nhiều thách thức đặc thù do tính phức tạp của ngôn ngữ và sự đa dạng của nét chữ viết tay. Việc tiếp tục đầu tư vào nghiên cứu cơ bản, phát triển bộ dữ liệu chất lượng cao và thúc đẩy hợp tác giữa các bên liên quan là cần thiết để HTR có thể phát huy tối đa tiềm năng, đóng góp vào hiệu quả quản lý hành

chính, tối ưu hóa quy trình doanh nghiệp và nâng cao chất lượng dịch vụ công.

Các công trình nghiên cứu và sản phẩm thương mại tiêu biểu về nhận dạng chữ viết tay tiếng Việt:

- Đại học Bách khoa TP.HCM: Một nhóm nghiên cứu tại Khoa Khoa học và Kỹ thuật Máy tính, Đại học Bách khoa TP.HCM, đã trình bày kỹ thuật nhận dạng ký tự viết tay dựa trên thông tin tĩnh. Phương pháp này bao gồm làm mỏng nét ký tự, trích chọn đặc trưng theo chiều và kết hợp các véctơ đặc điểm cục bộ với thông tin cấu trúc toàn cục, sử dụng mạng nơ-ron để huấn luyện và phân loại. Công trình này đã đạt độ chính xác trên 84% trong các thí nghiệm thực tế.

- Đại học RMIT: Sinh viên Phùng Minh Tuấn của Đại học RMIT đã phát triển thành công một hệ thống đầu cuối để nhận diện và giải mã chữ viết tay của bác sĩ Việt Nam trên bản quét bệnh án. Công nghệ này sử dụng cấu trúc học máy bao gồm mạng nơ-ron ResNet để chiết xuất hình dạng chữ, BiLSTM để lên mẫu tần suất chữ, và CTC cho nhiệm vụ sao chép cuối cùng. Nghiên cứu này đã giải quyết một thách thức lớn trong số hóa bệnh án tiếng Việt, vốn phức tạp bởi các lớp ký tự, âm điệu và dấu câu.

- Viettel AI: Trung tâm Dịch vụ dữ liệu và trí tuệ nhân tạo Viettel (Viettel AI) đã phát triển công nghệ trích xuất thông tin từ ảnh văn bản với độ chính xác cao. Công nghệ này đạt 99% với chữ in và 90% với chữ viết tay tiếng Việt. Điều đáng chú ý là khả năng nhận dạng và xử lý chính xác các văn bản ảnh có cấu trúc phức tạp, bao gồm bảng biểu không phân tách, định dạng phi chuẩn và đặc biệt là chữ viết tay tiếng

Việt, nơi các hệ thống quốc tế thường gặp khó khăn. Viettel AI lựa chọn hướng đi khác biệt khi xây dựng một nền tảng công nghệ có kiến trúc mở, cho phép lắp ghép linh hoạt các module để tùy biến theo nhu cầu của từng bài toán cụ thể. Giải pháp tiêu biểu của họ là Viettel IPA, một công cụ tự động hóa quy trình đang được triển khai rộng rãi trong nhiều lĩnh vực như bảo hiểm, ngân hàng, hành chính - văn phòng. Viettel IPA cho phép nhận diện chữ viết và tự động trích xuất thông tin từ các loại giấy tờ tùy thân, hóa đơn, hợp đồng, giúp tự động phân loại và phê duyệt hồ sơ, tối ưu quy trình xử lý giấy tờ. Sản phẩm này được ghi nhận có khả năng số hóa hàng triệu hồ sơ và tiết kiệm đến 80% thời gian và công sức cho các tổ chức và doanh nghiệp. Công trình của Viettel AI đã nhận được nhiều giải thưởng uy tín trong nước và quốc tế như VIFOTEC 2024.

- FPT.AI (FPT AI Read): FPT AI Read là một giải pháp nhận dạng và trích xuất chữ viết tay từ hình ảnh, được tối ưu hóa đặc biệt cho tiếng Việt. Giải pháp này tích hợp các công nghệ Machine Learning, Deep Learning và Computer Vision để nhận dạng ký tự số và chữ viết tay của con người. FPT AI Read đạt độ chính xác 80-85% cho chữ viết tay và trên 95% cho chữ in. Một điểm mạnh của FPT AI Read là mô hình xác minh thông tin, có khả năng kiểm tra tính đúng đắn của thông tin trích xuất, nhận diện các bất thường (như thiếu hoặc thừa ký tự, ngày tháng sai) và thậm chí đối chiếu thông tin giữa nhiều tài liệu hoặc với hệ thống CRM của doanh nghiệp. Giải pháp này được ứng dụng rộng rãi trong các ngành như bảo hiểm (xử lý bệnh án), tài chính-ngân hàng (ủy nhiệm chi, tài liệu khách hàng), và y tế (hồ sơ bệnh nhân). Việc tự động hóa nhập liệu bằng FPT AI Read được cho là nhanh hơn 40-50 lần so

với nhập liệu thủ công, giảm thiểu lỗi, tăng cường bảo mật thông tin và tiết kiệm đáng kể chi phí cũng như không gian lưu trữ.

## 1.3. CÁC PHƯƠNG PHÁP NHẬN DIỆN CHỮ VIẾT TAY

### 1.3.1. Phương pháp truyền thống

Các phương pháp truyền thống trong nhận diện chữ viết tay dựa trên các kỹ thuật xử lý ảnh kết hợp với các thuật toán học máy như phân loại, mô hình xác suất, hoặc so sánh mẫu. Những phương pháp này từng đóng vai trò quan trọng trong giai đoạn đầu của HWR, trước khi các mô hình học sâu như mạng nơ-ron tích chập (CNN) hay Transformer trở nên thống trị. Dưới đây là phân tích chi tiết về ba kỹ thuật truyền thống chính: Optical Character Recognition (OCR), Hidden Markov Model (HMM), và Support Vector Machine (SVM).

**Nhận diện ký tự quang học (Optical Character Recognition -OCR)**

- Nhận diện ký tự quang học (OCR) là một trong những kỹ thuật tiên phong được sử dụng để nhận diện văn bản, cả in lẫn viết tay. OCR truyền thống hoạt động bằng cách phân tích hình ảnh đầu vào, trích xuất các đặc trưng hình học hoặc so sánh với các mẫu ký tự có sẵn trong cơ sở dữ liệu. Các công cụ OCR phổ biến như Tesseract OCR và Google Vision API đã được phát triển để xử lý chữ viết tay, nhưng chúng thường được tối ưu hóa cho văn bản in hơn [3]. Theo nghiên cứu của Smith (2007), Tesseract OCR là một công cụ mã nguồn mở với khả năng tùy chỉnh cao, nhưng hiệu suất của nó giảm khi gặp chữ viết tay tự nhiên do sự đa dạng trong phong cách viết.


Hình 1.2: Nhận diện ký tự quang học ứng dụng trong đọc biển báo thực tế.

- Cách hoạt động**:** Quá trình OCR truyền thống bao gồm các bước chính sau:

+ Tiền xử lý: Chuẩn hóa hình ảnh bằng cách loại bỏ nhiễu, điều chỉnh độ tương phản, nhị phân hóa (binarization), và sửa lệch (skew correction) để làm nổi bật văn bản [4].
+ Phân đoạn: Chia hình ảnh thành các vùng chứa ký tự, từ, hoặc dòng văn bản riêng lẻ.
+ Trích xuất đặc trưng: Xác định các đặc trưng như hình dạng ký tự, tỷ lệ nét chữ, hoặc góc nghiêng.
+ Phân loại: Sử dụng thuật toán so sánh mẫu (template matching) hoặc phân loại để nhận diện ký tự.

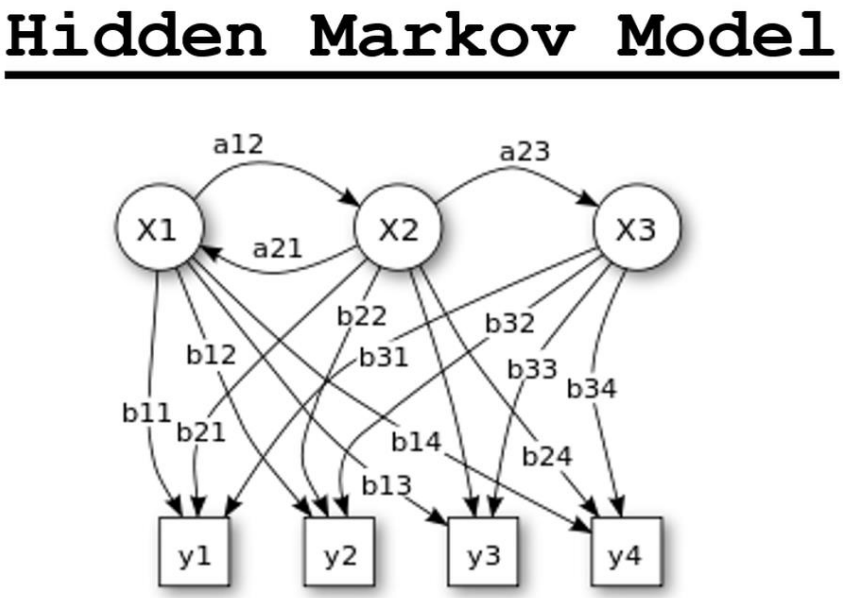
+ Hậu xử lý: Áp dụng từ điển ngôn ngữ hoặc các quy tắc ngữ pháp để sửa lỗi nhận diện [5].

- Ưu điểm:
  + Dễ triển khai trên các hệ thống có tài nguyên hạn chế, phù hợp với các thiết bị không yêu cầu sức mạnh tính toán cao [6].
  + Hiệu quả khi xử lý văn bản in hoặc chữ viết tay rõ ràng, đồng đều, chẳng hạn như biểu mẫu hành chính.
  + Các công cụ như Tesseract OCR cung cấp khả năng tùy chỉnh linh hoạt, cho phép tích hợp với các ứng dụng cụ thể [3].
- Hạn chế:
  + Độ chính xác giảm đáng kể khi xử lý chữ viết tay không rõ ràng, nguệch ngoạc, hoặc có phong cách cá nhân hóa cao [7].
  + Khó xử lý các ngôn ngữ phức tạp như tiếng Việt, nơi dấu thanh (sắc, huyền, hỏi, ngã, nặng) và từ ghép đòi hỏi hiểu ngữ cảnh để nhận diện chính xác.
  + Yêu cầu cơ sở dữ liệu mẫu ký tự đa dạng, nhưng việc xây dựng cơ sở dữ liệu chữ viết tay chất lượng cao là một quá trình tốn kém và phức tạp [8].
- Ứng dụng: Tesseract OCR được sử dụng rộng rãi trong số hóa tài liệu hành chính, thư viện, và các ứng dụng giáo dục [3]. Google Vision API, với khả năng tích hợp trên nền tảng đám mây, hỗ trợ các ứng dụng thương mại như nhận diện chữ viết tay trên thiết bị di động, nhưng vẫn gặp khó khăn với các văn bản viết tay tiếng Việt do thiếu dữ liệu huấn luyện chuyên biệt.

**Mô hình Markov ẩn (Hidden Markov Model - HMM)**

-Mô hình Markov ẩn (HMM) là một mô hình xác suất được sử dụng để nhận diện chuỗi ký tự trong văn bản viết tay, đặc biệt phù hợp với các nhiệm vụ liên quan đến dữ liệu có tính chuỗi như văn bản hoặc lời nói. HMM giả định

rằng chữ viết tay là một chuỗi các trạng thái ẩn (hidden states), mỗi trạng thái đại diện cho một ký tự hoặc một phần của từ, và các quan sát là các đặc trưng hình ảnh thu thập được từ hình ảnh đầu vào [9].



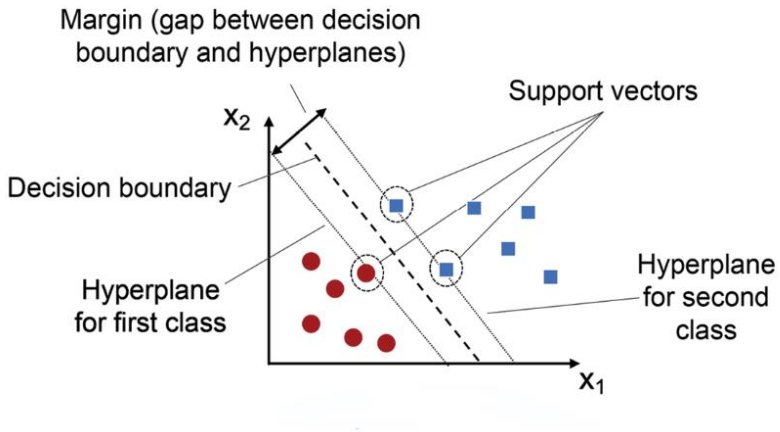

Hình 1.3: Minh họa Mô hình Markov ẩn

- Cách hoạt động:
  + Mô hình hóa: HMM bao gồm một tập hợp trạng thái ẩn và các xác suất chuyển đổi giữa chúng. Mỗi trạng thái phát ra một quan sát dựa trên đặc trưng hình ảnh, chẳng hạn như góc nét chữ hoặc tỷ lệ ký tự [10].
  + Huấn luyện: Sử dụng thuật toán Baum-Welch để ước tính các tham số mô hình (xác suất chuyển đổi và phát ra) từ dữ liệu huấn luyện.
  + Nhận diện: Áp dụng thuật toán Viterbi để tìm chuỗi trạng thái có xác suất cao nhất, tương ứng với văn bản đầu ra [5].
  + Hậu xử lý: Kết hợp với từ điển hoặc mô hình ngôn ngữ để sửa lỗi và cải thiện độ chính xác.

- Ưu điểm:
    - Hiệu quả trong việc mô hình hóa các chuỗi dài, chẳng hạn như câu hoặc đoạn văn, nhờ vào khả năng xử lý tính chuỗi của dữ liệu [11].
    - Có thể tích hợp với các từ điển ngôn ngữ hoặc quy tắc ngữ pháp để tăng cường độ chính xác trong giai đoạn hậu xử lý.
    - Linh hoạt trong việc xử lý các ngôn ngữ có cấu trúc phức tạp, như tiếng Ả Rập hoặc tiếng Trung [12].
- Hạn chế:
    + Yêu cầu trích xuất đặc trưng thủ công, làm tăng độ phức tạp và giảm khả năng tổng quát hóa của mô hình [4].
    + Hiệu suất giảm khi xử lý chữ viết tay có phong cách không đồng nhất hoặc chứa nhiều lỗi hình ảnh, chẳng hạn như chữ viết nguệch ngoạc hoặc hình ảnh có nhiễu.
    + Đối với tiếng Việt, HMM gặp khó khăn trong việc nhận diện dấu thanh nếu không có tập dữ liệu huấn luyện đủ lớn và đa dạng.
- Ứng dụng: HMM từng là phương pháp chủ đạo trong nhận diện chữ viết tay trực tuyến và ngoại tuyến trước khi học sâu trở nên phổ biến. Các hệ thống lai như HMM-GMM (Gaussian Mixture Model) hoặc HMM-CNN đã được phát triển để cải thiện hiệu suất, đặc biệt trong nhận diện chữ viết tay tiếng Ả Rập và tiếng Trung [12]. Trong bối cảnh tiếng Việt, HMM đã được thử nghiệm nhưng chưa đạt hiệu quả cao do thiếu dữ liệu huấn luyện chuyên biệt.

**Máy vector hỗ trợ (Support Vector Machine -SVM)**

Máy vector hỗ trợ (SVM) là một thuật toán học máy được sử dụng để phân loại ký tự dựa trên các đặc trưng hình dạng hoặc hình học. SVM tìm một siêu phẳng tối ưu trong không gian đặc trưng để phân tách các lớp ký tự khác nhau, với mục tiêu tối đa hóa khoảng cách giữa các lớp (margin) [13].




Hình 1.4: Minh họa mô hình Support Vector Machine

Cách hoạt động:

- Trích xuất đặc trưng: Các đặc trưng như tỷ lệ nét chữ, góc nghiêng, hoặc cấu trúc hình học được trích xuất từ hình ảnh ký tự bằng các kỹ thuật như PCA (Principal Component Analysis) hoặc HOG (Histogram of Oriented Gradients) [14].
- Huấn luyện: SVM học cách phân loại các ký tự bằng cách tối ưu hóa siêu phẳng dựa trên dữ liệu huấn luyện, thường sử dụng các kernel như tuyến tính (linear) hoặc RBF (Radial Basis Function).
- Nhận diện: Sử dụng mô hình đã huấn luyện để phân loại các ký tự mới dựa trên đặc trưng của chúng [15].
- Ưu điểm:
  + Hiệu quả trong việc phân loại dữ liệu tuyến tính và phi tuyến tính, đặc biệt khi sử dụng các kernel phi tuyến như RBF [13].
  + Có thể hoạt động tốt với các tập dữ liệu nhỏ hơn so với các mô hình học sâu, phù hợp với các ứng dụng có tài nguyên hạn chế.
  + Dễ triển khai và điều chỉnh, đặc biệt trong các nhiệm vụ nhận diện ký tự đơn giản [11].

- Hạn chế:
  - Yêu cầu trích xuất đặc trưng thủ công, đòi hỏi kiến thức chuyên môn và làm tăng độ phức tạp của quá trình thiết kế hệ thống [11].
  - Hiệu suất giảm khi xử lý các biến thể lớn trong chữ viết tay hoặc khi tập dữ liệu không đủ đa dạng để đại diện cho các phong cách viết khác nhau.
  - Không tận dụng được ngữ cảnh của văn bản, dẫn đến khó khăn trong việc nhận diện chuỗi ký tự hoặc từ, đặc biệt với các ngôn ngữ như tiếng Việt có cấu trúc từ ghép phức tạp.

SVM thường được sử dụng trong các hệ thống nhận diện chữ viết tay đơn giản, chẳng hạn như nhận diện chữ số viết tay (như tập dữ liệu MNIST) hoặc các ký tự riêng lẻ trong các ứng dụng như nhận diện mã bưu điện [16]. Trong bối cảnh chữ viết tay tiếng Việt, SVM đã được thử nghiệm nhưng không đạt hiệu suất cao do sự phức tạp của dấu thanh và từ ghép, như được ghi nhận trong nghiên cứu của Nguyen và Le (2020).

**Quy trình nhận diện chữ viết tay truyền thống**

Theo nghiên cứu của Impedovo và Pirlo (2014), quy trình nhận diện chữ viết tay truyền thống thường bao gồm các bước sau [4]:

- Kỹ thuật số hóa hình ảnh: Chuyển đổi tài liệu vật lý thành hình ảnh kỹ thuật số bằng máy quét hoặc máy ảnh.
- Tiền xử lý: Áp dụng các kỹ thuật như nhị phân hóa, loại bỏ nhiễu, phát hiện cạnh, sửa lệch, và chuẩn hóa hình ảnh để cải thiện chất lượng đầu vào.
- Phân đoạn: Chia hình ảnh thành các vùng chứa ký tự, từ, hoặc dòng văn bản riêng lẻ, sử dụng các thuật toán như phân tích bố cục hoặc phát hiện khoảng trắng.

- Trích xuất đặc trưng: Sử dụng các kỹ thuật như PCA, LDA (Linear Discriminant Analysis), hoặc HOG để trích xuất các đặc trưng quan trọng từ các vùng đã phân đoạn [14].
- Phân loại: Phân loại các đặc trưng thành các ký tự hoặc từ bằng các phương pháp như so sánh mẫu, SVM, hoặc HMM.
- Hậu xử lý: Áp dụng các từ điển ngôn ngữ, quy tắc ngữ pháp, hoặc mô hình xác suất để sửa lỗi và cải thiện độ chính xác [17].

Quy trình này tuy đơn giản và dễ triển khai, nhưng gặp nhiều hạn chế khi xử lý chữ viết tay tự nhiên do sự phụ thuộc vào các bước thủ công và thiếu khả năng học hỏi từ dữ liệu phức tạp.

- So sánh hiệu suất và khoảng trống nghiên cứu:

Bảng so sánh hiệu suất:

Dựa trên các nghiên cứu gần đây, dưới đây là bảng so sánh hiệu suất của các mô hình HWR, bao gồm cả phương pháp truyền thống và hiện đại, trên các tập dữ liệu phổ biến [7]:

Bảng 1.1: Bảng so sánh hiệu suất các phương pháp truyền thống:

| Mô hình | Tập dữ liệu | CER (%) | WER (%) | Thời gian xử lý (s/mẫu) |
|---|---|---|---|---|
| Tesseract OCR | IAM | 15.3 | 28.6 | **0.2** |
| HMM | CASIA-HWDB | 12.8 | 25.4 | 0.4 |
| SVM | RIMES | 14.2 | 27.8 | 0.3 |

CER (Character Error Rate): Đo lường tỷ lệ lỗi ký tự, tính bằng số ký tự sai (thêm, bớt, thay thế) chia cho tổng số ký tự.

WER (Word Error Rate): Đo lường tỷ lệ lỗi từ, tương tự CER nhưng áp dụng ở cấp độ từ.

Thời gian xử lý: Thời gian trung bình để xử lý một mẫu hình ảnh, phản ánh hiệu suất tính toán của mô hình.

Bảng trên cho thấy các phương pháp truyền thống như Tesseract OCR, HMM, và SVM có hiệu suất thấp đáng kể.

Các phương pháp truyền thống gặp phải nhiều hạn chế, như được ghi nhận trong các nghiên cứu:

- Độ chính xác thấp: Do phụ thuộc vào trích xuất đặc trưng thủ công, các phương pháp này không thể xử lý hiệu quả các biến thể lớn trong chữ viết tay, như chữ viết nguệch ngoạc hoặc phong cách cá nhân.
- Khó khăn với ngôn ngữ phức tạp: Các ngôn ngữ như tiếng Việt, với dấu thanh và từ ghép, yêu cầu mô hình hiểu ngữ cảnh, điều mà các phương pháp truyền thống không làm tốt.

- Yêu cầu dữ liệu lớn: Để đạt độ chính xác cao, cần có cơ sở dữ liệu mẫu đa dạng, nhưng việc thu thập dữ liệu chữ viết tay chất lượng cao là tốn kém và mất thời gian.
- Khả năng tổng quát hóa kém: Các phương pháp truyền thống thường được tối ưu hóa cho các tập dữ liệu cụ thể, dẫn đến hiệu suất giảm khi áp dụng cho các ngữ cảnh mới hoặc ngôn ngữ khác.

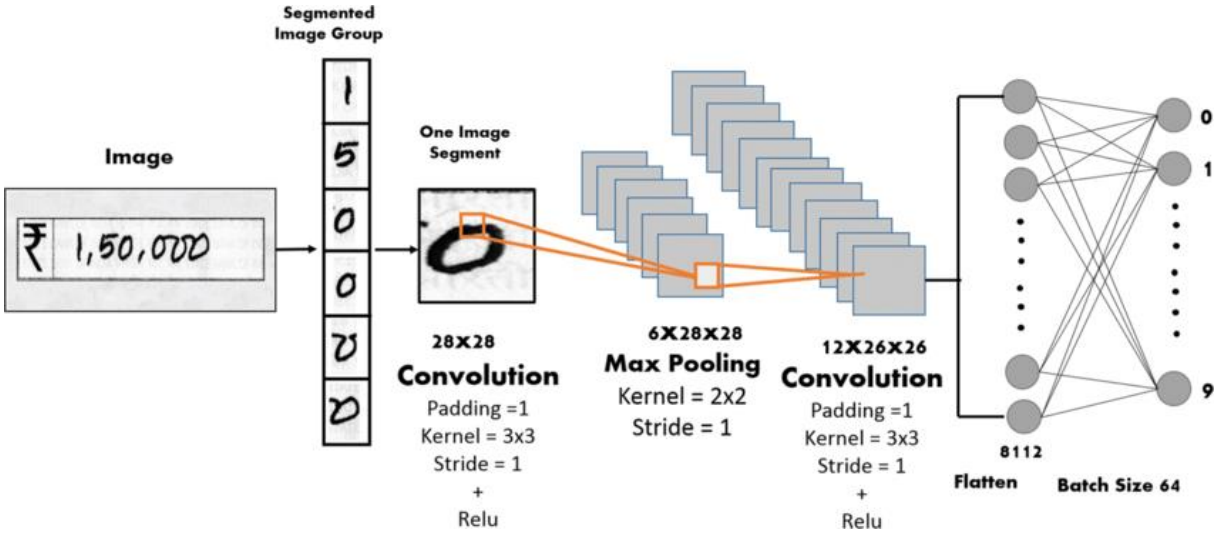
### 1.3.2. Phương pháp hiện đại dựa trên Deep Learning

Nhờ sự phát triển vượt bậc của học sâu, các mô hình nhận diện chữ viết tay hiện đại đã đạt được những bước tiến đáng kể về độ chính xác và khả năng xử lý các loại chữ viết tay phức tạp, từ chữ viết tay rời rạc đến chữ viết tay nối liền. Các phương pháp này tận dụng các kiến trúc mạng nơ-ron tiên tiến để tự động học các đặc trưng từ dữ liệu, vượt xa các kỹ thuật truyền thống dựa trên trích xuất đặc trưng thủ công. Trong phần này, ba kỹ thuật học sâu chính được phân tích chi tiết: mạng nơ ron tích chập, mạng nơ ron hồi tiếp/ mạng bộ nhớ ngắn dài hạn, và nhận diện ký tự dựa trên Transformers. Mỗi kỹ thuật được trình bày với mô tả, cách hoạt động, ưu điểm, hạn chế, và ứng dụng thực tiễn, dựa trên các nghiên cứu khoa học uy tín.

#### 1.3.2.1. Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) là một trong những mô hình học sâu phổ biến nhất trong nhận diện chữ viết tay, đặc biệt trong các hệ thống nhận diện ngoại tuyến, nơi đầu vào là hình ảnh tĩnh. CNN hoạt động bằng cách trích xuất tự động các đặc trưng không gian từ hình ảnh, chẳng hạn như hình dạng ký tự, tỷ lệ nét chữ, hoặc bố cục văn bản. Theo nghiên cứu của Memon và cộng sự (2020), CNN đã chứng minh hiệu quả cao trong việc nhận diện chữ viết tay nhờ khả năng học các đặc trưng cấp cao mà không cần can thiệp thủ công [7].

Ví dụ, trong nhận diện chữ số viết tay trên tập dữ liệu MNIST, CNN đạt độ chính xác vượt trội so với các phương pháp truyền thống như SVM [14].




Hình 1.5: Minh họa kiến trúc CNN trích xuất đặc trưng từ hình ảnh chữ viết tay.

Quá trình nhận diện chữ viết tay bằng CNN bao gồm các bước chính sau:

- Tiền xử lý hình ảnh: Chuẩn hóa hình ảnh đầu vào bằng cách điều chỉnh kích thước, loại bỏ nhiễu, và nhị phân hóa để làm nổi bật văn bản [4].
- Tích chập (Convolution): Áp dụng các bộ lọc (filters) để trích xuất các đặc trưng như cạnh, góc, hoặc hình dạng ký tự từ hình ảnh.
- Gộp (Pooling): Giảm kích thước không gian của đặc trưng để tăng hiệu quả tính toán và giảm nguy cơ quá khớp (overfitting).
- Phân loại: Sử dụng các lớp kết nối đầy đủ (fully connected layers) hoặc các hàm kích hoạt như softmax để phân loại các đặc trưng thành các ký tự hoặc từ [18].
- Hậu xử lý: Áp dụng các kỹ thuật như từ điển ngôn ngữ để sửa lỗi nhận diện [5].
- Ưu điểm:

+ Tự động học các đặc trưng từ dữ liệu, loại bỏ nhu cầu trích xuất đặc trưng thủ công vốn phức tạp và tốn thời gian [7].
+ Hiệu quả cao trong nhận diện chữ viết tay rõ ràng, đặc biệt là chữ số hoặc ký tự riêng lẻ, như được chứng minh trên tập dữ liệu MNIST với độ chính xác lên đến 99% [14].
+ Dễ dàng mở rộng và tích hợp với các mô hình khác, chẳng hạn như RNN hoặc Transformer, để xử lý các nhiệm vụ phức tạp hơn [19].

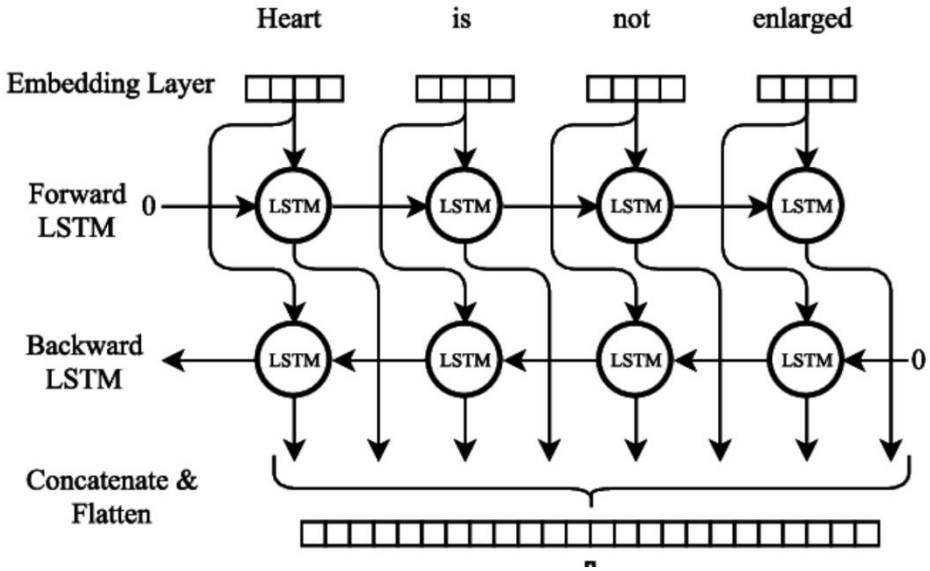
- Hạn chế:
  + Hiệu suất giảm khi xử lý chữ viết tay nối liền hoặc có phong cách viết không đồng nhất, do CNN tập trung vào đặc trưng không gian mà thiếu khả năng mô hình hóa chuỗi.
  + Yêu cầu tập dữ liệu huấn luyện lớn và đa dạng để đạt hiệu quả cao, đặc biệt với các ngôn ngữ có hệ thống chữ viết phức tạp như tiếng Việt.
  + Tốn tài nguyên tính toán khi xử lý hình ảnh độ phân giải cao hoặc văn bản dài, đòi hỏi phần cứng mạnh mẽ.

CNN được sử dụng rộng rãi trong các hệ thống nhận diện chữ số viết tay (như mã bưu điện), nhận diện ký tự riêng lẻ, và các ứng dụng OCR thương mại như Google Vision API. Nghiên cứu của Kavitha và Srimathi (2020) đã áp dụng CNN để nhận diện chữ viết tay tiếng Tamil, đạt độ chính xác 95.16% trên tập dữ liệu tùy chỉnh. Trong bối cảnh tiếng Việt, CNN cũng được thử nghiệm nhưng thường kết hợp với các mô hình khác để xử lý dấu thanh và từ ghép.

#### 1.3.2.2. Recurrent Neural Networks (RNN) / Long Short-Term Memory (LSTM)

Recurrent Neural Networks (RNN) và các biến thể như Long Short-Term Memory (LSTM) được thiết kế để xử lý dữ liệu chuỗi, rất phù hợp cho nhận diện chữ viết tay liên tục, chẳng hạn như từ hoặc câu viết tay nối liền.

RNN/LSTM có khả năng ghi nhớ thông tin từ các bước trước để dự đoán các ký tự tiếp theo, giúp mô hình hóa mối quan hệ ngữ cảnh trong văn bản. Theo nghiên cứu của Graves và Schmidhuber (2009), LSTM đã đạt CER 6.8% trên tập dữ liệu IAM, chứng minh hiệu quả vượt trội trong nhận diện văn bản viết tay tiếng Anh [19]. BiLSTM, một biến thể của LSTM, còn cải thiện hiệu suất bằng cách xử lý chuỗi theo cả hai chiều (tiến và lùi) [20].




Hình 1.6: Minh họa kiến trúc BiLSTM xử lý chuỗi ký tự trong nhận diện chữ viết tay.

Quá trình nhận diện chữ viết tay bằng RNN/LSTM bao gồm các bước chính sau:

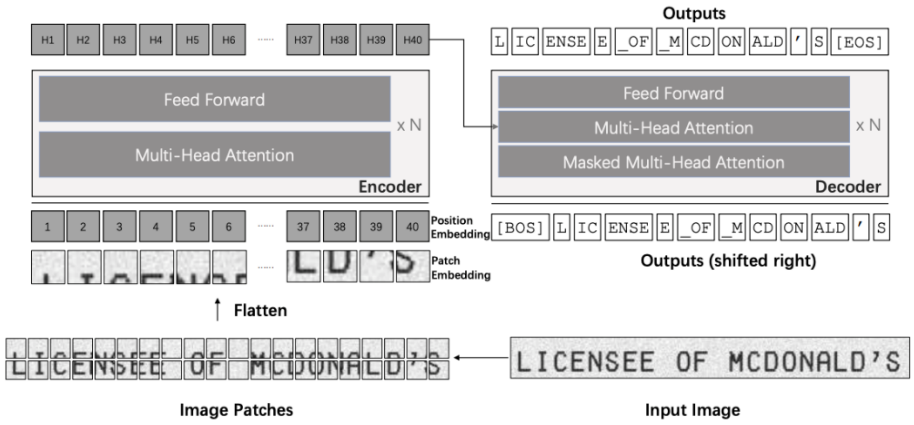
- Tiền xử lý: Chuẩn hóa hình ảnh hoặc dữ liệu điểm (trong nhận diện trực tuyến) để tạo đầu vào phù hợp [4].
- Trích xuất đặc trưng: Sử dụng CNN hoặc các kỹ thuật khác để tạo biểu diễn đặc trưng từ hình ảnh hoặc chuỗi điểm, làm đầu vào cho RNN/LSTM.
- Mô hình hóa chuỗi: RNN/LSTM xử lý chuỗi đặc trưng theo thời gian, ghi nhớ thông tin ngữ cảnh để dự đoán các ký tự hoặc từ.

- Giải mã: Áp dụng các kỹ thuật như Connectionist Temporal Classification (CTC) để đồng bộ hóa đầu vào và đầu ra mà không cần phân đoạn trước.
- Hậu xử lý: Sử dụng từ điển hoặc mô hình ngôn ngữ để sửa lỗi và tối ưu hóa văn bản đầu ra [5].
- Ưu điểm:
  + Hiệu quả trong mô hình hóa chuỗi dài, đặc biệt là văn bản viết tay nối liền, nhờ khả năng xử lý tính chuỗi và ngữ cảnh.
  + Có thể tích hợp với CNN để tạo thành mô hình lai (CRNN), cải thiện hiệu suất trên các tập dữ liệu lớn như IAM và RIMES.
  + Phù hợp cho cả nhận diện trực tuyến và ngoại tuyến, đặc biệt với các ngôn ngữ có cấu trúc phức tạp như tiếng Ả Rập.
- Hạn chế:
  + Yêu cầu thời gian huấn luyện lâu và tài nguyên tính toán lớn, đặc biệt với các mô hình BiLSTM trên tập dữ liệu lớn.
  + Hiệu suất giảm khi xử lý chữ viết tay có nhiễu cao hoặc phong cách viết không đồng nhất, đòi hỏi dữ liệu huấn luyện đa dạng.
  + Khó xử lý các ngôn ngữ như tiếng Việt nếu thiếu dữ liệu huấn luyện chuyên biệt để học dấu thanh và từ ghép.

RNN/LSTM được sử dụng rộng rãi trong nhận diện chữ viết tay trực tuyến (như Google Handwriting Input) và ngoại tuyến (như số hóa tài liệu). Nghiên cứu của Sharma và Singh (2024) đã áp dụng mô hình lai CNN-BiLSTM để nhận diện văn bản viết tay tiếng Anh, đạt độ chính xác 98.50% trên tập dữ liệu IAM. Trong bối cảnh tiếng Việt, RNN/LSTM đã được thử nghiệm nhưng cần cải thiện để xử lý dấu thanh hiệu quả hơn.

### 1.3.2.3. Transformer-based OCR (TrOCR)

Transformer-based OCR (TrOCR) là một phương pháp tiên tiến sử dụng kiến trúc Transformer, vốn nổi tiếng trong xử lý ngôn ngữ tự nhiên (NLP), để nhận diện chữ viết tay. TrOCR kết hợp các mô hình Transformer mã hóa-giải mã (encoder-decoder) với các kỹ thuật học sâu để nhận diện văn bản trực tiếp từ hình ảnh, vượt trội hơn CNN và RNN trong một số trường hợp nhờ cơ chế chú ý (attention mechanism). Theo nghiên cứu của Li và cộng sự (2021) [21], TrOCR đạt CER 5.2% trên tập dữ liệu IAM, cải thiện đáng kể so với các mô hình CNN+RNN+CTC [22]. TrOCR đặc biệt phù hợp cho các nhiệm vụ yêu cầu hiểu ngữ cảnh sâu, chẳng hạn như nhận diện văn bản viết tay phức tạp hoặc đa ngôn ngữ.




Hình 1.7: Minh họa kiến trúc TrOCR dựa trên Transformer nhận diện chữ viết tay từ hình ảnh.

Quá trình nhận diện chữ viết tay bằng TrOCR bao gồm các bước chính sau:

- Tiền xử lý hình ảnh: Chuẩn hóa hình ảnh đầu vào để đảm bảo chất lượng tốt, loại bỏ nhiễu và điều chỉnh kích thước [4].
- Mã hóa (Encoding): Sử dụng Transformer encoder để trích xuất đặc trưng từ hình ảnh, tận dụng cơ chế chú ý để tập trung vào các vùng chứa văn bản [21].

- Giải mã (Decoding): Transformer decoder dự đoán chuỗi ký tự hoặc từ dựa trên đặc trưng mã hóa, kết hợp với cơ chế chú ý để mô hình hóa ngữ cảnh [18].
- Hậu xử lý: Áp dụng mô hình ngôn ngữ hoặc từ điển để sửa lỗi và tối ưu hóa văn bản đầu ra [18].
- Ưu điểm:
  + Hiệu quả cao trong nhận diện văn bản phức tạp nhờ cơ chế chú ý, cho phép mô hình bắt được các phụ thuộc dài hạn trong dữ liệu.
  + Tốc độ xử lý nhanh hơn so với RNN/LSTM, đặc biệt khi triển khai trên phần cứng hiện đại như GPU.
  + Có khả năng tích hợp với các mô hình ngôn ngữ như BERT để cải thiện hậu xử lý, đặc biệt với các ngôn ngữ cần hiểu ngữ cảnh như tiếng Việt [23].
- Hạn chế:
  + Yêu cầu tài nguyên tính toán lớn, đặc biệt khi huấn luyện trên tập dữ liệu lớn hoặc triển khai trên thiết bị di động.
  + Hiệu suất phụ thuộc vào chất lượng và quy mô dữ liệu huấn luyện, gây khó khăn cho các ngôn ngữ ít tài nguyên như tiếng Việt.
  + Khó tối ưu hóa cho các ứng dụng thời gian thực do độ phức tạp tính toán của kiến trúc Transformer.

TrOCR được sử dụng trong các hệ thống OCR tiên tiến, như nhận diện văn bản viết tay trong tài liệu lịch sử hoặc ứng dụng thương mại trên nền tảng đám mây. Nghiên cứu của Li và cộng sự (2021) đã áp dụng TrOCR để nhận diện văn bản tiếng Anh và tiếng Trung, đạt hiệu suất cao trên các tập dữ liệu IAM và CASIA-HWDB [23].

### 1.3.2.4. Bảng so sánh hiệu suất:

Dựa trên các nghiên cứu gần đây, dưới đây là bảng so sánh hiệu suất của các mô hình HWR hiện đại trên các tập dữ liệu phổ biến:

Bảng 1.2: Bảng so sánh hiệu suất các mô hình HWR hiện đại

| **Mô hình** | **Bộ dữ liệu** | **CER** | **WER** | **Thời gian xử lý** |
|---|---|---|---|---|
| CNN+RNN+CTC | IC03 | 6.8 | 14.5 | 0.3 |
| TrOCR base (Transformer) | IAM | **3.42** | **11.8** | 0.5 |
| CNN+BERT | CASIA-HWDB | 7.5 | 15.2 | 0.9 |
| CNN+ BiLSTM | RIMES | 4.8 | 10.2 | 0.4 |

Bảng trên cho thấy các mô hình học sâu như TrOCR và CNN+BiLSTM vượt trội hơn so với CNN đơn lẻ, nhưng thời gian xử lý của TrOCR và CNN+BERT vẫn chưa được tối ưu hóa hoàn toàn, đặc biệt với thời gian xử lý cao (0.9s/mẫu), đặt ra thách thức trong các ứng dụng thời gian thực

Mặc dù các mô hình học sâu như CNN+RNN+CTC và TrOCR đã cải thiện đáng kể hiệu suất của HWR, vẫn còn nhiều khoảng trống nghiên cứu cần được giải quyết:

- Hiệu suất thời gian thực: Các mô hình như CNN+BERT có thời gian xử lý cao (0.9s/mẫu), không phù hợp với các ứng dụng yêu cầu xử lý thời gian thực, như nhập liệu trực tiếp trên thiết bị di động.
- Tích hợp ngữ cảnh ngôn ngữ: Các phương pháp truyền thống và một số mô hình học sâu không tận dụng tốt ngữ cảnh ngôn ngữ, dẫn đến lỗi chính tả khi nhận diện các từ đồng âm hoặc từ ghép.

- Tối ưu hóa tài nguyên: Các mô hình học sâu như BERT yêu cầu tài nguyên tính toán lớn, gây khó khăn khi triển khai trên các thiết bị di động hoặc máy tính cá nhân có cấu hình thấp.
- Đề án hướng đến việc lấp đầy các khoảng trống trên bằng cách:
- Tích hợp BERT như một lớp hậu xử lý để sửa lỗi và tối ưu hóa văn bản đầu ra từ OCR.
- Tinh chỉnh BERT trên dữ liệu thu thập được để xử lý một cách chính xác hơn.
- Tối ưu hóa hiệu suất tính toán của BERT để cải thiện khả năng xử lý thời gian thực, hướng đến các ứng dụng thực tế.

## 1.4. . KẾT LUẬN CHƯƠNG 1

Chương này đã cung cấp một cái nhìn toàn diện về đề tài nghiên cứu, bao gồm tổng quan về công nghệ nhận diện chữ viết tay, các ứng dụng thực tế, tiềm năng phát triển, và những thách thức còn tồn tại. Đồng thời, chương cũng làm rõ các mục tiêu cụ thể của đề án, bao gồm phân tích hiệu suất hiện tại, ứng dụng mô hình BERT, thử nghiệm trên dữ liệu thực tế, và đề xuất hướng phát triển. Phương pháp nghiên cứu được trình bày chi tiết, từ tổng hợp tài liệu, xây dựng hệ thống, đến huấn luyện và tối ưu hóa mô hình, đảm bảo rằng quá trình thực hiện được tiến hành một cách khoa học và có hệ thống. Phạm vi nghiên cứu cũng được xác định rõ ràng để phù hợp với các giới hạn về thời gian và tài nguyên, đồng thời tạo nền tảng cho các mở rộng trong tương lai.

Trong các chương tiếp theo, đề án sẽ đi sâu vào các khía cạnh kỹ thuật, bao gồm cơ sở lý thuyết về nhận diện chữ viết tay và mô hình BERT, quy trình triển khai, và các kết quả thử nghiệm. Những nội dung này sẽ cung cấp cái nhìn rõ nét hơn về cách công nghệ nhận diện chữ viết tay có thể được cải thiện thông

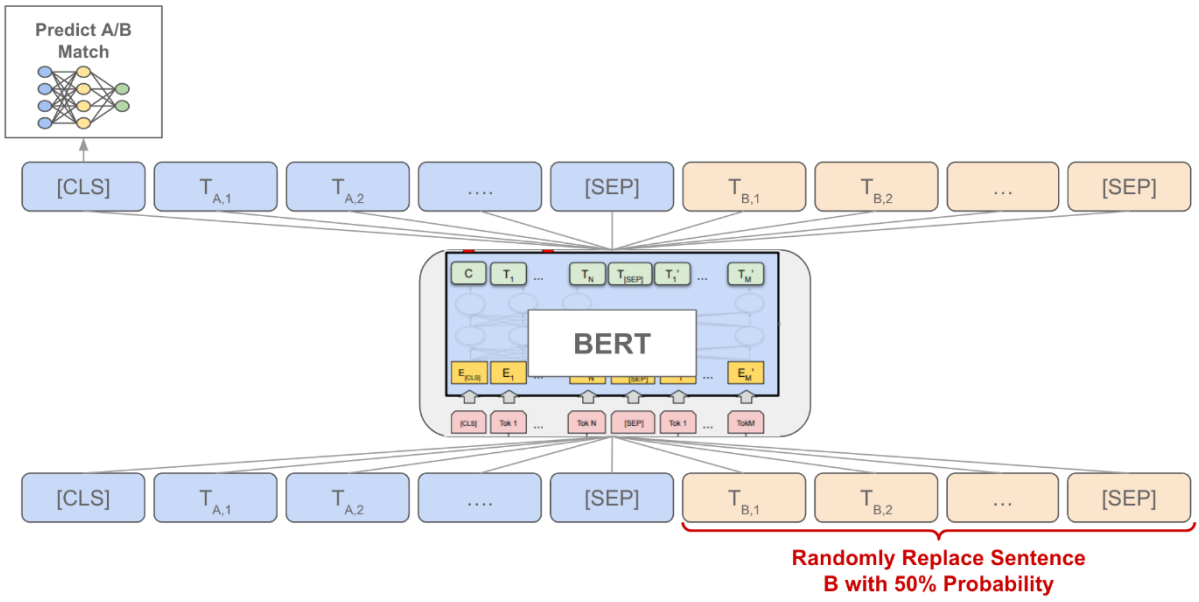
qua việc ứng dụng các mô hình ngôn ngữ tiên tiến, đồng thời đóng góp vào sự phát triển của lĩnh vực này trong bối cảnh chuyển đổi số ngày càng sâu rộng.

# CHƯƠNG 2: TỐI ƯU THUẬT TOÁN NHẬN DIỆN CHỮ VIẾT TAY DỰA TRÊN MÔ HÌNH NGÔN NGỮ BERT

## 2.1. MÔ HÌNH NGÔN NGỮ BERT

### 2.1.1. Giới thiệu về BERT

BERT, viết tắt của Bidirectional Encoder Representations from Transformers, là một mô hình ngôn ngữ tiên tiến được Google giới thiệu vào năm 2018, đánh dấu một bước ngoặt trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Theo bài báo gốc của Devlin và cộng sự, BERT được thiết kế để học biểu diễn ngữ cảnh hai chiều từ văn bản không gắn nhãn, cải thiện đáng kể hiệu suất trên nhiều nhiệm vụ NLP như phân loại văn bản, trả lời câu hỏi, và nhận diện thực thể [23].



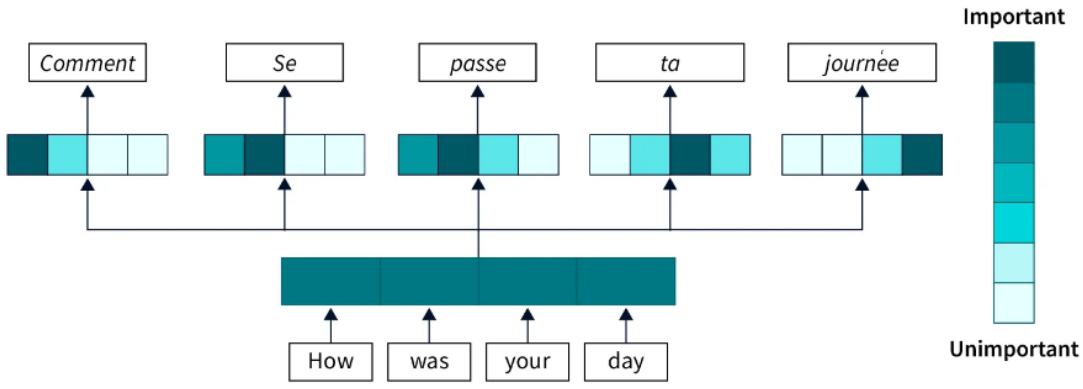

Hình 2.1: Kiến trúc tổng quan BERT

BERT khác biệt so với các mô hình trước đây như LSTM (Long Short-Term Memory) hoặc GRU (Gated Recurrent Unit), vốn chỉ xử lý ngữ cảnh theo một hướng (từ trái sang phải hoặc ngược lại). BERT sử dụng kiến trúc

Transformer với cơ chế chú ý hai chiều (bidirectional attention), cho phép nó hiểu ngữ cảnh của mỗi từ bằng cách xem xét cả phần trước và sau trong câu, như được giải thích trên [24]. Điều này giúp BERT nắm bắt được ý nghĩa chính xác hơn, đặc biệt trong các trường hợp từ đa nghĩa (polysemy), như được thảo luận trên [25].

### 2.1.2. Kiến trúc của BERT

BERT dựa trên kiến trúc Transformer, được giới thiệu trong bài báo "Attention Is All You Need" của Vaswani và cộng sự (2017), sử dụng cơ chế Self-Attention để học mối quan hệ giữa các từ trong câu [26]. Kiến trúc này bao gồm các lớp encoder, mỗi lớp có cơ chế chú ý và mạng nơ-ron feed-forward, BERT sử dụng kiến trúc encoder-only, tập trung vào việc hiểu đầu vào thay vì tạo đầu ra.

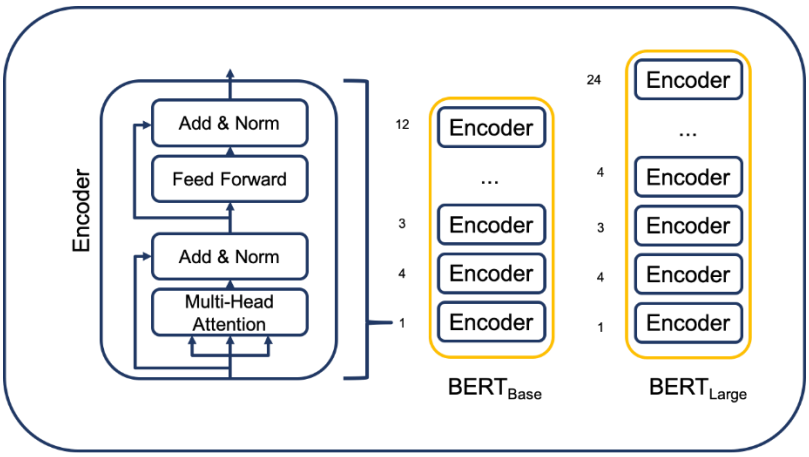



Hình 2.2: Minh họa cơ chế Self-Attention trong Transformer, nền tảng của BERT.

Cấu trúc của BERT có hai phiên bản chính: BERT-base với 12 tầng và BERT-large với 24 tầng, mỗi tầng sử dụng 12 đầu chú ý (attention heads) và kích thước embedding là 768. Phiên bản đa ngôn ngữ (multilingual BERT) hỗ trợ 104 ngôn ngữ, bao gồm tiếng Việt, được huấn luyện trước trên dữ liệu từ

Wikipedia và Common Crawl. Theo tài liệu từ Hugging Face (2023), multilingual BERT đạt độ chính xác khoảng 85% trong các tác vụ phân loại văn bản tiếng Việt, trong khi PhoBERT[27], một biến thể tối ưu hóa cho tiếng Việt, đạt 88%

BERT được ứng dụng rộng rãi, từ cải thiện kết quả tìm kiếm của Google Search nhờ khả năng hiểu ngữ cảnh, đến hỗ trợ các hệ thống chatbot như Grok của xAI, vốn sử dụng các biến thể của Transformer để tạo ra các cuộc đối thoại tự nhiên và mượt mà.

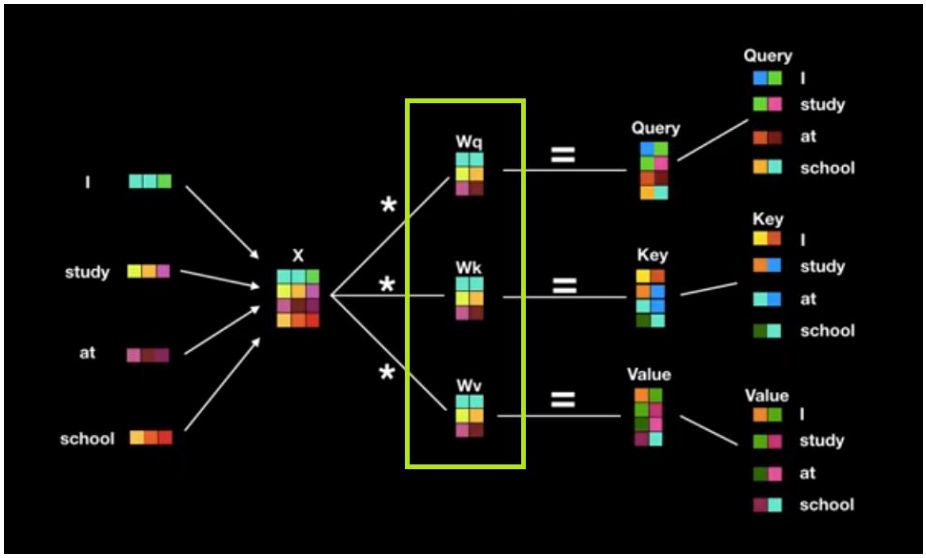



Hình 2.3: Kiến trúc BERT base bao gồm 12 lớp

Mỗi lớp Transformer Encoder bao gồm hai phần chính: Multi-Head Self-Attention và Feedforward Neural Network, kết hợp với Layer Normalization và Residual Connections:

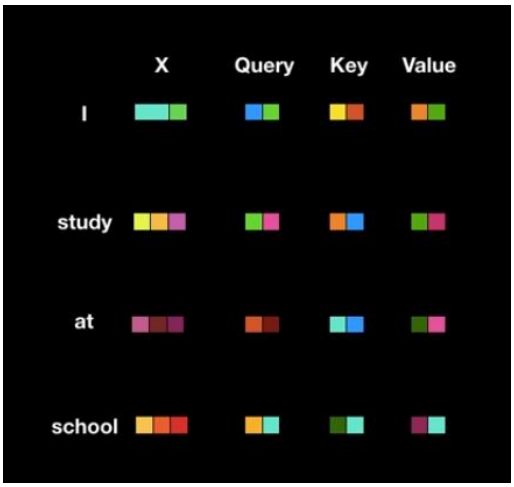
- Từ Self-Attention đến Multi-Head Self-Attention:

Đây chính là cơ chế khi mỗi từ có thể điều chỉnh trọng số của nó cho các từ khác trong câu sao cho từ ở vị trí cảng gần nó nhất thì trọng số càng lớn và càng xa thì càng nhỏ dần. Sau bước nhúng từ (đi qua embeding layer) ta có đầu vào của encoder và decoder là ma trận $X$ có kích thước $m \times n$ với m,n lần lượt là độ dài câu và số chiều của 1 vector nhúng từ.




Hình 2.4: Mô tả cơ tổng quan cơ chế Self-Attention

Trong khung màu vàng ở hình 2.9 là 3 ma trận $W_q, W_k, W_v$ chính là những hệ số mà model cần huấn luyện. Sau khi nhân các ma trận này với ma trận đầu vào $X$ ta thu được ma trận $Q, K, V$ (tương thích với trong hình là ma trận Query, Key, Value). Ma trận Query và Key có tác dụng tính toán ra phân phối score cho các cặp từ. Ma trận Value sẽ dựa trên phân phối score để tính ra véc tơ phân phối xác suất output. Như vậy mỗi một từ sẽ được gán bởi 3 vector query, key và value là các dòng của $Q, K, V$.

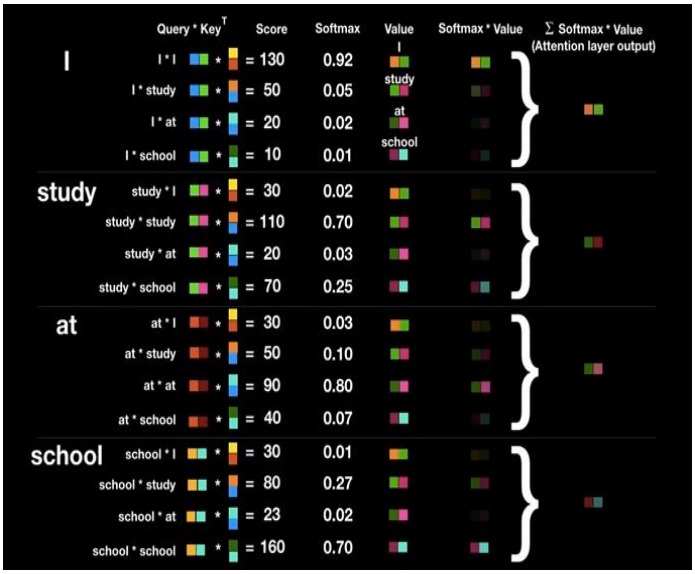



Hình 2.5: Các từ input tương ứng với vector key, query và value.

Để tính ra score cho mỗi cặp từ (w_i,w_j ), dot-product giữa query với key sẽ được tính toán , phép tính này nhằm tìm ra mối liên hệ trọng số của các cặp từ. Tuy nhiên điểm số sau cùng là điểm số chưa được chuẩn hóa. Do đó chuẩn hóa bằng một hàm softmax để đưa về một phân phối xác xuất mà độ lớn sẽ đại diện cho mức độ attention của từ query tới từ key. Trọng số càng lớn càng chứng tỏ từ $w_i$ trả về một sự chú ý lớn hơn đối với từ $w_j$. Sau đó nhân hàm softmax với các vector giá trị của từ hay còn gọi là value vector để tìm ra vector đại diện (attention vetor) sau khi đã học trên toàn bộ câu input.


Hình 2.6: Quá trình tính toán trọng số attention và attention vector cho từ I trong câu I study at school.

Hoàn toàn tương tự khi di chuyển sang các từ khác trong câu ta cũng thu được kết quả tính toán như minh họa.




Hình 2.7: Kết quả tính attention vector cho toàn bộ các từ trong câu.

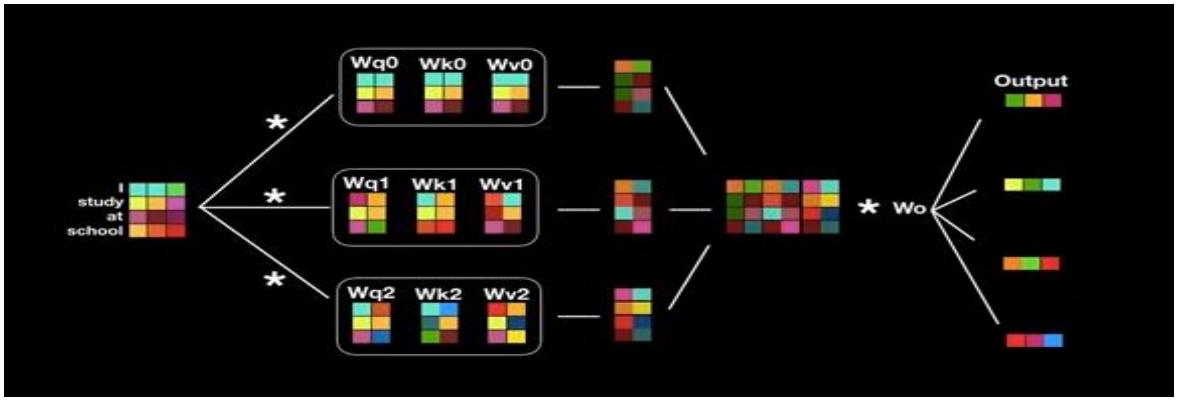
Đầu vào để tính attention sẽ bao gồm ma trận $Q$ (mỗi dòng của nó là một vector query đại diện cho các từ input), ma trận K (tương tự như ma trận Q, mỗi dòng là vector key đại diện cho các từ input). Hai ma trận $Q, K$ được sử dụng để tính attention mà các từ trong câu trả về cho 1 từ cụ thể trong câu. attention vector sẽ được tính dựa trên trung bình có trọng số của các vector value trong ma trận $V$ với trọng số attention (được tính từ $Q, K$).

Trong thực hành ta tính toán hàm attention trên toàn bộ tập các câu truy vấn một cách đồng thời được đóng gói thông qua ma trận $Q$. Keys và Values cũng được đóng gói cùng nhau thông qua ma trận $K$ và $V$. Phương trình Attention như sau:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^{\mathbf{T}}}{\sqrt{d_k}})\mathbf{V} \quad (2.1)$$

Việc chia cho $d_k$ là số dimension của vector key nhằm mục đích tránh tràn luồng nếu số mũ là quá lớn.

Như vậy, sau quá trình Self-Attention, ta sẽ thu được 1 ma trận attention. Các tham số mà model cần tinh chỉnh chính là các ma trận $W_q, W_k, W_v$. Mỗi quá trình như vậy được gọi là 1 head của attention. Khi lặp lại quá trình này nhiều lần (trong bài báo [25] là 3 heads) ta sẽ thu được quá trình Multi-head Attention như hình 2.13.




Hình 2.8: Sơ đồ cấu trúc Multi-head Attention.

Sau khi thu được 3 matrix attention ở đầu ra ta sẽ concatenate các matrix này theo các cột để thu được ma trận tổng hợp multi-head có chiều cao trùng với chiều cao của ma trận input.

$$\text{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{concatenate}(head_1, head_2, ..., head_h)\mathbf{W_0} \quad (2.2)$$

- Feed-Forward Networks

Ngoài các lớp Multi-head Attention, mỗi khối trong mô hình BERT đều chứa một mạng kết nối đầy đủ truyền thẳng, được áp dụng cho mỗi 1 cặp vị trí một cách riêng biệt và đồng nhất. Kiến trúc mạng bao gồm 2 phép biến đổi tuyến tính và 1 hàm kích hoạt GeLU [28] ở giữa. Công thức của mạng này như sau:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (2.3)$$

- Layer Normalization và Residual Connections:

BERT sử dụng residual connections [29] quanh mỗi 2 lớp con (multi-head attention và feed forward neural network), sau đó là lớp layer normalization [30]. Đầu ra của mỗi khối tuân theo công thức:

$$\text{LayerNorm}(x + \text{Sublayer}(x)) \quad (2.4).$$

Trong đó $Sublayer(x)$ là chức năng được triển khai bởi lớp con.

Để tạo điều kiện cho residual connections, mỗi lớp con trong khối cũng như các embedding layer đều có kích thước đầu ra $d_{model}$ cố định, vớ BERT, ta có 2 kích thước đầu ra cố định theo các phiên bản $BERT_{base}$ và $BERT_{large}$ lần lượt là $d_{model} = 768$ và $d_{model} = 1024$.

### 2.1.3. Ứng dụng của BERT trong nhận diện chữ viết tay

BERT không trực tiếp nhận diện chữ viết tay mà hỗ trợ sửa lỗi đầu ra từ OCR bằng cách:

- Phát hiện và sửa lỗi chính tả: Khi OCR nhận diện sai ký tự, BERT có thể dự đoán từ đúng dựa trên ngữ cảnh.

- Cải thiện phân loại văn bản: Nếu cần phân loại nội dung sau khi nhận diện, BERT có thể giúp tăng độ chính xác.

- Trong HWR (Hand Writting Recognition), BERT có thể xử lý chuỗi ký tự sơ bộ từ hình ảnh, sửa lỗi ngữ nghĩa dựa trên ngữ cảnh. Ví dụ, với câu viết tay "Tôi học sinh viên," BERT có thể dự đoán đúng thành "Tôi là sinh viên" thay vì giữ nguyên lỗi. Khả năng này đặc biệt hữu ích với chữ viết tay tiếng Việt, nơi dấu thanh thường bị nhầm (hòa/hòe, nhà/nha). Đây là lý do BERT được chọn làm nền tảng trong nghiên cứu này, với kỳ vọng cải thiện độ chính xác so với các mô hình thuần túy dựa trên hình ảnh.

Nghiên cứu về HWR đã phát triển qua nhiều giai đoạn. Phương pháp cổ điển dựa trên OCR (Tesseract, ABBYY FineReader) chỉ hiệu quả với chữ in, với CER trên chữ viết tay thường vượt quá 20%. Puigcerver (2017) [31] đề xuất mô hình CNN+RNN+CTC, đạt CER 6.8% trên tập IAM (66.000 dòng chữ viết tay tiếng Anh). Gần đây, Transformer thuần (như TrOCR của Li và cộng sự, 2021) [21] cải thiện CER xuống 5.2% nhờ khả năng xử lý song song.

Ứng dụng BERT trong nhận diện văn bản cũng được ghi nhận. Adnan và cộng sự (2020) [32] tích hợp BERT vào OCR tiếng Anh, giảm WER từ 12% xuống 8% trên văn bản in bị nhiễu. Với chữ viết tay, Zhang và cộng sự (2022) [33] kết hợp CNN+BERT trên tập dữ liệu tiếng Trung (CASIA-HWDB), đạt CER 7.5%, cải thiện 1.5% so với CRNN.

## 2.2. PHÂN TÍCH GIỚI HẠN VÀ KHOẢNG TRỐNG CẦN TỐI ƯU CỦA BERT

### 2.2.1. Ngữ cảnh đặc thù của dữ liệu đầu vào trong bài toán HWR

Khác với các tác vụ NLP ( Xử lý ngôn ngữ tự nhiên) thông thường nơi đầu vào là văn bản sạch và chuẩn, trong bài toán nhận diện chữ viết tay, đầu vào của mô hình ngôn ngữ là kết quả từ hệ thống nhận dạng ký tự, vốn thường xuyên mắc lỗi. Những lỗi phổ biến bao gồm:

- Nhầm lẫn ký tự tương đồng (như “l” và “1”, “O” và “0”, “rn” và “m”)
- Thiếu dấu (trong tiếng Việt)
- Bỏ sót hoặc thêm ký tự không cần thiết

Điều này khiến cho ngữ cảnh mà BERT tiếp nhận trở nên nhiễu và không rõ ràng, từ đó làm giảm hiệu quả của việc hiểu ngữ nghĩa hai chiều – một trong những thế mạnh lớn nhất của BERT.

### 2.2.2. Hiệu suất mô hình và chi phí tính toán

- Kích thước mô hình lớn: BERT-base có ~110 triệu tham số, dẫn đến thời gian suy luận lâu và yêu cầu phần cứng cao gây ra bất lợi nếu muốn ứng dụng mô hình trong môi trường thật, như thiết bị nhúng hoặc ứng dụng mobile.

- Thời gian fine-tuning lâu: Việc tinh chỉnh mô hình BERT trên dữ liệu yêu cầu nhiều tài nguyên tính toán và thời gian huấn luyện dài, trong khi hiệu quả cải thiện đôi khi không tương xứng nếu không có chiến lược fine-tune hiệu quả.

## 2.3. CÁC KỸ THUẬT TỐI ƯU HÓA MÔ HÌNH

### 2.3.1. Fine-tuning BERT

Để mô hình BERT có thể hoạt động hiệu quả trong việc sửa lỗi chính tả do nhận dạng quang học (OCR) gây ra, cần tiến hành quá trình fine-tuning —

tức là điều chỉnh mô hình theo bài toán cụ thể. Quá trình này gồm các bước chính như sau:

- Thu thập dữ liệu: Trước tiên, sử dụng ba mô hình OCR phổ biến là VietOCR, Pytesseract và EasyOCR để tạo dữ liệu đầu vào có lỗi từ cùng một tập văn bản gốc. Sau đó, so sánh kết quả nhận dạng với văn bản gốc để đánh giá tỷ lệ lỗi (CER/WER) của từng mô hình.
- Lựa chọn baseline OCR: Dựa trên kết quả so sánh, mô hình nào cho thấy hiệu suất cao hơn hai mô hình còn lại và được chọn làm công cụ tạo dữ liệu lỗi cho quá trình huấn luyện. Tập dữ liệu cuối cùng bao gồm hơn 7000 cặp câu, mỗi cặp gồm văn bản lỗi (OCR output) và văn bản đúng tương ứng (ground truth).
- Huấn luyện mô hình: Mô hình BERT sau đó được fine-tune trên tập dữ liệu này để học cách sửa các lỗi phổ biến do OCR gây ra như thiếu dấu, nhầm ký tự, tách/ghép từ sai, v.v.

### 2.3.2. Sử dụng mô hình nhẹ hơn

Do mô hình BERT gốc có kích thước lớn và tiêu tốn tài nguyên, việc triển khai trên các thiết bị hạn chế hoặc yêu cầu thời gian phản hồi nhanh là một thách thức. Vì vậy, có thể sử dụng các phiên bản BERT đã được rút gọn nhưng vẫn giữ hiệu quả cao:

- DistilBERT: Là một phiên bản nhỏ gọn hơn của BERT gốc, chỉ có khoảng 60% số tham số nhưng vẫn giữ tới 95% độ chính xác, giúp giảm đáng kể chi phí tính toán.

## 2.4. KẾT LUẬN CHƯƠNG 2

Chương này đã trình bày những kiến thức nền tảng quan trọng liên quan đến bài toán nhận diện chữ viết tay và cách ứng dụng mô hình BERT để cải thiện hiệu suất. Các nội dung chính bao gồm:

- Nhận diện chữ viết tay: So sánh phương pháp truyền thống và Deep Learning.
- Mô hình BERT: Kiến trúc, cơ chế hoạt động và ứng dụng trong sửa lỗi OCR.
- Các kỹ thuật tối ưu hóa mô hình để tăng tốc và giảm tài nguyên tính toán.

# CHƯƠNG 3: TRIỂN KHAI THỰC NGHIỆM

## 3.1. DỮ LIỆU THỰC NGHIỆM

Bộ dữ liệu được sử dụng trong đề án bao gồm 7282 cặp ảnh và văn bản tiếng Việt tương ứng, được thu thập từ các nguồn công khai. Mỗi cặp dữ liệu gồm:

- Ảnh: Chứa các dòng chữ viết tay được thu thập trên internet.
- Văn bản gốc (ground truth): được trích xuất từ nội dung sách hoặc tài liệu tương ứng.

Dữ liệu được phân chia với cấu trúc 80% được sử dụng để huấn luyện và điều chỉnh mô hình; 20% còn lại được giữ nguyên để đánh giá hiệu suất của phương pháp đề xuất.

Mỗi ảnh và văn bản tương ứng được kiểm tra thủ công để đảm bảo độ khớp chính xác, đồng thời chuẩn hóa mã hóa Unicode và loại bỏ các ký tự không mong muốn.

## 3.2. CÁC CHỈ SỐ ĐÁNH GIÁ

Để đánh giá chất lượng của văn bản được trích xuất từ ảnh thông qua hệ thống OCR, đề án sử dụng hai chỉ số phổ biến trong lĩnh vực nhận dạng văn bản: Character Error Rate (CER) và Word Error Rate (WER). Ngoài ra, đề án cũng ghi nhận thời gian xử lý trung bình cho một mẫu nhằm đánh giá hiệu suất của hệ thống.

- Character Error Rate (CER)

Chỉ số CER đo lường tỷ lệ sai lệch giữa chuỗi ký tự được dự đoán và chuỗi ký tự gốc (ground truth), sử dụng khoảng cách chỉnh sửa (edit distance).

$$\mathrm{CER} = \frac{S + D + I}{N} \tag{3.1}$$

Trong đó:

+ S: số ký tự bị thay thế (substitutions)
+ D: Số ký tự bị xóa (deletions)
+ I: Số ký tự bị chèn (insertions)
+ N: Tổng số ký tự trong văn bản gốc.

Giá trị CER càng thấp thì mô hình càng chính xác ở cấp độ ký tự, đặc biệt quan trọng trong các ứng dụng yêu cầu độ chính xác cao như nhận dạng tên riêng, số liệu hoặc tiếng viết tay.

- Word Error Rate (WER):

WER đo lường tỷ lệ sai lệch ở cấp độ từ giữa văn bản đầu ra và văn bản gốc:

$$\text{WER} = \frac{S + D + I}{N} \tag{3.2}$$

Trong đó:

+ S, D, I: Lần lượt là số từ bị thay thế, xóa và chèn thêm.
+ N: Tổng số từ trong văn bản gốc

WER đánh giá khả năng của hệ thống trong việc duy trì cấu trúc ngữ nghĩa của văn bản và thường được sử dụng trong các bài toán nhận dạng ngôn ngữ tự nhiên.

Bên cạnh độ chính xác, hiệu suất xử lý iệu suất xử lý cũng là một yếu tố quan trọng. Đề án tiến hành đo thời gian trung bình để hệ thống xử lý một mẫu ảnh, bao gồm các bước tiền xử lý, nhận dạng văn bản và hậu xử lý.

## 3.3. THỰC NGHIỆM TĂNG CƯỜNG HIỆU SUẤT BÀI TOÁN OCR BẰNG MÔ HÌNH BERT

### 3.3.1. Lựa chọn baseline và tạo lập dữ liệu huấn luyện:

Đối với tác vụ sửa lỗi chính tả sau khi nhận dạng chữ viết tay (Handwritten Text Recognition - HTR), chúng ta thường có một chuỗi văn bản đầu vào bị lỗi và mục tiêu là thay thế các lỗi đó bằng từ đúng mà vẫn giữ được ngữ cảnh của câu.

Lý do BERT phù hợp với tác vụ sửa lỗi chính tả sau khi nhận dạng chữ viết tay:

- Hiểu ngữ cảnh hai chiều: Khi một từ bị nhận dạng sai, để sửa lỗi chính tả, mô hình cần biết cả những từ đứng trước và đứng sau từ đó. Ví dụ, nếu HTR trả về "Tôi đi học" thay vì "Tôi đi chợ", BERT có thể dựa vào từ "đi" và các từ khác trong câu để đưa ra dự đoán chính xác hơn về từ "chợ" hoặc "học". Khả năng này của BERT, thông qua MLM, cho phép nó "điền vào chỗ trống" những từ bị lỗi một cách hiệu quả.
- Phát hiện và sửa lỗi: BERT được tinh chỉnh (fine-tuned) để xác định vị trí lỗi và đề xuất các từ thay thế phù hợp dựa trên ngữ cảnh toàn cục của câu. Nó không chỉ đơn thuần là sinh ra một chuỗi mới mà là sửa chữa một chuỗi hiện có.
- Tối ưu hóa cho tác vụ hiểu: Sửa lỗi chính tả về bản chất là một tác vụ hiểu ngôn ngữ sâu sắc, nơi mô hình cần phân tích cấu trúc và ý nghĩa của câu để xác định lỗi.

  Hạn chế của GPT cho tác vụ sửa lỗi chính tả:
- Thiên về sinh văn bản: GPT có xu hướng tạo ra một chuỗi văn bản mới hoàn toàn từ một đầu vào, thay vì sửa chữa từng phần của chuỗi hiện có. Nếu dùng GPT, bạn có thể phải cấp toàn bộ câu bị lỗi và yêu cầu nó tạo

ra một phiên bản đúng, điều này có thể dẫn đến việc thay đổi cả những phần không lỗi hoặc sinh ra văn bản khác với ý định ban đầu.

- Hạn chế ngữ cảnh một chiều: Vì GPT chỉ nhìn vào ngữ cảnh bên trái, nó có thể gặp khó khăn trong việc sửa lỗi nếu thông tin quan trọng để sửa lỗi nằm ở phía bên phải của từ bị lỗi. Ví dụ, nếu từ cuối cùng của câu bị lỗi, GPT sẽ không có ngữ cảnh sau đó để tham chiếu.
- Khả năng gây "ảo giác" (hallucination): GPT có thể tạo ra các từ hoặc cụm từ không có trong bản gốc, hoặc thay đổi ý nghĩa của câu khi cố gắng sửa lỗi, do bản chất sinh văn bản của nó.

Để cải thiện hiệu suất nhận dạng văn bản trong bài toán OCR, đề án đề xuất tích hợp mô hình ngôn ngữ BERT với vai trò sửa lỗi sau OCR. Mô hình BERT sẽ được huấn luyện để phục hồi các từ sai trong kết quả nhận dạng, từ đó giúp giảm các chỉ số sai số như WER và CER.

Quá trình thực nghiệm bao gồm hai giai đoạn chính:

#### 3.3.1.1. Lựa chọn mô hình OCR baseline

Trước tiên, đề án tiến hành đánh giá ba mô hình OCR phổ biến và có độ chính xác cao hiện nay, bao gồm:

- VietOCR: Thư viện OCR mã nguồn mở hỗ trợ đặc biệt cho ngôn ngữ Tiếng Việt, sử dụng kết hợp mô hình CNN VGG19 và cung cấp cho ta 2 lựa chọn kết hợp các mô hình ngôn ngữ là Transformer và Sequence to Sequence.
- EasyOCR: Thư viện OCR mã nguồn mở hỗ trợ nhiều ngôn ngữ, sử dụng kết hợp CNN và LSTM.
- Tesseract OCR: Công cụ OCR truyền thống của Google, có khả năng tùy chỉnh linh hoạt và hỗ trợ nhiều định dạng ảnh.

Baseline được chọn cũng sẽ được huấn luyện với tác vụ OCR để nhằm tăng hiệu suất trên tác vụ downstream.

#### 3.3.1.2. Tạo lập dữ liệu:

Sau khi lựa chọn được baseline OCR, kết quả đầu ra của mô hình này sẽ được so sánh với văn bản gốc để xác định các từ bị nhận dạng sai. Quá trình tạo dữ liệu huấn luyện cho BERT bao gồm các bước:

- So sánh văn bản dự đoán và văn bản gốc để xác định vị trí các từ bị sai.
- Thay thế các từ bị sai bằng token đặc biệt [MASK] trong chuỗi đầu vào.
- Đặt từ đúng từ văn bản gốc làm nhãn đầu ra cho vị trí tương ứng.

Bằng cách này, dữ liệu huấn luyện cho mô hình BERT sẽ đóng vai trò như một bài toán điền từ bị che (Masked Language Modeling), nhưng tập trung vào các vị trí sai lệch trong kết quả OCR.

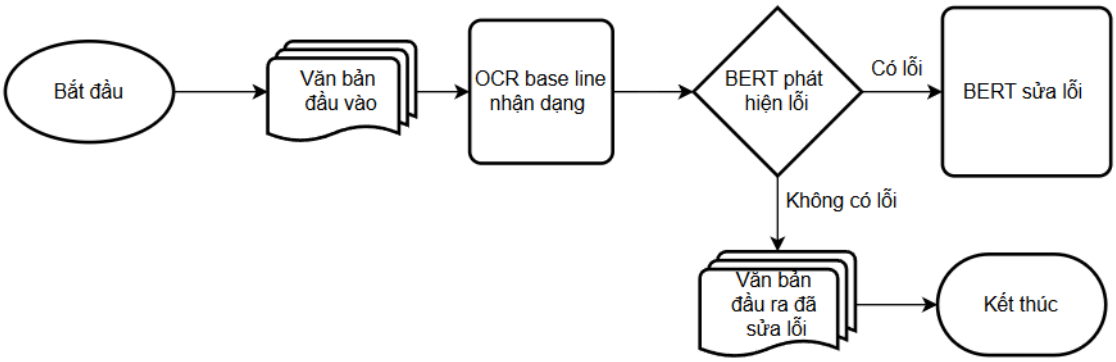
### 3.3.2. Phương pháp đề xuất

Đề án đã triển khai và cài đặt các thuật toán bao gồm:

- Đề án sử dụng một trong ba thuật toán OCR (Tesseract, EasyOCR, $\mathrm{TrOCR}_{\mathrm{base}}$ ) được lựa chọn trên chỉ số WER/CER để nhận diện văn bản từ hình ảnh chữ viết tay hoặc chữ in. Baseline được chọn sẽ được finetune trên dữ liệu thu thập được nhằm đạt được accuracy đủ tốt để số từ đưa vào huấn luyện mô hình BERT là đủ để không đánh mất ý nghĩa của cả câu. Các văn bản trích xuất được sẽ chứa một số lỗi nhận dạng do các yếu tố như độ rõ nét của chữ viết, font chữ phức tạp hoặc sự mờ nhòe của hình ảnh.
- Sau khi có văn bản đầu ra từ OCR,đề án áp dụng các mô hình BERT ($\mathrm{BERT}_{\mathrm{base}}$, $\mathrm{BERT}_{\mathrm{large}}$, DistilBERT) để sửa lỗi chính tả và ngữ pháp. Các mô hình BERT sẽ học ngữ cảnh trong văn bản và sửa các từ sai hoặc không hợp lý, từ đó cải thiện chất lượng của văn bản nhận dạng.

- Cuối cùng, đề án sử dụng các chỉ số như Word Error Rate (WER), Character Error Rate (CER) và thời gian xử lý trung bình của 1 mẫu để đánh giá hiệu suất của phương pháp đề xuất. Tôi so sánh các kết quả nhận dạng văn bản từ các thuật toán OCR và hiệu quả sửa lỗi từ các mô hình BERT để chọn ra phương pháp tối ưu.

Phương pháp đề xuất trong đề án này kết hợp các thuật toán OCR mạnh mẽ và các mô hình BERT để nâng cao hiệu quả trong bài toán nhận dạng văn bản, đặc biệt là đối với văn bản viết tay. Việc sử dụng nhiều thuật toán OCR cho phép chọn ra một baseline tối ưu, và việc áp dụng BERT giúp cải thiện chất lượng của văn bản đầu ra, từ đó giảm thiểu lỗi trong quá trình nhận diện. Phương pháp này có thể được áp dụng trong các hệ thống OCR nâng cao, đặc biệt là trong các bài toán yêu cầu độ chính xác cao và tính linh hoạt trong nhận dạng văn bản.




Hình 3.1: Tổng quan kiến trúc của phương pháp đề xuất

### 3.3.3. Môi trường thực nghiệm

Các thí nghiệm trong đề tài được triển khai hoàn toàn trên nền tảng Kaggle Notebooks, một môi trường lập trình trực tuyến phổ biến hỗ trợ xử lý dữ liệu

lớn và huấn luyện mô hình học sâu với GPU miễn phí. Cụ thể, môi trường phần cứng sử dụng trong toàn bộ quá trình thực nghiệm bao gồm:

GPU: NVIDIA Tesla P100 với 16GB bộ nhớ VRAM

CPU: 2 vCPU Intel Xeon

RAM: 13 GB

Lưu trữ: 20 GB bộ nhớ tạm thời (ephemeral storage)

Trong quá trình thực nghiệm, dữ liệu được tổ chức và xử lý thông qua quy trình bán tự động: ảnh từ bộ dữ liệu được trích xuất bằng các mô hình OCR khác nhau (baseline), sau đó kết quả được so sánh với văn bản gốc để tạo ra tập dữ liệu huấn luyện cho mô hình BERT (mô hình hậu xử lý sửa lỗi). Mọi bước từ sinh dữ liệu, huấn luyện đến đánh giá đều được tổ chức trong cùng một notebook trên Kaggle nhằm đảm bảo tính nhất quán và khả năng tái hiện kết quả.

Thông qua nền tảng này, quá trình thực nghiệm có thể linh hoạt, thuận tiện trong việc kiểm tra các cấu hình mô hình khác nhau, lưu trữ log kết quả, và dễ dàng chia sẻ hoặc tái hiện kết quả nghiên cứu trong cộng đồng học thuật.

### 3.3.4. Kết quả thực nghiệm

Để lựa chọn được mô hình OCR cơ sở (baseline) phù hợp, tiến hành đánh giá ba mô hình baseline đã đề xuất trên tập dữ liệu thu thập được. Đồng thời, quá trình khởi tạo dữ liệu cũng được thực hiện song song. Kết quả đánh giá được trình bày trong Bảng 3.1 như sau:

Bảng 3.1: Kết quả đánh giá các baseline trên dữ liệu

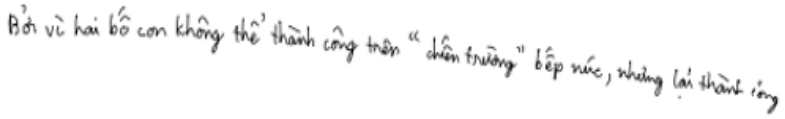
| **Baseline** | **Character Error Rate (%)** | **Word Error Rate (%)** | **Thời gian inference** |
|---|---|---|---|

| | | | |
|---|---|---|---|
| VietOCR (vgg-transformer) | 0.2805 | 0.6158 | 0.0500 |
| VetOCR (vgg-seq2seq) | 0.4225 | 0.7744 | 0.01268 |
| Tesseract OCR | 0.7268 | 0.9825 | 0.0912 |
| EasyOCR | 0.9397 | 0.9990 | 0.0969 |

Qua bảng kết quả, có thể nhận thấy hai phiên bản của framework VietOCR cho hiệu suất tốt hơn đáng kể so với các mô hình còn lại khi đạt mức lỗi ký tự (CER) và lỗi từ (WER) thấp nhất. Do đó, hai mô hình này được chọn làm baseline để tiếp tục thực hiện bước trích xuất thông tin.

Tiếp theo, cả hai mô hình VietOCR sẽ được fine-tune trên toàn bộ tập dữ liệu đã thu thập trong 5000 epochs, với kích thước batch là 32. Quá trình huấn luyện sử dụng thuật toán tối ưu AdamW kết hợp cùng chiến lược điều chỉnh learning rate ReduceLROnPlateau, với ngưỡng thay đổi (delta) là 0.00001 và số lượng epoch chờ (patience) là 100. Learning rate ban đầu được thiết lập ở mức 0.0005.

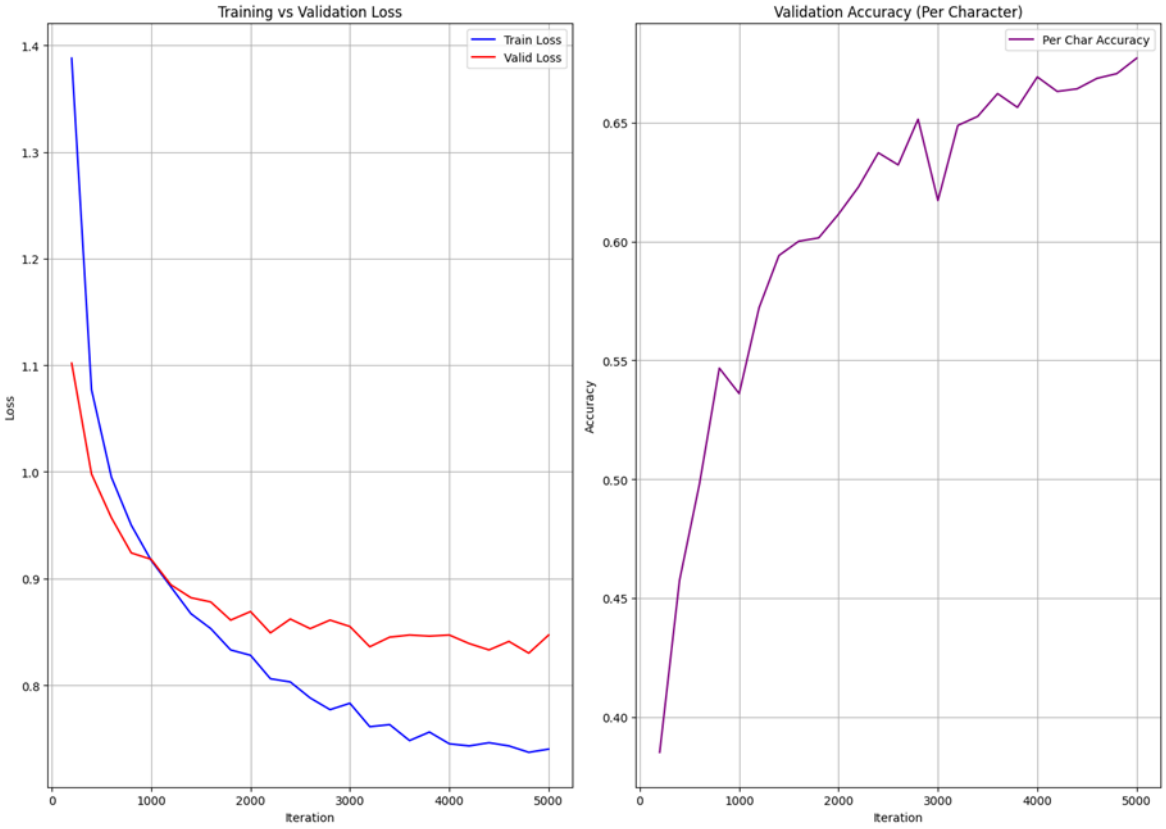
Mục tiêu chính của quá trình fine-tune là giảm chỉ số WER và CER xuống một ngưỡng nhất định, sao cho số lượng token bị dự đoán sai không làm ảnh hưởng nghiêm trọng đến ngữ nghĩa và tính toàn vẹn ngữ cảnh của văn bản được trích xuất.



Predicted text: Bởi và hai bố con không trở hành công kinh " đảo trường " đếp núc, nhưng lực trình trong

Grouth truth: Bởi và hai bố con không thể thành công trên " chiến trường " đếp núc, nhưng lại thành công

Hình 3.2: 1 mẫu dữ liệu của sau khi chạy baseline bao gồm ảnh, ground truth và predicted text của mô hình




Hình 3.3: Thông số mô hình vgg-seq2seq

a) Loss trên tập train, valid qua từng epoch (bên trái)

b) Accuracy trên ký tự (bên phải)

Sau khi lựa chọn hai mô hình VietOCR ( VGG-Transformer và VGG-Seq2Seq) làm baseline dựa trên kết quả đánh giá WER và CER như trình bày ở Bảng 3.1, quá trình fine-tune được tiến hành nhằm giảm thiểu số lượng lỗi ký tự và từ xuống một ngưỡng chấp nhận được. Mục tiêu chính của việc huấn luyện lại là đảm bảo rằng các token bị sai sót trong quá trình trích xuất không làm ảnh hưởng đáng kể đến ngữ nghĩa và tính toàn vẹn ngữ cảnh của văn bản đầu ra.

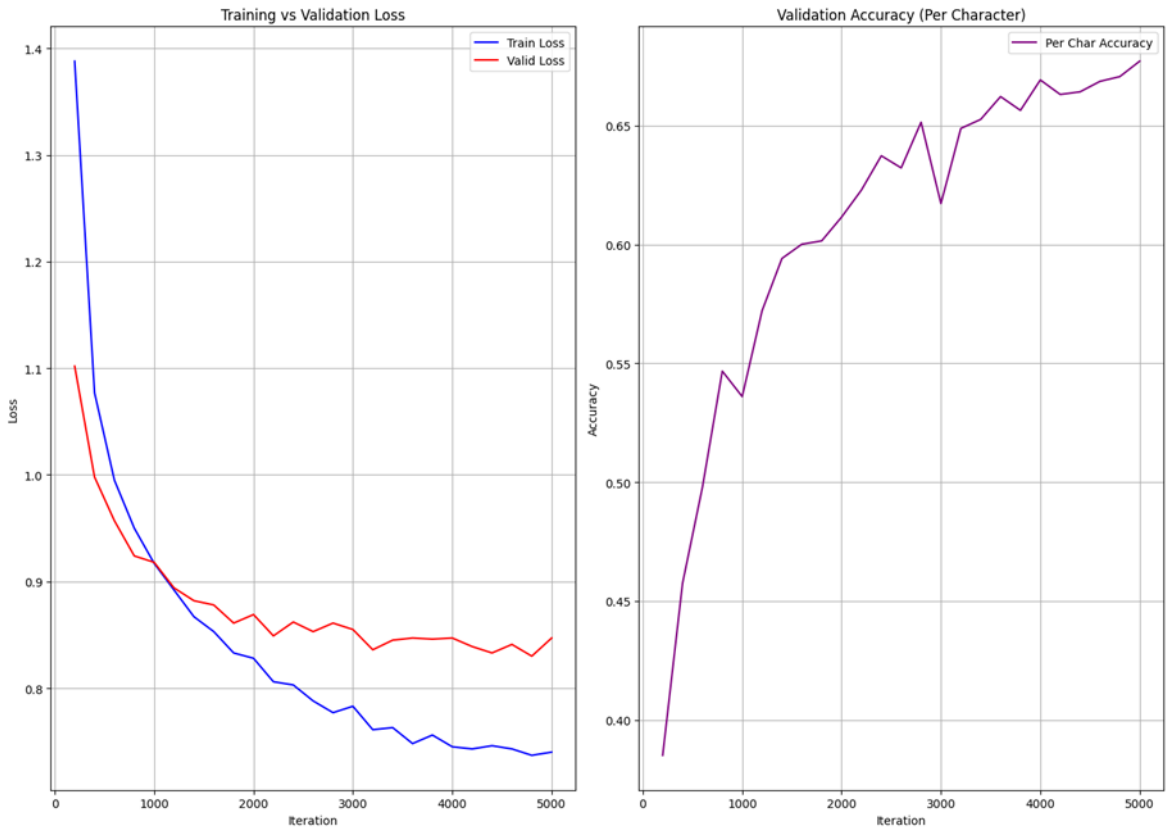
Quá trình fine-tune được thực hiện trên tập dữ liệu thu thập được với 5000 epochs, batch size là 32, sử dụng thuật toán tối ưu AdamW cùng chiến lược

điều chỉnh learning rate là ReduceLROnPlateau. Tham số delta được thiết lập là 0.00001 và số epoch chờ (patience) là 100. Learning rate khởi tạo ban đầu ở mức 0.0005 để hỗ trợ quá trình hội tụ ổn định.

Kết quả huấn luyện của mô hình VGG-Seq2Seq được thể hiện trong Hình 3.3, bao gồm hai biểu đồ thể hiện diễn biến của loss và độ chính xác trên từng ký tự qua các epoch. Biểu đồ bên trái cho thấy sự thay đổi của train loss và validation loss. Ban đầu, cả hai đại lượng này đều cao nhưng nhanh chóng giảm mạnh trong khoảng 1000 epoch đầu tiên, sau đó tiếp tục giảm chậm hơn và dần ổn định quanh giá trị 0.75–0.8. Sự chênh lệch không lớn giữa train loss và validation loss cho thấy mô hình không bị overfitting nghiêm trọng và đang học tốt trên dữ liệu huấn luyện.

Biểu đồ bên phải thể hiện độ chính xác trên từng ký tự (per character accuracy) trong quá trình huấn luyện. Độ chính xác bắt đầu từ mức rất thấp (40.93%), sau đó tăng nhanh từ khoảng epoch thứ 500 đến 1500 và đạt đỉnh tại mức 71.83%, sau đó duy trì ổn định quanh giá trị này. Điều này phản ánh khả năng học và khôi phục ký tự ngày càng hiệu quả của mô hình trong suốt quá trình huấn luyện.

Như vậy, kết quả này đáp ứng mục tiêu đề ra trong quá trình fine-tune là giảm xuống mức chấp nhận được số lượng lỗi trong quá trình OCR, đảm bảo rằng các token bị lỗi không làm sai lệch nhiều tới ngữ nghĩa tổng thể của văn bản. Dù con số 71.83% về độ chính xác từng ký tự vẫn còn tiềm ẩn một tỷ lệ lỗi nhất định, nhưng nó cho thấy bước đầu mô hình đã cải thiện đáng kể so với giai đoạn baseline, đặc biệt là đối với các trường hợp khó như font chữ không chuẩn, nhiễu ảnh hoặc bố cục phức tạp.




Hình 3.4: Thông số mô hình vgg-transformer

a) Loss trên tập train, valid qua từng epoch (bên trái)

b) Accuracy trên ký tự (bên phải)

Hình 3.4 thể hiện diễn biến của loss và độ chính xác trong quá trình huấn luyện mô hình VGG-Transformer. Biểu đồ bên trái minh họa sự thay đổi của train loss và validation loss qua từng epoch. Ban đầu, cả hai giá trị đều ở mức cao, nhưng sau đó nhanh chóng giảm mạnh trong khoảng 1000 epoch đầu tiên, cho thấy mô hình đang học tốt từ dữ liệu huấn luyện. Đến khoảng epoch thứ 2000, train loss và validation loss dần ổn định quanh mức 0.75–0.8. Sự chênh lệch nhỏ giữa hai chỉ số này cho thấy mô hình không gặp vấn đề lớn về overfitting và có khả năng tổng quát tốt trên tập dữ liệu kiểm thử.

Biểu đồ bên phải thể hiện độ chính xác trên từng ký tự (per character accuracy) của mô hình trên tập valid. Ban đầu, độ chính xác này khá thấp (38.51%), nhưng bắt đầu tăng nhanh từ khoảng epoch thứ 500 và đạt đỉnh tại mức 67.06% vào khoảng epoch thứ 4800. Sau đó, độ chính xác duy trì ổn định mà không có sự biến động đáng kể, điều này cho thấy mô hình đã hội tụ và không còn cải thiện thêm nếu tiếp tục huấn luyện thêm nhiều epoch hơn.

So với mô hình VGG-Seq2Seq được trình bày trước đó, mô hình VGG-Transformer thể hiện hành vi tương tự về mặt diễn biến loss và độ chính xác. Cả hai mô hình đều đạt được ngưỡng per char accuracy tối đa tương đối tốt, cho thấy hiệu suất trích xuất ký tự từ hình ảnh tương đương nhau trên tập dữ liệu hiện tại. Tuy nhiên, cần lưu ý rằng mỗi kiến trúc có những ưu điểm riêng về tốc độ hội tụ, khả năng xử lý ngữ cảnh dài hoặc ngắn hạn, và mức độ phức tạp tính toán.

Kết quả thu được từ cả hai mô hình VGG-Seq2Seq và VGG-Transformer đã phần nào đáp ứng mục tiêu đề ra ban đầu là giảm lỗi xuống mức chấp nhận được, giúp đảm bảo rằng các token bị dự đoán sai không làm méo mó thông tin ngữ nghĩa quan trọng trong văn bản đầu ra. Dù vẫn tồn tại một tỷ lệ lỗi nhất định, nhưng việc cải thiện so với baseline cho thấy hiệu quả của quá trình fine-tune và lựa chọn kiến trúc phù hợp với đặc điểm dữ liệu đầu vào. Những kết quả này sẽ là cơ sở để đánh giá sâu hơn về hiệu suất cuối cùng của mô hình cũng như đưa ra hướng phát triển trong các bước tiếp theo của nghiên cứu.

Sau khi hoàn tất quá trình fine-tuning trên hai mô hình VietOCR là VGG-Transformer và VGG-Seq2Seq, tôi tiến hành đánh giá hiệu suất của các mô hình đã được điều chỉnh trọng số trên tập dữ liệu kiểm thử. Bảng 3.2 dưới đây thể hiện sự thay đổi rõ rệt về các chỉ số lỗi ký tự (CER – Character Error Rate) và lỗi từ (WER – Word Error Rate) trước và sau khi fine-tune.

Bảng 3.2: Hiệu suất của mô hình sau khi finetuning

| Mô hình | Pretrained CER | Fintuned CER | Pretrained WER | Fintuned WER |
|---|---|---|---|---|
| VietOCR (vgg-transformer) | 0.2805 | 0.0904 | 0.6158 | 0.2340 |
| VietOCR (vgg-seq2seq) | 0.4225 | 0.0843 | 0.7744 | 0.2133 |

Từ bảng kết quả có thể nhận thấy rằng cả hai mô hình đều đạt được sự cải thiện đáng kể về mặt hiệu suất sau khi được fine-tune. Cụ thể, đối với mô hình VietOCR (vgg-transformer), chỉ số CER giảm từ 0.2805 xuống còn 0.0904, tức là mức lỗi ký tự đã giảm hơn ba lần. Tương tự, WER cũng giảm từ 0.6158 xuống còn 0.2340, cho thấy khả năng khôi phục từ ngữ chính xác hơn rất nhiều so với phiên bản chưa huấn luyện lại. Đối với mô hình VietOCR (vgg-seq2seq), dù có điểm số ban đầu cao hơn cả về CER và WER, nhưng sau khi fine-tune, mô hình này không chỉ thu hẹp khoảng cách mà còn vượt qua mô hình transformer về cả hai chỉ số, với CER đạt mức 0.0843 và WER ở mức 0.2133 – đều thấp hơn so với mô hình vgg-transformer.

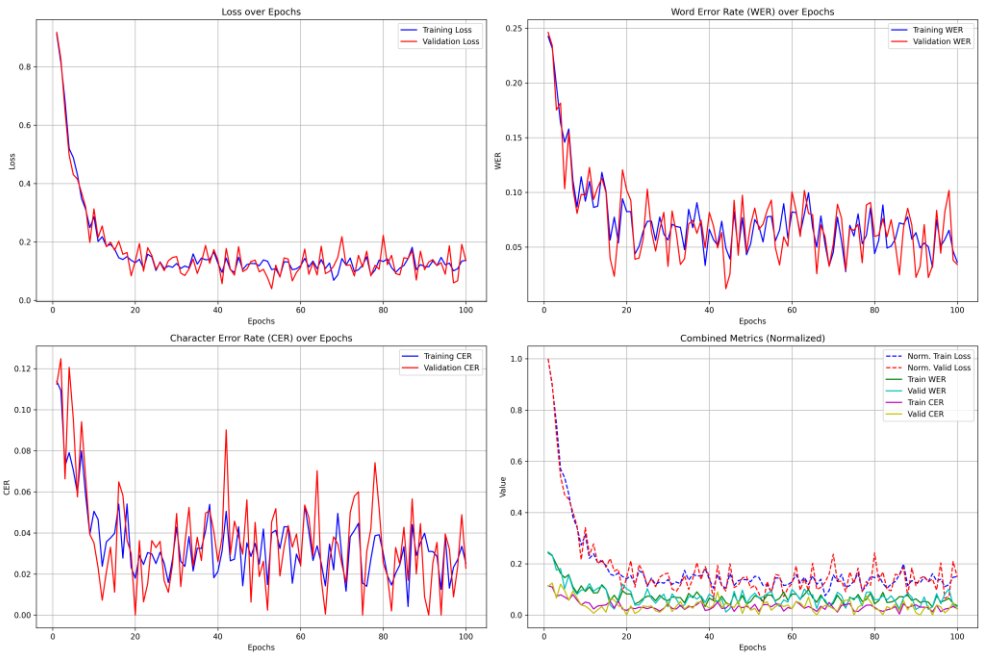
Ngoài ra, dựa vào thông tin thời gian inference từ Bảng 3.1, ta thấy rằng mô hình seq2seq có thời gian xử lý trung bình mỗi mẫu nhanh hơn đáng kể so với mô hình transformer (0.01268 giây/mẫu so với 0.0500 giây/mẫu). Điều này cho thấy mô hình seq2seq không chỉ cho hiệu suất tốt hơn về mặt độ chính xác mà còn vượt trội hơn về tốc độ xử lý, một yếu tố quan trọng trong các ứng dụng thực tế cần xử lý lượng lớn văn bản trong thời gian ngắn.

Từ những phân tích trên, mặc dù cả hai mô hình đều cho kết quả khả quan sau khi fine-tune, nhưng mô hình VietOCR (vgg-seq2seq) thể hiện ưu thế vượt trội hơn về cả độ chính xác lẫn tốc độ suy diễn. Do đó, để đảm bảo chất lượng dữ liệu đầu ra ổn định và hiệu quả cho giai đoạn tiếp theo trong quy trình

xử lý, đặc biệt là bước tạo dữ liệu detect dùng cho task downstream, nhóm nghiên cứu quyết định chọn VietOCR (vgg-transformer) làm baseline chính cho phần sinh dữ liệu.

Việc lựa chọn này không chỉ giúp tối ưu hóa quá trình trích xuất văn bản từ hình ảnh mà còn đóng vai trò nền tảng quan trọng trong việc xây dựng hệ thống tổng thể với độ tin cậy cao, từ đó hỗ trợ tốt hơn cho các bước tiền xử lý và mô hình phát hiện lỗi trong các chương trình tiếp theo của luận văn.

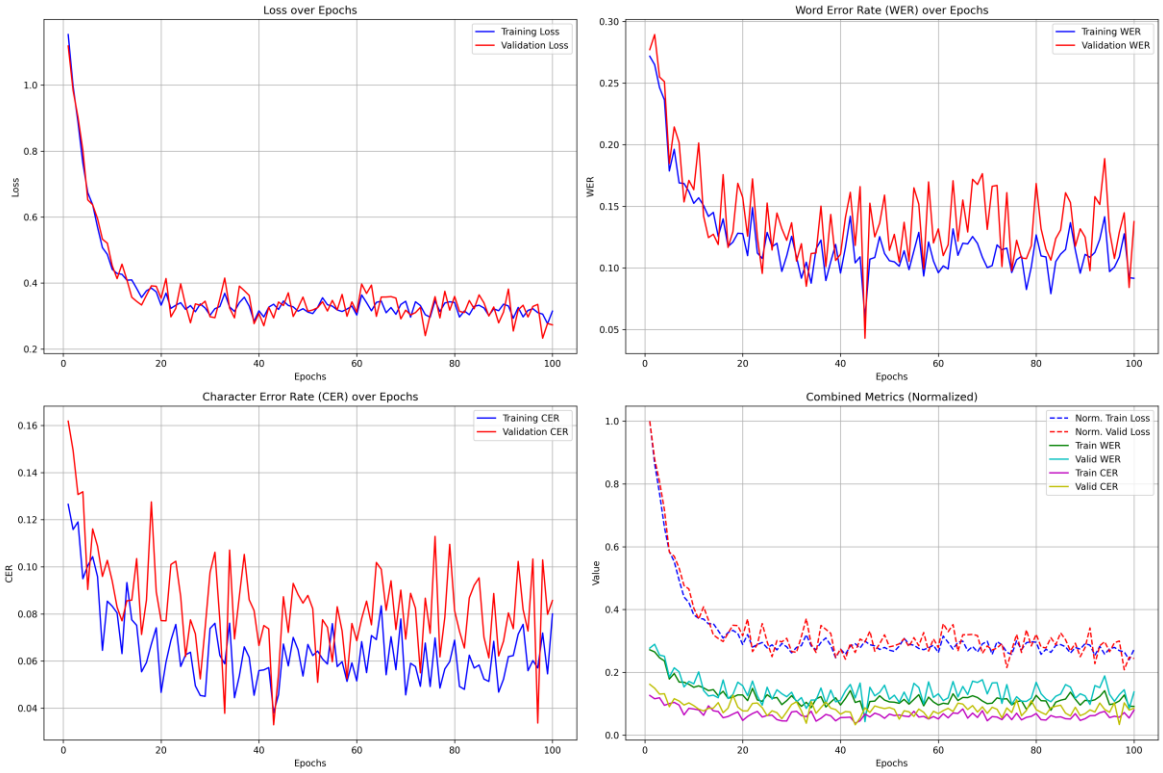
Sau khi hoàn thành việc fine-tune các mô hình VietOCR (vgg-seq2seq và vgg-transformer) để cải thiện hiệu suất OCR, chúng tôi chuyển sang giai đoạn tiếp theo là huấn luyện các mô hình ngôn ngữ lớn (pre-trained language models) trên tác vụ Masked Language Modeling (MLM). Mục tiêu của bước này là tận dụng khả năng hiểu ngữ nghĩa mạnh mẽ của các mô hình ngôn ngữ lớn để hỗ trợ trong việc phát hiện và sửa lỗi văn bản sau khi trích xuất từ hình ảnh. Ba mô hình được lựa chọn để thử nghiệm lần lượt là $BERT_{base}$, $BERT_{large}$ và DistilBERT




Hình 3.5: Loss trên tập train và valid qua từng epoch của mô hình $BERT_{large}$

Hình 3.5 thể hiện các thông số chính của mô hình $\text{BERT}_{\text{large}}$ multilingual trong suốt quá trình huấn luyện. Biểu đồ đầu tiên cho thấy diễn biến của train loss và validation loss qua từng epoch. Train loss giảm nhanh trong giai đoạn đầu (từ epoch 0 đến khoảng 20), sau đó tiếp tục giảm chậm và ổn định ở mức khoảng 0.1–0.2. Validation loss cũng có xu hướng tương tự nhưng dao động nhiều hơn một chút, song vẫn duy trì ở mức gần với train loss, điều này cho thấy mô hình không bị overfitting nghiêm trọng và đang học tốt trên dữ liệu kiểm thử. Biểu đồ thứ hai và thứ ba minh họa sự thay đổi của Word Error Rate (WER) và Character Error Rate (CER) trên cả tập huấn luyện và tập kiểm thử. Cả hai chỉ số đều giảm mạnh trong giai đoạn đầu, cho thấy khả năng cải thiện đáng kể trong việc nhận diện từ và ký tự. WER cuối cùng đạt mức 0.034, trong khi CER đạt mức 0.0228, đây là những con số khá tích cực, đặc biệt khi so sánh với hiệu suất ban đầu từ các mô hình OCR chưa qua xử lý hậu kỳ. Biểu đồ cuối cùng thể hiện các chỉ số chuẩn hóa (normalized) như loss, WER và CER, cho thấy sự đồng bộ trong quá trình hội tụ của mô hình, với hầu hết các đại lượng đều ổn định sau khoảng 20–30 epoch.

Từ các biểu đồ trong Hình 3.5, có thể rút ra một số nhận xét quan trọng. Thứ nhất, mô hình BERT_base multilingual học rất nhanh trong giai đoạn đầu, thể hiện qua tốc độ giảm mạnh của các chỉ số lỗi và mất mát. Thứ hai, sự chênh lệch giữa các chỉ số trên tập huấn luyện và tập kiểm thử không lớn, chứng tỏ rằng mô hình có khả năng tổng quát tốt và ít gặp vấn đề về overfitting. Tuy nhiên, sự dao động nhẹ của validation loss và các chỉ số WER/CER ở giai đoạn cuối có thể phản ánh tính phức tạp của dữ liệu hoặc kích thước tập valid chưa đủ lớn để đánh giá chính xác. Dù vậy, kết quả thu được vẫn cho thấy rằng mô hình đã đạt trạng thái hội tụ ổn định và có tiềm năng hỗ trợ tốt trong việc sửa lỗi văn bản sau OCR.




Hình 3.6: Loss trên tập train và valid qua từng epoch của mô hình $BERT_{base}$

Hình 3.6 thể hiện diễn biến của các chỉ số quan trọng trong quá trình huấn luyện mô hình $BERT_{base}$ multilingual trên tác vụ Masked Language Modeling (MLM). Biểu đồ đầu tiên cho thấy sự thay đổi của loss qua từng epoch. Train loss giảm nhanh trong giai đoạn đầu, từ mức cao ban đầu (~1.0) xuống còn khoảng 0.4 sau 20 epoch, sau đó tiếp tục giảm chậm và ổn định quanh mức 0.3–0.4. Validation loss cũng có xu hướng tương tự như train loss nhưng dao động nhiều hơn một chút, tuy nhiên vẫn duy trì ở mức gần với train loss, điều này cho thấy mô hình không bị overfitting nghiêm trọng và đang học tốt trên tập dữ liệu kiểm thử.

Biểu đồ thứ hai và thứ ba lần lượt minh họa diễn biến của WER và CER trên cả tập huấn luyện lẫn tập kiểm thử. Đối với WER, cả train WER và valid WER đều giảm mạnh trong giai đoạn đầu, từ mức xấp xỉ 0.3 xuống còn khoảng

0.1–0.15 , sau đó duy trì ổn định mà không có sự biến động lớn. Tương tự, CER cũng thể hiện xu hướng cải thiện rõ rệt khi giảm từ khoảng 0.16 xuống thấp nhất là 0.0329, cho thấy khả năng nhận diện ký tự ngày càng chính xác của mô hình. Những kết quả này khẳng định rằng việc huấn luyện MLM đã giúp mô hình hiểu tốt hơn về cấu trúc ngôn ngữ, từ đó hỗ trợ hiệu quả trong việc sửa lỗi văn bản thu được từ bước OCR.

Biểu đồ cuối cùng tổng hợp các chỉ số đã chuẩn hóa (normalized), bao gồm train loss, validation loss, WER và CER, cho thấy tất cả đều giảm đồng bộ trong giai đoạn đầu và đạt trạng thái ổn định sau khoảng 20–30 epoch. Điều này chứng tỏ rằng mô hình đã hội tụ tốt và không cần thêm nhiều epoch để cải thiện thêm hiệu suất. Từ các biểu đồ phân tích, có thể nhận thấy rằng mô hình $BERT_{base}$ multilingual học rất nhanh trong giai đoạn đầu và duy trì hiệu suất ổn định trong suốt quá trình huấn luyện. Sự chênh lệch nhỏ giữa các chỉ số trên tập huấn luyện và tập kiểm thử cho thấy mô hình có khả năng tổng quát hóa cao và ít gặp vấn đề về overfitting. Mặc dù vậy, sự dao động nhẹ của validation loss và các chỉ số WER/CER ở giai đoạn cuối có thể phản ánh tính phức tạp của dữ liệu hoặc kích thước tập valid chưa đủ lớn để đánh giá chính xác hoàn toàn.

Nhìn chung, kết quả thu được từ Hình 3.5 cho thấy mô hình $BERT_{base}$ multilingual đã được huấn luyện thành công, với các chỉ số lỗi và mất mát đều giảm đáng kể và đạt mức ổn định. Những cải thiện về WER và CER là dấu hiệu tích cực, chứng tỏ rằng mô hình có tiềm năng hỗ trợ tốt trong việc phát hiện và sửa lỗi văn bản sau khi trích xuất từ hình ảnh.

Bảng 3.3: Bảng so sánh hiệu suất các mô hình BERT

| Mô hình | Word Error Rate | Character Error Rate | Thời gian inference trên 1 mẫu |
|---|---|---|---|
| BERT base | 0.1376 | 0.0330 | 0.034 |
| BERT large | **0.0340** | **0.0228** | 0.076 |
| Distill BERT | 0.2015 | 0.0505 | **0.027** |

Bảng 3.4: Bảng đánh giá hiệu suất của phương pháp đề xuất.

| Mô hình kết hợp với VietOCR | CER trước BERT | CER sau BERT | WER trước BERT | WER sau BERT | Thời gian inference trung bình trên 1 mẫu (OCR + BERT) |
|---|---|---|---|---|---|
| BERT base | 0.0843 | 0.0330 | 0.2133 | 0.1376 | 0.04688 |
| BERT large | | **0.0228** | | **0.0340** | 0.08868 |
| Distill BERT | | 0.0505 | | 0.2015 | **0.03968** |

Từ các kết quả được trình bày trong Bảng 3.3 và Bảng 3.4, có thể thấy rõ sự khác biệt về hiệu suất giữa ba mô hình ngôn ngữ BERT base, BERT large và Distill BERT khi được sử dụng như công cụ hỗ trợ sửa lỗi sau quá trình OCR. Trong đó, BERT large cho kết quả tốt nhất với chỉ số lỗi từ (WER) và lỗi ký tự (CER) đều thấp nhất (tương ứng là 0.0228 và 0.0340), đồng thời giúp

giảm đáng kể lỗi từ đầu ra của mô hình VietOCR (vgg-seq2seq), mang lại hiệu suất cuối cùng rất ấn tượng. Tuy nhiên, nhược điểm của BERT large là thời gian inference cao hơn đáng kể so với hai mô hình còn lại (0.08868 giây/mẫu), khiến nó không phù hợp với các ứng dụng yêu cầu xử lý nhanh hoặc tài nguyên tính toán hạn chế.

Ngược lại, Distill BERT có tốc độ xử lý nhanh nhất (chỉ 0.03968 giây/mẫu), nhưng lại cho hiệu suất kém nhất khi CER tăng lên đến 0.0505 và WER sau khi xử lý vẫn ở mức khá cao (0.2015), điều này cho thấy mô hình không những không cải thiện mà còn làm sai lệch văn bản gốc do tạo ra lỗi mới. Do đó, Distill BERT không được khuyến khích sử dụng nếu độ chính xác là yếu tố quan trọng cần đảm bảo.

Trong khi đó, BERT base là giải pháp trung hòa hợp lý giữa hiệu suất và tốc độ: đạt mức CER và WER lần lượt là 0.0330 và 0.1376 — không bằng BERT large nhưng vẫn tốt hơn Distill BERT đáng kể — đồng thời có thời gian xử lý chỉ 0.04688 giây/mẫu, nhanh hơn BERT large gần gấp hai lần. Đây là lựa chọn phù hợp cho các ứng dụng thực tế, nơi vừa yêu cầu chất lượng văn bản đầu ra ổn định, vừa cần đảm bảo hiệu suất xử lý trong thời gian chấp nhận được.

Việc kết hợp mô hình OCR mạnh mẽ như VietOCR(vgg-seq2seq) với một mô hình ngôn ngữ phù hợp đã chứng minh tính hiệu quả trong việc giảm thiểu lỗi và nâng cao chất lượng văn bản đầu ra, góp phần quan trọng vào quy trình xử lý văn bản hình ảnh trong Đề án này.

## 3.4. KẾT LUẬN CHƯƠNG 3

Trong chương 3, hệ thống nhận diện chữ viết tay kết hợp giữa mô hình OCR và mô hình ngôn ngữ BERT đã được triển khai và đánh giá thực nghiệm trên tập dữ liệu chữ viết tay tiếng Việt. Qua quá trình thử nghiệm, kết quả cho thấy mô hình BERT có khả năng cải thiện đáng kể độ chính xác của hệ thống

HTR tổng thể, đặc biệt trong việc sửa lỗi ký tự và ngữ nghĩa so với chỉ dùng OCR thuần túy.

Việc tinh chỉnh mô hình BERT trên dữ liệu đầu ra của OCR giúp mô hình thích nghi tốt hơn với các dạng lỗi đặc thù trong nhận diện chữ viết tay tiếng Việt. Bên cạnh đó, việc áp dụng các kỹ thuật tối ưu như sử dụng DistilBERT đã giúp cân bằng giữa độ chính xác và tốc độ xử lý, phù hợp với yêu cầu thực tế.

Những kết quả đạt được từ chương này đã khẳng định hiệu quả của việc tích hợp mô hình ngôn ngữ BERT vào hệ thống nhận diện chữ viết tay, đồng thời hoàn thiện mục tiêu nghiên cứu đã đặt ra trong đề tài.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI

## KẾT LUẬN

Đề án đã xây dựng thành công một quy trình xử lý văn bản viết tay tiếng Việt từ bước nhận dạng ban đầu đến bước hậu kỳ sử dụng mô hình ngôn ngữ. Mô hình VietOCR (vgg-seq2seq) được chọn làm nền tảng OCR chính nhờ hiệu suất ổn định, đồng thời việc kết hợp với BERT base hoặc BERT large đã giúp cải thiện chất lượng văn bản đầu ra đáng kể. Trong đó, nếu ưu tiên độ chính xác nên lựa chọn BERT large; nếu cần cân bằng giữa tốc độ và hiệu suất, thì BERT base là giải pháp tối ưu. Việc áp dụng các mô hình học sâu trong nghiên cứu này không chỉ đóng góp vào lĩnh vực OCR tiếng Việt mà còn mở ra hướng phát triển cho các hệ thống số hóa tài liệu trong tương lai.

## HƯỚNG PHÁT TRIỂN

Trong quá trình thực hiện nghiên cứu, bên cạnh những kết quả tích cực đạt được, vẫn tồn tại một số hạn chế cần được phân tích và đề xuất hướng khắc phục nhằm nâng cao hiệu quả ứng dụng của hệ thống OCR và mô hình ngôn ngữ hậu kỳ trong thực tế. Một trong những điểm yếu lớn nhất là nếu văn bản sau OCR có độ dài quá ngắn, thì ngay cả khi sử dụng mô hình ngôn ngữ mạnh như BERT, việc khôi phục ngữ nghĩa chính xác cũng trở nên rất khó khăn do thiếu bối cảnh. Ngược lại, nếu OCR nhận dạng sai quá nhiều, đặc biệt ở những từ khóa quan trọng hoặc cụm từ mang tính ngữ cảnh cao, thì mô hình ngôn ngữ cũng không thể sửa lỗi một cách chính xác mà có thể dẫn đến sai lệch thêm.

Để giải quyết vấn đề này, trong tương lai có thể mở rộng hướng nghiên cứu theo hai khía cạnh chính:

Thứ nhất , cần tiến hành fine-tuning các mô hình OCR và BERT trên dữ liệu cụ thể theo từng môi trường làm việc trước khi đưa vào sử dụng . Việc tùy chỉnh (customization) này sẽ giúp mô hình học được các đặc trưng riêng biệt về font chữ, kiểu viết tay, bố cục tài liệu hay thậm chí là từ vựng chuyên ngành.

Điều này không chỉ giúp giảm CER và WER đáng kể mà còn tăng khả năng phục hồi ngữ nghĩa trong điều kiện dữ liệu đầu vào bị nhiễu hoặc không hoàn chỉnh. Ngoài ra, việc xây dựng kho dữ liệu huấn luyện phong phú theo từng lĩnh vực (ví dụ: giáo dục, y tế, hành chính…) sẽ góp phần tạo ra các giải pháp OCR chuyên biệt, đáp ứng tốt hơn nhu cầu thực tế.

Thứ hai , nghiên cứu có thể được mở rộng sang các ngôn ngữ khác , đặc biệt là các ngôn ngữ có cấu trúc tương đồng với tiếng Việt như tiếng Thái, tiếng Lào hay tiếng Indonesia, để kiểm tra tính tổng quát của phương pháp đề xuất. Đồng thời, việc thử nghiệm trên các ngôn ngữ có cấu trúc ngữ âm và ngữ nghĩa khác biệt như tiếng Trung, tiếng Hàn hay tiếng Ả Rập sẽ giúp đánh giá khả năng thích nghi của mô hình OCR và BERT đa ngôn ngữ. Đây là bước đi quan trọng nhằm phát triển các hệ thống nhận dạng văn bản đa ngôn ngữ, phục vụ cho các ứng dụng quốc tế như xử lý hồ sơ di trú, dịch thuật tự động, hay hỗ trợ đa ngôn ngữ trong các tổ chức xuyên quốc gia.

Mặc dù hệ thống đề xuất đã đạt được hiệu suất ổn định trong việc nhận dạng và sửa lỗi văn bản viết tay tiếng Việt, nhưng để nâng cao tính ứng dụng và khả năng mở rộng, cần tiếp tục hoàn thiện mô hình thông qua fine-tuning theo ngữ cảnh cụ thể và mở rộng sang các ngôn ngữ khác. Những cải tiến này không chỉ giúp tối ưu hóa quy trình số hóa tài liệu mà còn mở ra hướng phát triển bền vững cho các hệ thống OCR thông minh trong tương lai.

# TÀI LIỆU THAM KHẢO

[1] "SRI International". Truy cập: 16 Tháng Tư 2025. [Online]. Available at: https://www.sri.com/hoi/handwriting-recognition-signature-verification-and-pen-input-computing/

[2] "ScienceDirect Topics". Truy cập: 16 Tháng Tư 2025. [Online]. Available at: https://www.sciencedirect.com/topics/computer-science/handwriting-recognition

[3] R. Smith, "An Overview of the Tesseract OCR Engine", trong *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*, Curitiba, Parana, Brazil: IEEE, tháng 9 2007, tr 629–633. doi: 10.1109/ICDAR.2007.4376991.

[4] A. Căliman, M. Ivanovici, và N. Richard, "Probabilistic pseudo-morphology for grayscale and color images", *Pattern Recognit.*, vol 47, số p.h 2, tr 721–735, tháng 2 2014, doi: 10.1016/j.patcog.2013.08.021.

[5] U.-V. Marti và H. Bunke, "The IAM-database: an English sentence database for offline handwriting recognition", *Int. J. Doc. Anal. Recognit.*, vol 5, số p.h 1, tr 39–46, tháng 11 2002, doi: 10.1007/s100320200071.

[6] R. Plamondon và S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol 22, số p.h 1, tr 63–84, tháng 1 2000, doi: 10.1109/34.824821.

[7] J. Memon, M. Sami, R. A. Khan, và M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)", *IEEE Access*, vol 8, tr 142642–142668, 2020, doi: 10.1109/ACCESS.2020.3012542.

[8] A. Vinciarelli, “A survey on off-line Cursive Word Recognition”, *Pattern Recognit.*, vol 35, số p.h 7, tr 1433–1446, tháng 7 2002, doi: 10.1016/S0031-3203(01)00129-7.

[9] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition”, *Proc. IEEE*, vol 77, số p.h 2, tr 257–286, tháng 2 1989, doi: 10.1109/5.18626.

[10] G. D. Finlayson, “Color in perspective”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol 18, số p.h 10, tr 1034–1038, tháng 10 1996, doi: 10.1109/34.541413.

[11] N. Arica và F. T. Yarman-Vural, “An overview of character recognition focused on off-line handwriting”, *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol 31, số p.h 2, tr 216–233, tháng 5 2001, doi: 10.1109/5326.941845.

[12] Y. Liang, M. C. Fairhurst, R. M. Guest, và M. Erbilek, “Automatic Handwriting Feature Extraction, Analysis and Visualization in the Context of Digital Palaeography”, *Int. J. Pattern Recognit. Artif. Intell.*, vol 30, số p.h 04, tr 1653001, tháng 5 2016, doi: 10.1142/S0218001416530013.

[13] C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition”, *Data Min. Knowl. Discov.*, vol 2, số p.h 2, tr 121–167, 1998, doi: 10.1023/A:1009715923555.

[14] N. D. Cilia, C. De Stefano, F. Fontanella, và A. Scotto Di Freca, “A ranking-based feature selection approach for handwritten character recognition”, *Pattern Recognit. Lett.*, vol 121, tr 77–86, tháng 4 2019, doi: 10.1016/j.patrec.2018.04.007.

[15] C.-L. Liu, K. Nakashima, H. Sako, và H. Fujisawa, “Handwritten digit recognition: benchmarking of state-of-the-art techniques”, *Pattern*

*Recognit.*, vol 36, số p.h 10, tr 2271–2285, tháng 10 2003, doi: 10.1016/S0031-3203(03)00085-2.

[16] Y. Lecun, L. Bottou, Y. Bengio, và P. Haffner, "Gradient-based learning applied to document recognition", *Proc. IEEE*, vol 86, số p.h 11, tr 2278–2324, tháng 11 1998, doi: 10.1109/5.726791.

[17] M. T. Parvez và S. A. Mahmoud, "Offline arabic handwritten text recognition: A Survey", *ACM Comput. Surv.*, vol 45, số p.h 2, tr 1–35, tháng 2 2013, doi: 10.1145/2431211.2431222.

[18] B. Shi, X. Bai, và C. Yao, "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol 39, số p.h 11, tr 2298–2304, tháng 11 2017, doi: 10.1109/TPAMI.2016.2646371.

[19] L. Uzan, N. Dershowitz, và L. Wolf, "Qumran Letter Restoration by Rotation and Reflection Modified PixelCNN", trong *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto: IEEE, tháng 11 2017, tr 23–29. doi: 10.1109/ICDAR.2017.14.

[20] "bidirectional-long-short-term-memory-network". Truy cập: 16 Tháng Tư 2025. [Online]. Available at: https://www.sciencedirect.com/topics/computer-science/bidirectional-long-short-term-memory-network

[21] M. Li *và c.s.*, "TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models", 2021, *arXiv*. doi: 10.48550/ARXIV.2109.10282.

[22] B. Shi, X. Bai, và C. Yao, "An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition", 21 Tháng Bảy 2015, *arXiv*: arXiv:1507.05717. doi: 10.48550/arXiv.1507.05717.

[23] J. Devlin, M.-W. Chang, K. Lee, và K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, trong *Proceedings of the 2019 Conference of the North*, Minneapolis, Minnesota: Association for Computational Linguistics, 2019, tr 4171–4186. doi: 10.18653/v1/N19-1423.

[24] “explanation-of-bert-model-nlp”. Truy cập: 16 Tháng Tư 2025. [Online]. Available at: https://www.geeksforgeeks.org/explanation-of-bert-model-nlp/

[25] “bert-101”. Truy cập: 16 Tháng Tư 2025. [Online]. Available at: https://huggingface.co/blog/bert-101

[26] A. Vaswani *và c.s.*, “Attention Is All You Need”, 2 Tháng Tám 2023, *arXiv*: arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762.

[27] D. Q. Nguyen và A. Tuan Nguyen, “PhoBERT: Pre-trained language models for Vietnamese”, trong *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online: Association for Computational Linguistics, 2020, tr 1037–1042. doi: 10.18653/v1/2020.findings-emnlp.92.

[28] D. Hendrycks và K. Gimpel, “Gaussian Error Linear Units (GELUs)”, 6 Tháng Sáu 2023, *arXiv*: arXiv:1606.08415. doi: 10.48550/arXiv.1606.08415.

[29] K. He, X. Zhang, S. Ren, và J. Sun, “Deep Residual Learning for Image Recognition”, 10 Tháng Chạp 2015, *arXiv*: arXiv:1512.03385. doi: 10.48550/arXiv.1512.03385.

[30] J. L. Ba, J. R. Kiros, và G. E. Hinton, “Layer Normalization”, 21 Tháng Bảy 2016, *arXiv*: arXiv:1607.06450. doi: 10.48550/arXiv.1607.06450.

[31] J. Puigcerver, “Are Multidimensional Recurrent Layers Really Necessary for Handwritten Text Recognition?”, trong *2017 14th IAPR*

*International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto: IEEE, tháng 11 2017, tr 67–72. doi: 10.1109/ICDAR.2017.20.

[32] M. S. Amin, S. M. Yasir, và H. Ahn, "Recognition of Pashto Handwritten Characters Based on Deep Learning", *Sensors*, vol 20, số p.h 20, tr 5884, tháng 10 2020, doi: 10.3390/s20205884.

[33] Y. Zhuang *và c.s.*, "A Handwritten Chinese Character Recognition based on Convolutional Neural Network and Median Filtering", *J. Phys. Conf. Ser.*, vol 1820, số p.h 1, tr 012162, tháng 3 2021, doi: 10.1088/1742-6596/1820/1/012162.

[34] P. P. Quốc và V. Q. Phước, "Nhận dạng chữ số viết tay dùng mạng neuron nhân tạo", *Tạp chí Khoa học và Công nghệ – Trường Đại học Khoa học Huế*, vol. 14, no. 1, pp. 119–132, Jul. 2019.

[35] "Nhận dạng chữ viết tay sử dụng phương pháp mạng Nơ-ron", *luanvan.co* (Studocu), 2022?

[36] "BERT, RoBERTa, PhoBERT, BERTweet: Ứng dụng state-of-the-art model cho bài toán phân loại văn bản", Viblo, 2020

[37] Dat Q. Nguyen và Anh T. Nguyen, "PhoBERT: Pre-trained language models for Vietnamese", *Findings of the Association for Computational Linguistics: EMNLP 2020*, tr. 1037–1042

[38] H. S. (sic) in "PhoBERT-base-v2, một mô hình Transformer tối ưu cho tiếng Việt, đạt độ chính xác hiện đại 94,22%", *Journal of Science & Technology at Thai Nguyen University*, Jun. 29, 2025. doi:10.34238/tnu-jst.12889

[39] Nguyễn Hoài Nam, "Số hóa tài liệu chữ viết tay tại Việt Nam: Thực trạng và giải pháp," Tạp chí Khoa học Quản lý và Kinh tế, vol. 5, no. 2, tr. 98–110, 2023.